Part Three

Syllabus, Chapter 5:

Building a Bayesian Network

The construction of a Bayesian network (BN)

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

- to identify the (random) variables and their values;
- to construct the digraph of the network;
- to assess the (conditional) probability distributions required for the variables' assessment functions.

Methodologies for building networks hardly exist, just some best practices!

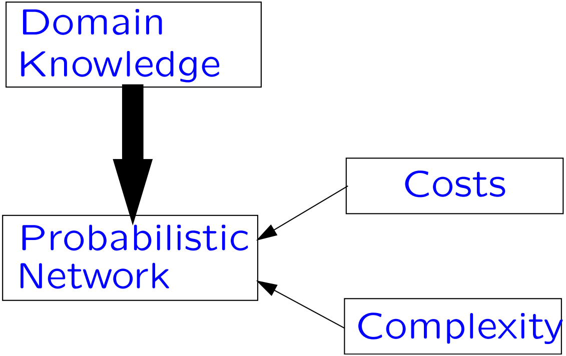
Building a BN resembles building any type of system, thereby warranting the use of an overall systems-engineering approach.

In practice, the construction of a BN is an iterative process involving testing and evaluation as well.

The trade-off in construction

The construction of a BN 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 BN 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 random 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 random variables.

Assuming a Bayesian network with discrete variables only:

- 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 random variables.

Single-valued variables

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

- To discretise a continuous variable, its value range must 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 discrete random 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 random variable;
- a multi-valued variable is usually modelled as a collection of single-valued random variables.

Multi-valued variables, an example

Consider the domain variable *BloodCount* that adopts one or more of the values $normal$, $lymphocytosis$, $lymphocytopenia$, $leucocytosis$, and $leucocytopenia$; possible combinations are:

$\{normal\}$	$\{lymphocytosis, leucocytosis\}$
$\{leucocytosis\}$	$\{lymphocytosis, leucocytopenia\}$
$\{lymphocytosis\}$	$\{lymphocytopenia, leucocytosis\}$
$\{leucocytopenia\}$	$\{lymphocytopenia, leucocytopenia\}$
$\{lymphocytopenia\}$	

- the variable can be modelled as a single random variable with the nine possible combinations of its values;
- the variable can be modelled by two random variables:
  - the variable *LymphocyteCount* with the three values $normal$, $lymphocytosis$, $lymphocytopenia$;
  - the variable *LeucocyteCount* with the three values $normal$, $leucocytosis$, $leucocytopenia$. ■

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.

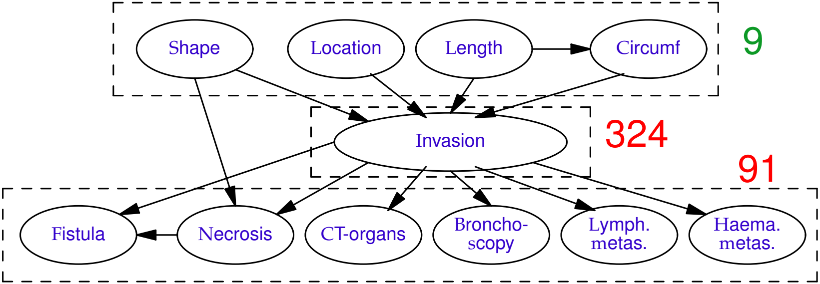
Example: Consider modelling the depth of invasion of an oesophageal tumour

- 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.

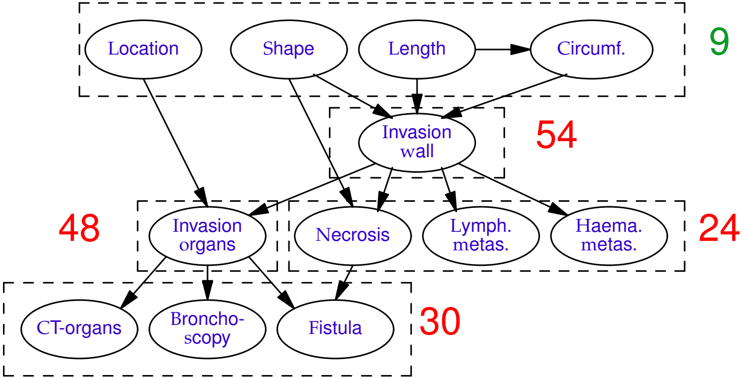
Example: Consider modelling the depth of invasion of an oesophageal tumour

- 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* $\vee$ *T2* $\vee$ *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:



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*

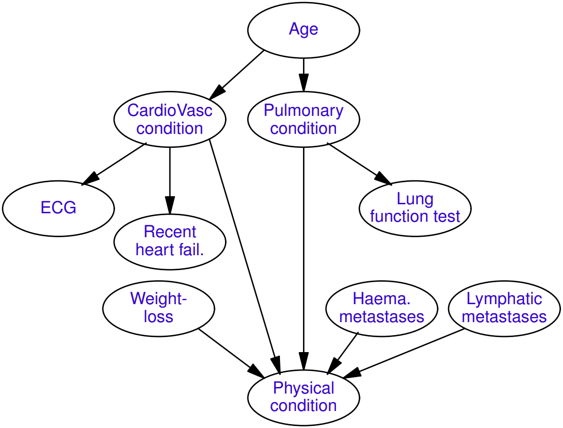
The number of non-redundant probability assessments required in the second representation is less than 40% of that required in the first representation!

The level of detail

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



An unambiguous description of: *Location*

Definition: The variable *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.

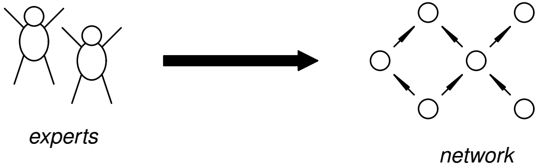
Values: The variable *Location* can adopt one of the values $proximal$, mid and $distal$:

- $proximal$: the tumour's center is in the upper $\frac{1}{3}$ of the oesophagus;
- mid: the tumour's center is in the middle $\frac{1}{3}$ of the oesophagus;
- $distal$: the tumour's center is in the lower $\frac{1}{3}$ of the oesophagus.

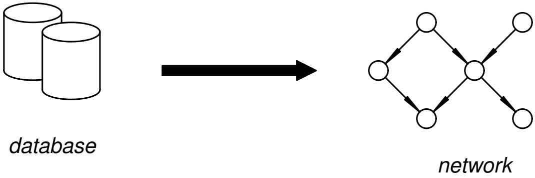
Probabilistic information: For the variable *Location* are specified 3 probabilities: $\Pr(Location)$

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 be constructed automatically from a suitable up-to-date dataset.



Constructing the digraph by hand

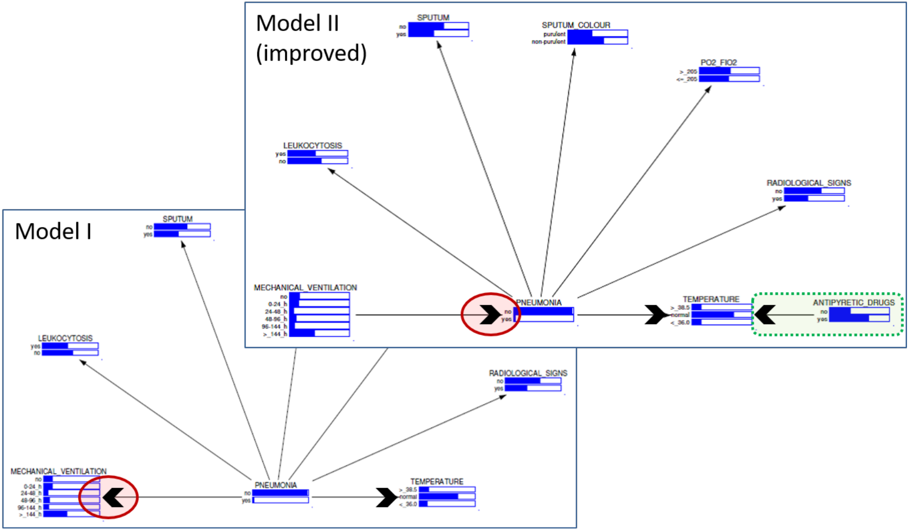
For the construction of the digraph of a Bayesian network by hand, the notion of causality is used as a heuristic guiding principle:

> *"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 !

Causal anecdote



Bayesian network models for the management of ventilator-associated pneumonia (S. Visscher, PhD Thesis, UU, 2008)

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.

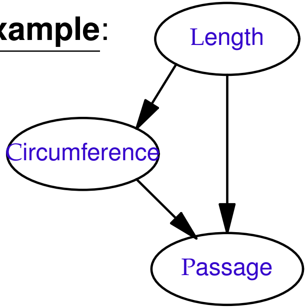
Fine-tuning the digraph: indirect arcs

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

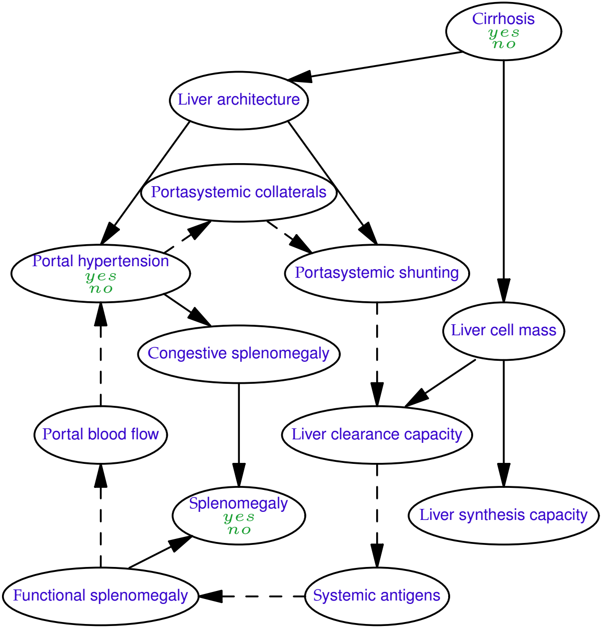
By using causality as a guiding principle, cycles may arise.

- the cycle can be the consequence of an erroneous arc;
- the cycle can model a feedback process in the domain of application.

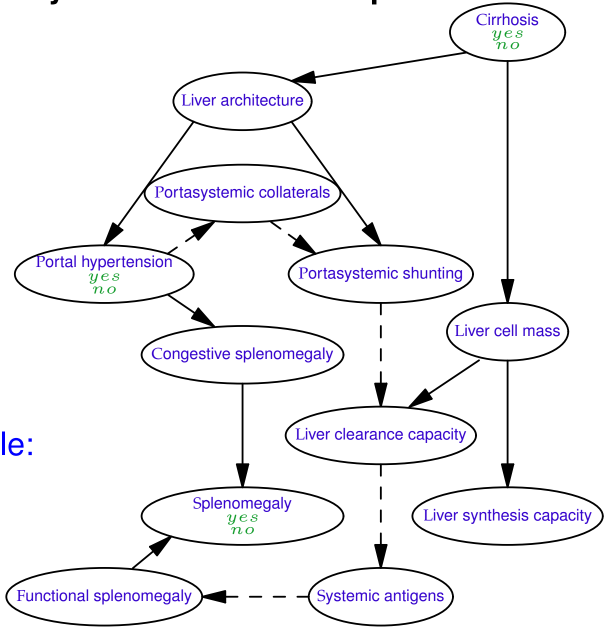
Any cycle needs to be broken, for example by

- 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



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.

Algorithms for automated graph construction

Consider a set of variables $\boldsymbol{V}$. The digraph of a BN can be automatically constructed from a dataset $\boldsymbol{D}$ by (possibly a combination of):

- constraint-based approaches
  - perform (conditional) independence tests on data
  - add arcs to G to match these independences
- score and search-based approaches
  - search in model space; e.g. the space of possible DAGs
  - measure match between model and data distributions

In both cases we need to create graphs, extract probabilistic information from data, and decide on the quality of the match.

These algorithms are often called structure learning algorithms and are typically iterative.

A dataset

Definition:

Let $\boldsymbol{V}$ be a set of domain variables. A dataset $\boldsymbol{D}$ over $\boldsymbol{V}$ is a multi-set of cases, which are configurations $c_{\boldsymbol{V}}$ of $\boldsymbol{V}$.

$\boldsymbol{D}$ can be used for learning a Bayesian network $\mathcal{B} = (G, \Gamma)$ if:

- the variables and values in $\boldsymbol{D}$ are (easily) translated to the variables and values of the network under construction;
- every case in $\boldsymbol{D}$ specifies a value for each variable;
- the cases in $\boldsymbol{D}$ are generated independently;
- $\boldsymbol{D}$ reflects a time-independent process;
- $\boldsymbol{D}$ contains sufficient and reliable information.

The information in a dataset describes a joint probability distribution $\Pr_{\boldsymbol{D}}(\boldsymbol{V})$ over its variables; this is an approximation of the true distribution $\Pr(\boldsymbol{V})$.

A CI structure learning algorithm (brief)

A conditional independence (CI) algorithm for learning a DAG from a dataset $\boldsymbol{D}$:

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
$I_{\boldsymbol{D}}(\{V_i\}, \delta(V_i), \{V_1, \ldots, V_{i-1}\} \setminus \delta(V_i))$;
$\boldsymbol{\rho}(V_i) \leftarrow \delta(V_i)$;

Benefit: guaranteed acyclic
Drawback: structure, and hence compactness, depends heavily on chosen ordering

Assessing probabilities from data

Let $\boldsymbol{V} = \{V_1, \ldots, V_n\}, n \geq 1$, be a set of discrete random variables and let $\boldsymbol{D}$ be a dataset over $\boldsymbol{V}$ with N cases.

Any probability from $\Pr_{\boldsymbol{D}}$ can be obtained from $\boldsymbol{D}$ by frequency counting.

For example, consider a variable $V_i \in \boldsymbol{V}$ and a subset of variables $\boldsymbol{W} \subseteq \boldsymbol{V} \setminus \{V_i\}$. Then, e.g.

$$\Pr_{\boldsymbol{D}}(c_{V_i}) = \frac{N(c_{V_i})}{N}, \quad \text{and}$$

$$\Pr_{\boldsymbol{D}}(c_{V_i} \mid c_{\boldsymbol{W}}) = \frac{\Pr_{\boldsymbol{D}}(c_{V_i} \wedge c_{\boldsymbol{W}})}{\Pr_{\boldsymbol{D}}(c_{\boldsymbol{W}})} = \frac{N(c_{V_i} \wedge c_{\boldsymbol{W}})/N}{N(c_{\boldsymbol{W}})/N} = \frac{N(c_{V_i} \wedge c_{\boldsymbol{W}})}{N(c_{\boldsymbol{W}})}$$

where $N(c)$ is the number of cases consistent with c.

Establishing assessment functions for $\mathcal{B}$

Let $\boldsymbol{V}$ be a set of discrete random variables, let $\boldsymbol{D}$ be a dataset over $\boldsymbol{V}$ with N cases and let G be a DAG with $\boldsymbol{V}_G = \boldsymbol{V}$.

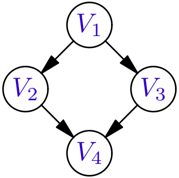
For G, a corresponding set $\Gamma = \{\gamma_{V_i} \mid V_i \in \boldsymbol{V}_G\}$ of assessment functions is obtained from $\boldsymbol{D}$, by frequency counting. That is,

$$\begin{array}{rcll} \gamma(c_{V_i} \mid c_{\boldsymbol{\rho}(V_i)}) & = & \Pr_{\boldsymbol{D}}(c_{V_i} \mid c_{\boldsymbol{\rho}(V_i)}) & \text{and} \\ \gamma(c_{V_i}) & = & \Pr_{\boldsymbol{D}}(c_{V_i}) & \text{if } \boldsymbol{\rho}(V_i) = \emptyset \end{array}$$

for each variable $V_i \in \boldsymbol{V}$, every configuration c_{V_i} of V_i and all configurations $c_{\boldsymbol{\rho}(V_i)}$ of the parent set $\boldsymbol{\rho}(V_i)$ of V_i in G.

Assessing γ_{V_i}: an example (1)

Consider the following dataset $\boldsymbol{D}$ and graph G:



$\neg v_1 \wedge \neg v_2 \wedge v_3 \wedge \neg v_4$ ✓
$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$
$\neg v_1 \wedge \neg v_2 \wedge v_3 \wedge v_4$ ✓
$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge \neg v_3 \wedge v_4$
$\neg v_1 \wedge \neg v_2 \wedge v_3 \wedge \neg v_4$ ✓

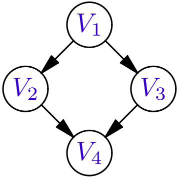
$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$\neg v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$ ✓
$\neg v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$ ✓
$v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$
$\neg v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$ ✓
$v_1 \wedge v_2 \wedge \neg v_3 \wedge v_4$

The values of γ_{V_1} are assessed as follows:

$$\gamma(\neg v_1) = \frac{N(\neg v_1)}{N} = \frac{6}{15} = 0.4 \quad \text{and} \quad \gamma(v_1) = \frac{N(v_1)}{N} = \ldots$$

Assessing γ_{V_i}: an example (2)

Consider the following dataset $\boldsymbol{D}$ and graph G:



$\neg v_1 \wedge \neg v_2 \wedge v_3 \wedge \neg v_4$ ✓
$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$
$\neg v_1 \wedge \neg v_2 \wedge v_3 \wedge v_4$ ✓
$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge \neg v_3 \wedge v_4$
$\neg v_1 \wedge \neg v_2 \wedge v_3 \wedge \neg v_4$ ✓

$v_1 \wedge v_2 \wedge \neg v_3 \wedge \neg v_4$
$\neg v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$ ✓✓
$\neg v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$ ✓✓
$v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$
$v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$
$\neg v_1 \wedge v_2 \wedge v_3 \wedge \neg v_4$ ✓✓
$v_1 \wedge v_2 \wedge \neg v_3 \wedge v_4$

The values of γ_{V_2} are assessed as follows:

$$\gamma(v_2 \mid \neg v_1) = \frac{N(\neg v_1 \wedge v_2)}{N(\neg v_1)} = \frac{3}{6} = 0.5, \quad \text{etc.} \ldots$$

A metric algorithm for structure learning

An (unsupervised metric) algorithm for automated construction of a BN $\mathcal{B}$ from a dataset $\boldsymbol{D}$ 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 $\Pr_{\boldsymbol{D}}$?

  We consider the MDL quality measure. The measure requires a complete network with probabilities; these are obtained by frequency 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).

The quality of a graph given the data

Definition: ('MDL quality measure')
Let $\boldsymbol{D}$ be a dataset with N cases over variables $\boldsymbol{V}$.

Let P be a joint distribution over the set of *all* DAGs G with node set $\boldsymbol{V}_G = \boldsymbol{V}$.

The quality of G given $\boldsymbol{D}$, notation: $Q(G, \boldsymbol{D})$, is defined as

$$Q(G, \boldsymbol{D}) = \log P(G) - N \cdot H(G, \boldsymbol{D}) - \frac{1}{2} K \cdot \log N$$

where

$$H(G, \boldsymbol{D}) = - \sum_{V_i \in \boldsymbol{V}} \sum_{c_{V_i}} \sum_{c_{\boldsymbol{\rho}(V_i)}} \left(\frac{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})}{N} \right) \cdot \log \left(\frac{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})}{N(c_{\boldsymbol{\rho}(V_i)})} \right)$$

and $K = \sum_{V_i \in \boldsymbol{V}} 2^{|\boldsymbol{\rho}(V_i)|}$ for binary-valued variables.

The entropy term $H(G, \boldsymbol{D})$

Let $\Pr$ be the joint distribution defined by $\mathcal{B}$ with DAG G with $\boldsymbol{V}_G = \boldsymbol{V}$, and Γ is obtained from $\boldsymbol{D}$. Then,

$$\log P'(\boldsymbol{D} \mid \mathcal{B}) = \log \prod_{c_{\boldsymbol{V}} \in \boldsymbol{D}} \Pr(c_{\boldsymbol{V}}) = \log \prod_{c_{\boldsymbol{V}} \in \boldsymbol{D}} \prod_{V_i \in \boldsymbol{V}} \gamma(c_{V_i} \mid c_{\boldsymbol{\rho}(V_i)}) =$$

$$= \log \prod_{V_i \in \boldsymbol{V}} \prod_{c_{V_i}} \prod_{c_{\boldsymbol{\rho}(V_i)}} \gamma_{V_i}(c_{V_i} \mid c_{\boldsymbol{\rho}(V_i)})^{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})} =$$

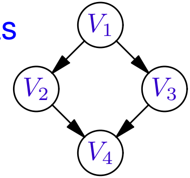
$$= \log \prod_{V_i \in \boldsymbol{V}} \prod_{c_{V_i}} \prod_{c_{\boldsymbol{\rho}(V_i)}} \left(\frac{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})}{N(c_{\boldsymbol{\rho}(V_i)})} \right)^{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})}$$

$$= N \cdot \sum_{V_i \in \boldsymbol{V}} \sum_{c_{V_i}} \sum_{c_{\boldsymbol{\rho}(V_i)}} \left(\frac{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})}{N} \right) \cdot \log \left(\frac{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})}{N(c_{\boldsymbol{\rho}(V_i)})} \right)$$

$$= -N \cdot H(G, \boldsymbol{D})$$

Computing quality $Q(G, \boldsymbol{D})$: an example (1)

Consider the same dataset $\boldsymbol{D}$ as before and the following graph G.



We first compute $-N \cdot H(G, \boldsymbol{D})$:

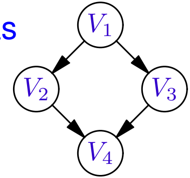
For V_1:

$$N(v_1) \log \frac{N(v_1)}{N} + N(\neg v_1) \log \frac{N(\neg v_1)}{N} = 9 \cdot \log \frac{9}{15} + 6 \cdot \log \frac{6}{15} = -4.384$$

(if we use the ${}^{10}\log$ for easy computation)

Computing quality $Q(G, \boldsymbol{D})$: an example (2)

Consider the same dataset $\boldsymbol{D}$ as before and the following graph G.



-4.384

We first compute $-N \cdot H(G, \boldsymbol{D})$:

For V_2:

$$N(v_2 \wedge v_1) \log \frac{N(v_2 \wedge v_1)}{N(v_1)} + N(\neg v_2 \wedge v_1) \log \frac{N(\neg v_2 \wedge v_1)}{N(v_1)} +$$

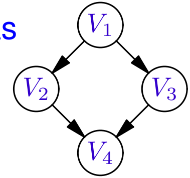
$$+ N(v_2 \wedge \neg v_1) \log \frac{N(v_2 \wedge \neg v_1)}{N(\neg v_1)} + N(\neg v_2 \wedge \neg v_1) \log \frac{N(\neg v_2 \wedge \neg v_1)}{N(\neg v_1)} =$$

$$= 9 \log \frac{9}{9} + 0 \log \frac{0}{9} + 3 \log \frac{3}{6} + 3 \log \frac{3}{6} = -1.806 \quad (\text{using } {}^{10}\log)$$

By convention $0 \log \frac{0}{a} = 0$: zero counts shouldn't contribute; moreover $\lim_{x \to 0} x \log x = 0$

Computing quality $Q(G, \boldsymbol{D})$: an example (3)

Consider the same dataset $\boldsymbol{D}$ as before and the following graph G.



-4.384

-1.806

We first compute $-N \cdot H(G, \boldsymbol{D})$:

For V_3:

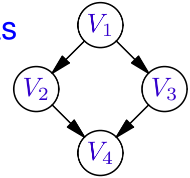
$$N(v_3 \wedge v_1) \log \frac{N(v_3 \wedge v_1)}{N(v_1)} + N(\neg v_3 \wedge v_1) \log \frac{N(\neg v_3 \wedge v_1)}{N(v_1)} +$$

$$+ N(v_3 \wedge \neg v_1) \log \frac{N(v_3 \wedge \neg v_1)}{N(\neg v_1)} + N(\neg v_3 \wedge \neg v_1) \log \frac{N(\neg v_3 \wedge \neg v_1)}{N(\neg v_1)} =$$

$$= 3 \log \frac{3}{9} + 6 \log \frac{6}{9} + 6 \log \frac{6}{6} + 0 \log \frac{0}{6} = -2.49$$

Computing quality $Q(G, \boldsymbol{D})$: an example (4)

Consider the same dataset $\boldsymbol{D}$ as before and the following graph G.



-4.384

-1.806

-2.488

We first compute $-N \cdot H(G, \boldsymbol{D})$:

For V_4:

$$N(v_4 \wedge v_2 \wedge v_3) \log \frac{N(v_4 \wedge v_2 \wedge v_3)}{N(v_2 \wedge v_3)} + N(\neg v_4 \wedge v_2 \wedge v_3) \log \frac{N(\neg v_4 \wedge v_2 \wedge v_3)}{N(v_2 \wedge v_3)}$$

$$+ N(v_4 \wedge \neg v_2 \wedge v_3) \log \frac{N(v_4 \wedge \neg v_2 \wedge v_3)}{N(\neg v_2 \wedge v_3)} + N(\neg v_4 \wedge \neg v_2 \wedge v_3) \log \frac{N(\neg v_4 \wedge \neg v_2 \wedge v_3)}{N(\neg v_2 \wedge v_3)}$$

$$+ N(v_4 \wedge v_2 \wedge \neg v_3) \log \frac{N(v_4 \wedge v_2 \wedge \neg v_3)}{N(v_2 \wedge \neg v_3)} + N(\neg v_4 \wedge v_2 \wedge \neg v_3) \log \frac{N(\neg v_4 \wedge v_2 \wedge \neg v_3)}{N(v_2 \wedge \neg v_3)}$$

$$+ N(v_4 \wedge \neg v_2 \wedge \neg v_3) \log \frac{N(v_4 \wedge \neg v_2 \wedge \neg v_3)}{N(\neg v_2 \wedge \neg v_3)} + N(\neg v_4 \wedge \neg v_2 \wedge \neg v_3) \log \frac{N(\neg v_4 \wedge \neg v_2 \wedge \neg v_3)}{N(\neg v_2 \wedge \neg v_3)}$$

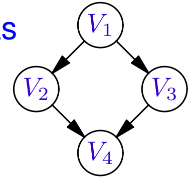
$$= 0 \log \tfrac{0}{6} + 6 \log \tfrac{6}{6} + 2 \log \tfrac{2}{3} + 1 \log \tfrac{1}{3} + 2 \log \tfrac{2}{6}$$

$$+ 4 \log \tfrac{4}{6} + \underbrace{0 \log \tfrac{0}{0} + 0 \log \tfrac{0}{0}}_{} = -2.488$$

$=$ 0 by convention ($\lim_{x \to 0} x \log \frac{x}{x} = 0$)

Computing quality $Q(G, D)$: an example (5)

Consider the same dataset $\boldsymbol{D}$ as before and the following graph G.



We first compute $-N \cdot H(G, \boldsymbol{D})$:

-4.384

-1.806

-2.488

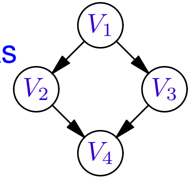
-2.488

$$-N \cdot H(G, \boldsymbol{D}) = -4.384 - 1.806 - 2.488 - 2.488 = -11.167$$

(if we again use the $^{10}\log$ for easy computation)

Computing quality $Q(G, D)$: an example (6)

Consider the same dataset $\boldsymbol{D}$ as before and the following graph G.



We have that

- $-N \cdot H(G, \boldsymbol{D}) = -11.167$
- $-\frac{1}{2}K \cdot \log N = -\frac{1}{2} \cdot (1 + 2 + 2 + 4) \cdot \log 15 = -5.292$

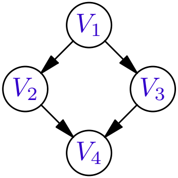
Suppose that P is a uniform distribution with $\log P(G) = C$. Then

$$Q(G, \boldsymbol{D}) = C - 16.459$$

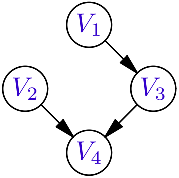
■

Comparing graphs: an example

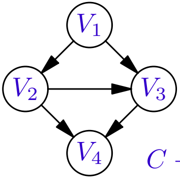
Consider the same dataset D as before. Consider the following graphs and their quality with respect to D:



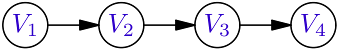
$C - 16.459$



$C - 17.324$



$C - 17.636$



$C - 16.941$

Among these graphs, the top left one best fits the data.

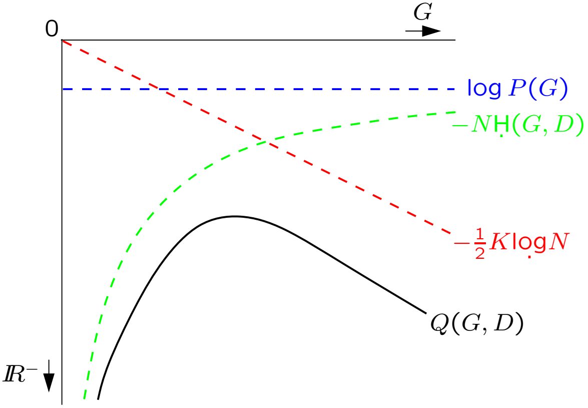
Which graph is best? The interaction among the terms

Reconsider the quality of acyclic digraph G given dataset $\boldsymbol{D}$:

$$Q(G, \boldsymbol{D}) = \log P(G) - N \cdot H(G, \boldsymbol{D}) - \frac{1}{2} K \cdot \log N$$

Assuming uniform P, the following interactions exist among the different terms of $Q(G, \boldsymbol{D})$:

NB: x-axis captures density of G



Finding the best graph: a search procedure

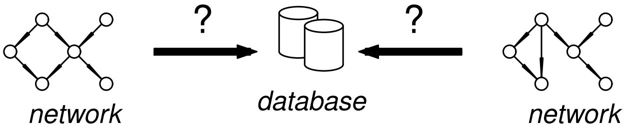
The search procedure of the learning algorithm is a heuristic for finding a DAG with the highest quality given the data.

number of nodes	number of acyclic digraphs
1	1
2	3
3	25
4	543
5	$29,281$
6	$3,781,503$
7	$1,138,779,265$
8	$783,702,329,343$
9	$1,213,442,454,842,881$
10	$4,175,098,976,430,598,143$

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.



Repeat until an increase in quality can no longer be achieved.

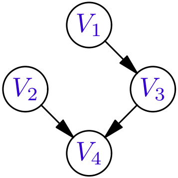
The B search heuristic

PROCEDURE CONSTRUCT-DIGRAPH ($\boldsymbol{V}$, $\boldsymbol{D}$, G):

FOR EACH $V_i \in \boldsymbol{V}$ DO
    $\boldsymbol{\rho}(V_i) := \emptyset$
OD;
REPEAT
  FOR EACH PAIR $V_i, V_j \in \boldsymbol{V}$ SUCH THAT ADDITION OF
  THE ARC (V_i, V_j) TO G DOES NOT INTRODUCE A CYCLE DO
    $\mathrm{diff}(V_i, V_j) := q(V_j, \boldsymbol{\rho}(V_j) \cup \{V_i\}, \boldsymbol{D}) - q(V_j, \boldsymbol{\rho}(V_j), \boldsymbol{D})$
  OD;
  SELECT THE PAIR $V_i, V_j \in \boldsymbol{V}$ FOR WHICH $\mathrm{diff}(V_i, V_j)$ IS MAXIMAL;
  IF $\mathrm{diff}(V_i, V_j) > 0$
  THEN $\boldsymbol{\rho}(V_j) := \boldsymbol{\rho}(V_j) \cup \{V_i\}$
  FI
UNTIL $\mathrm{diff}(V_i, V_j) \leq 0$.

B search $\text{diff}(V_i, V_j)$: an example (1)

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 $\boldsymbol{V}$, $\boldsymbol{D}$, N and G be as before.

The quality of a node $V_i \in \boldsymbol{V}_G$ given $\boldsymbol{D}$, notation: $q(V_i, \boldsymbol{\rho}(V_i), \boldsymbol{D})$, is defined as

$$q(V_i, \boldsymbol{\rho}(V_i), \boldsymbol{D}) = \sum_{c_{V_i}} \sum_{c_{\boldsymbol{\rho}(V_i)}} N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)}) \cdot \log \left(\frac{N(c_{V_i} \wedge c_{\boldsymbol{\rho}(V_i)})}{N(c_{\boldsymbol{\rho}(V_i)})} \right)$$

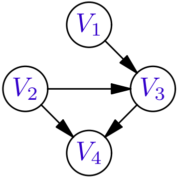
$$- \frac{1}{2} \cdot 2^{|\boldsymbol{\rho}(V_i)|} \cdot \log N$$

Lemma: (without proof)

$$Q(G, \boldsymbol{D}) = \log P(G) + \sum_{V_i \in \boldsymbol{V}_G} q(V_i, \boldsymbol{\rho}(V_i), \boldsymbol{D})$$

B search $\text{diff}(V_i, V_j)$: an example (2)

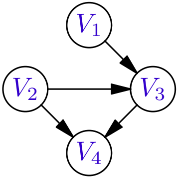
Consider the same dataset $\boldsymbol{D}$ as before and suppose (!) that the search procedure has constructed the following graph:



We consider the increase in quality for arc (V_2, V_3):

$$\text{diff}(V_2, V_3) = q(V_3, \{V_1, V_2\}, \boldsymbol{D}) - q(V_3, \{V_1\}, \boldsymbol{D})$$

B search $\mathrm{diff}(V_i, V_j)$: an example (3)



$$q(V_3, \{V_1, V_2\}, \boldsymbol{D}) =$$

$$= N(v_3 \wedge v_1 \wedge v_2) \log \frac{N(v_3 \wedge v_1 \wedge v_2)}{N(v_1 \wedge v_2)} + N(\overline{v}_3 \wedge v_1 \wedge v_2) \log \frac{N(\overline{v}_3 \wedge v_1 \wedge v_2)}{N(v_1 \wedge v_2)}$$

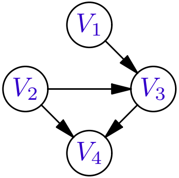
$$+ N(v_3 \wedge \overline{v}_1 \wedge v_2) \log \frac{N(v_3 \wedge \overline{v}_1 \wedge v_2)}{N(\overline{v}_1 \wedge v_2)} + N(\overline{v}_3 \wedge \overline{v}_1 \wedge v_2) \log \frac{N(\overline{v}_3 \wedge \overline{v}_1 \wedge v_2)}{N(\overline{v}_1 \wedge v_2)}$$

$$+ N(v_3 \wedge v_1 \wedge \overline{v}_2) \log \frac{N(v_3 \wedge v_1 \wedge \overline{v}_2)}{N(v_1 \wedge \overline{v}_2)} + N(\overline{v}_3 \wedge v_1 \wedge \overline{v}_2) \log \frac{N(\overline{v}_3 \wedge v_1 \wedge \overline{v}_2)}{N(v_1 \wedge \overline{v}_2)}$$

$$+ N(v_3 \wedge \overline{v}_1 \wedge \overline{v}_2) \log \frac{N(v_3 \wedge \overline{v}_1 \wedge \overline{v}_2)}{N(\overline{v}_1 \wedge \overline{v}_2)} + N(\overline{v}_3 \wedge \overline{v}_1 \wedge \overline{v}_2) \log \frac{N(\overline{v}_3 \wedge \overline{v}_1 \wedge \overline{v}_2)}{N(\overline{v}_1 \wedge \overline{v}_2)}$$

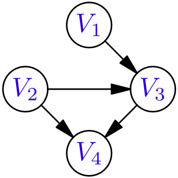
$$- \frac{1}{2} \cdot 4 \log N = -4.84$$

B search $\mathrm{diff}(V_i, V_j)$: an example (4)



$$
\begin{aligned}
& q(V_3, \{V_1\}, \boldsymbol{D}) \; = \\
& = \; N(v_3 \wedge v_1) \log \tfrac{N(v_3 \wedge v_1)}{N(v_1)} + N(\overline{v}_3 \wedge v_1) \log \tfrac{N(\overline{v}_3 \wedge v_1)}{N(v_1)} \\
& + N(v_3 \wedge \overline{v}_1) \log \tfrac{N(v_3 \wedge \overline{v}_1)}{N(\overline{v}_1)} + N(\overline{v}_3 \wedge \overline{v}_1) \log \tfrac{N(\overline{v}_3 \wedge \overline{v}_1)}{N(\overline{v}_1)} \\
& - \tfrac{1}{2} \cdot 2 \log N = -3.66
\end{aligned}
$$

B search $\text{diff}(V_i, V_j)$: an example (5)



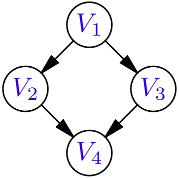
We consider the increase in quality for arc (V_2, V_3):

$$\text{diff}(V_2, V_3) = q(V_3, \{V_1, V_2\}, \boldsymbol{D}) - q(V_3, \{V_1\}, \boldsymbol{D})$$

$$= -4.84 - -3.66 = -1.18$$

The increase in quality for arc (V_2, V_3) is negative; the arc will therefore not be selected by the search procedure.

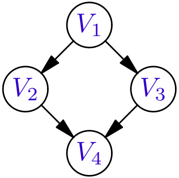
B search $\mathrm{diff}(V_i, V_j)$: an example (6)



We consider the increase in quality for the arc (V_1, V_2):

$$\mathrm{diff}(V_1, V_2) = q(V_2, \{V_1\}, \boldsymbol{D}) - q(V_2, \emptyset, \boldsymbol{D})$$

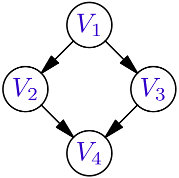
B search $\mathrm{diff}(V_i, V_j)$: an example (7)



$$
\begin{aligned}
q(V_2, \{V_1\}, \boldsymbol{D}) \ = \\
= \ N(v_2 \wedge v_1) \log \tfrac{N(v_2 \wedge v_1)}{N(v_1)} + N(\overline{v}_2 \wedge v_1) \log \tfrac{N(\overline{v}_2 \wedge v_1)}{N(v_1)} \\
+ N(v_2 \wedge \overline{v}_1) \log \tfrac{N(v_2 \wedge \overline{v}_1)}{N(\overline{v}_1)} + N(\overline{v}_2 \wedge \overline{v}_1) \log \tfrac{N(\overline{v}_2 \wedge \overline{v}_1)}{N(\overline{v}_1)} \\
- \tfrac{1}{2} \cdot 2 \cdot \log N = -2.98
\end{aligned}
$$

$$
\begin{aligned}
q(V_2, \emptyset, \boldsymbol{D}) \ = \\
= N(v_2) \log \tfrac{N(v_2)}{N} + N(\overline{v}_2) \log \tfrac{N(\overline{v}_2)}{N} - \tfrac{1}{2} \cdot \log N \\
= \ -3.85
\end{aligned}
$$

B search $\text{diff}(V_i, V_j)$: an example (8)



We consider the increase in quality for the arc (V_1, V_2):

$$\text{diff}(V_1, V_2) = q(V_2, \{V_1\}, \boldsymbol{D}) - q(V_2, \emptyset, \boldsymbol{D})$$

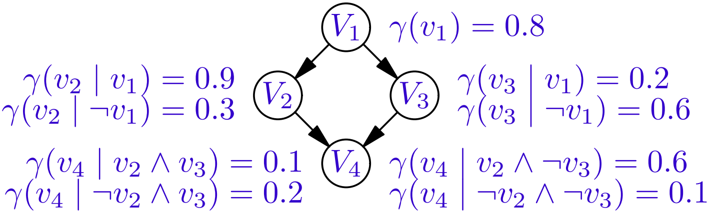
$$= -2.98 - -3.85 = 0.87$$

The increase in quality for arc (V_1, V_2) is positive; the arc may be selected by the search procedure, but only if it has the largest increase of all options.

Evaluation

Is the presented metric algorithm any good?

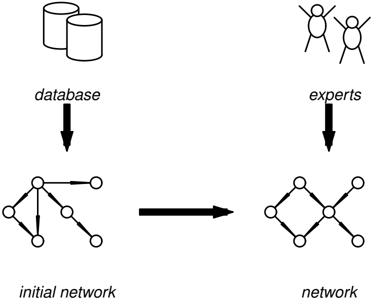
- our example dataset D was generated from the following network:



- the MDL score is asymptotically correct: for best MDL-scoring $\mathcal{B}$, $\Pr_{\mathcal{B}}$ will be arbitrarily close to the sampled distribution, given sufficient independent samples.

Some remarks (1)

- 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 $\Pr(C \mid \boldsymbol{F})$ rather than $\Pr(C\boldsymbol{F})$) but expensive.

Parameter learning

Many structure learning algorithms learn a whole BN, including network-parameters.

Given a network structure, different algorithms exist for learning model-parameters θ from data.

- frequentistic approaches: θ is a fixed unknown constant
  MLE $\hat{\theta} = \arg\max_\theta \log P(\boldsymbol{D} \mid \theta)$
- Bayesian approach: θ is a random variable
  Full: $P(\theta \mid \boldsymbol{D}) = \frac{P(\boldsymbol{D}|\theta) \cdot P(\theta)}{P(\boldsymbol{D})}$
  MAP $\hat{\theta} = \arg\max_\theta P(\boldsymbol{D} \mid \theta) \cdot P(\theta)$

Suitability of these approaches depends on the available data.

Data as a source of probabilistic information

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;
- …

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.

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:

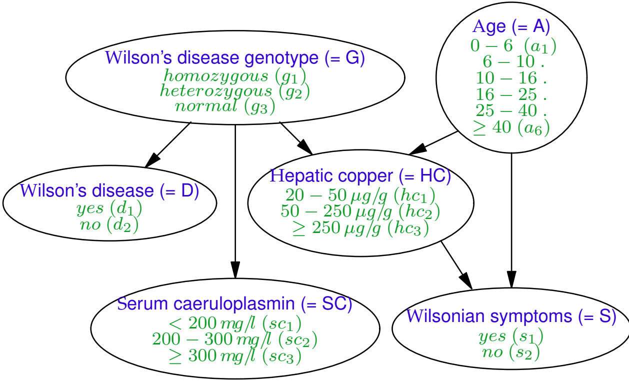
- changes to the
  - definitions of the variables and values;
  - graphical structure;

  may help reduce the number of required probabilities;
- the use of
  - domain models;
  - canonical models;

  may help reduce the number of probabilities to be assessed.

The use of domain models: an example

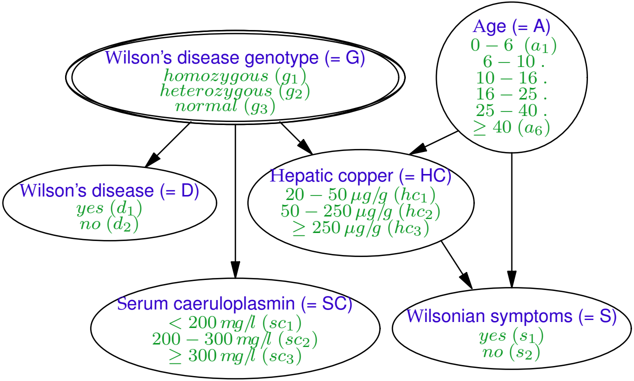
Consider building a Bayesian network for Wilson's disease, a recessively inherited disease of the liver:



From the disease being recessively inherited, we have for the variable '*Wilson's disease*' that

$$\gamma(d_1 \mid g_1) = 1 \qquad \gamma(d_2 \mid g_1) = 0$$
$$\gamma(d_1 \mid g_2) = 0 \qquad \gamma(d_2 \mid g_2) = 1$$
$$\gamma(d_1 \mid g_3) = 0 \qquad \gamma(d_2 \mid g_3) = 1$$

The use of domain models: the example continued



Consider the node *'Wilson's disease genotype'*. By Mendel's law:

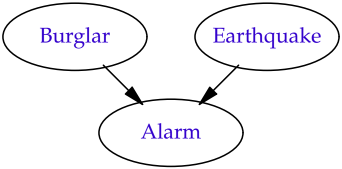
$$\Pr(g_1) = \Pr(g_1)\cdot\Pr(g_1) + \frac{1}{2}\cdot 2\cdot\Pr(g_1)\cdot\Pr(g_2) + \frac{1}{4}\cdot\Pr(g_2)\cdot\Pr(g_2)$$

With $\Pr(g_1) = \Pr(d_1) = 0.005$, we now find

$$\gamma(g_1) = 0.005, \quad \gamma(g_2) = 0.131, \quad \text{and} \quad \gamma(g_3) = 0.864$$

The use of canonical models

Consider the following causal mechanism:



The node *Alarm* requires the following probabilities:

$\gamma(alarm \mid \neg burglar \wedge \neg earthq.)$ $\gamma(alarm \mid burglar \wedge \neg earthq.)$

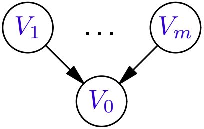
$\gamma(alarm \mid \neg burglar \wedge earthq.)$ $\gamma(alarm \mid burglar \wedge earthq.)$

The underlying mechanisms that cause the alarm have 'nothing to do with each other' $\rightarrow$ hard to assess probabilities in a straightforward manner.

A canonical approach requires just two assessments and provides parameterized rules for computing the other ones.

Disjunctive interaction, informally

Consider the following causal mechanism:



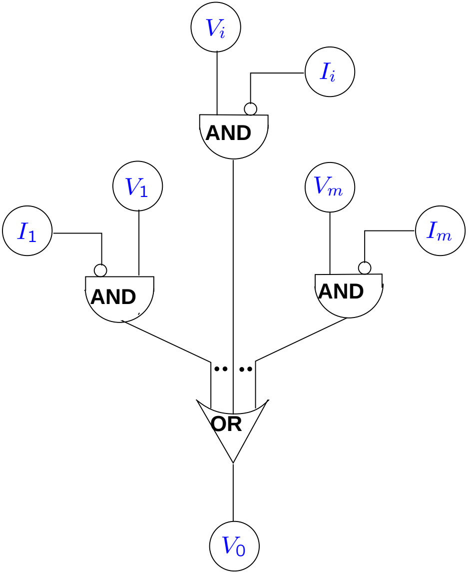
The variables $V_1, \ldots, V_m$, $m \geq 2$, exhibit a disjunctive interaction with respect to variable V_0 if, for $i = 1, \ldots, m$, we have that:

- $V_i =$ *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 canonical model that describes a causal mechanism with a disjunctive interaction is called a noisy-or gate.

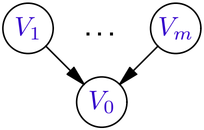
Disjunctive interaction, continued

The semantics of a disjunctive interaction can be depicted as



Disjunctive interaction, more formally

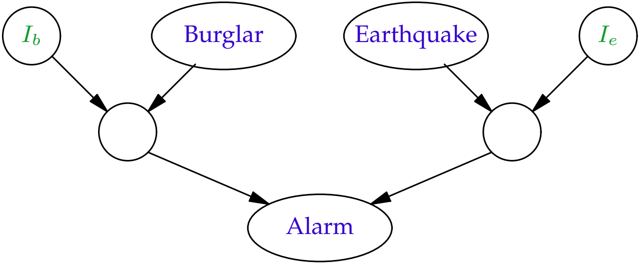
Consider the following causal mechanism:



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*, . . . , $V_m =$ *true*, that is, $\Pr(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_m) = 0$
- exception independence:
  1) for each V_i, an inhibitor I_i can be defined such that
     $\Pr(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_{i-1} \wedge (v_i \wedge i_i) \wedge \neg v_{i+1} \wedge \ldots \wedge \neg v_m) = 0$
     $\Pr(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_{i-1} \wedge (v_i \wedge \neg i_i) \wedge \neg v_{i+1} \wedge \ldots \wedge \neg v_m) = 1$
  2) the inhibitors I_i are mutually independent.

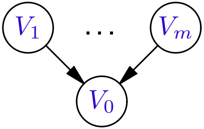
An example



- 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?

Probabilities for the noisy-or gate

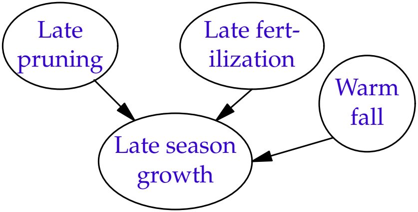


For the variable V_0, the noisy-or gate specifies:

- using the property of accountability:
  $\gamma(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_m) = 0$
- using the property of exception independence:
  - $\gamma(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_{i-1} \wedge v_i \wedge \neg v_{i+1} \wedge \ldots \wedge \neg v_m) = 1 - q_i^a$ where $\Pr(i_i) = q_i^a$ for inhibitor I_i of V_i;
  - for each configuration $\mathbf{c}$ of $\{V_1, \ldots, V_m\}$ with $T_{\mathbf{c}} = \{i \mid \mathbf{c}$ contains $v_i\}$, $T_{\mathbf{c}} \neq \emptyset$: $\quad \gamma(v_0 \mid \mathbf{c}) = 1 - \prod_{i \,\in\, T_{\mathbf{c}}} q_i^a$

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:

$$\gamma(lsg \mid lp \wedge \neg lf \wedge \neg wf) = 0.8 \qquad \Pr(i_{lp}) = 0.2$$

$$\gamma(lsg \mid \neg lp \wedge lf \wedge \neg wf) = 0.8 \Longrightarrow \Pr(i_{lf}) = 0.2$$

$$\gamma(lsg \mid \neg lp \wedge \neg lf \wedge wf) = 0.6 \qquad \Pr(i_{wf}) = 0.4$$

An example noisy-or gate

$$\begin{aligned} \gamma(lsg \mid lp \wedge \neg lf \wedge \neg wf) &= 0.8 & \Pr(i_{lp}) &= 0.2 \\ \gamma(lsg \mid \neg lp \wedge lf \wedge \neg wf) &= 0.8 \Longrightarrow & \Pr(i_{lf}) &= 0.2 \\ \gamma(lsg \mid \neg lp \wedge \neg lf \wedge wf) &= 0.6 & \Pr(i_{wf}) &= 0.4 \end{aligned}$$

We then compute, for example,

$$\gamma(lsg \mid lp \wedge lf \wedge \neg wf) = 1 - \Pr(i_{lp}) \cdot \Pr(i_{lf}) = 1 - 0.2 \cdot 0.2 = 0.96$$

Late pruning		*false*		*true*	
Late fertilisation		*false*	*true*	*false*	*true*
Warm fall	*false*	0	0.8	0.8	0.96
	true	0.6	0.92	0.92	0.98

The example continued

Now compare:

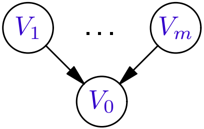
- the probabilities obtained from the noisy-or gate:

Late pruning	false		true	
Late fertilisation	false	true	false	true
Warm fall false	0	0.8	0.8	0.96
Warm fall true	0.6	0.92	0.92	0.98

- the probabilities assessed by domain experts:

Late pruning	false		true	
Late fertilisation	false	true	false	true
Warm fall false	0.1	0.8	0.8	0.9
Warm fall true	0.6	0.9	0.9	1.0

If accountability is violated



Suppose that exception independence holds, but accountability does not, that is,

$$\Pr(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_m) = p \text{ with } p > 0$$

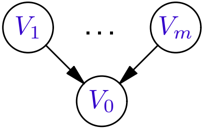
- the noisy-or gate can be applied after including an additional parent V_{m+1} of V_0 with

$$\gamma(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_m \wedge \neg v_{m+1}) = 0$$
$$\gamma(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_m \wedge v_{m+1}) = p$$

- the leaky noisy-or gate can be used.

The leaky noisy-or gate

Consider the following causal mechanism with exception independence:



Suppose that $\Pr(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_m) = p$, where $p = 1 - q_0 > 0$ is the leak probability. The leaky noisy-or gate specifies for V_0:

- $\gamma(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_m) = p$;
- $\gamma(v_0 \mid \neg v_1 \wedge \ldots \wedge \neg v_{i-1} \wedge v_i \wedge \neg v_{i+1} \wedge \ldots \wedge \neg v_m) = 1 - q_i^l$ where $\Pr(i_i) = q_i^l = q_0 \cdot q_i^a$ for inhibitor I_i of V_i;
- for each configuration $\mathbf{c}$ with $T_{\mathbf{c}} \neq \emptyset$, we have

$$\gamma(v_0 \mid \mathbf{c}) = 1 - q_0 \cdot \prod_{i \,\in\, T_{\mathbf{c}}} q_i^a = 1 - q_0 \cdot \prod_{i \,\in\, T_{\mathbf{c}}} \left(\frac{q_i^l}{q_0} \right)$$

For variable V_0 only $m + 1$ probabilities need to be assessed.

An example leaky noisy-or gate

Reconsider the late-pruning example:

$$\begin{array}{lcl}\gamma(lsg \mid lp \wedge \neg lf \wedge \neg wf) = 0.8 & & \Pr(i_{lp}) = 0.2 \\ \gamma(lsg \mid \neg lp \wedge lf \wedge \neg wf) = 0.8 & \Longrightarrow & \Pr(i_{lf}) = 0.2 \\ \gamma(lsg \mid \neg lp \wedge \neg lf \wedge wf) = 0.6 & & \Pr(i_{wf}) = 0.4\end{array}$$

With a leak probability $\Pr(lsg \mid \neg lp \wedge \neg lf \wedge \neg wf) = 0.1$, giving $q_0 = 0.9$, we compute

Late pruning		*false*		*true*	
Late fertilisation		*false*	*true*	*false*	*true*
Warm fall	*false*	0.1	0.8	0.8	0.96
	true	0.6	0.91	0.91	0.98

Subjective probabilities

Probability assessment often requires the help of domain experts $\rightarrow$ assessments are based upon personal knowledge and experience, i.e. subjective.

This can result in a number of problems:

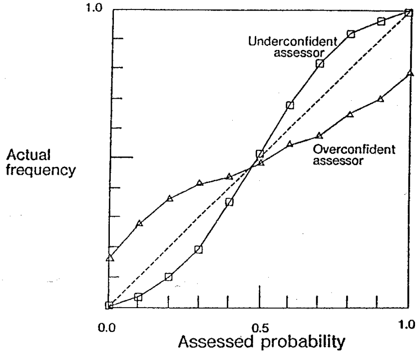
- assessments are incoherent[6]:
  - $\Pr(a) < \Pr(a \wedge b)$;
  - $\Pr(a) > \Pr(b)$ and yet $\Pr(a \mid b) < \Pr(b \mid a)$.
- assessments are biased as a result of various psychological factors, and therefore uncalibrated[7];
- the domain expert is not capable of expressing his knowledge and experience in terms of numbers.

---

[6]assessments do not adhere to the postulates of probability theory

[7]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:

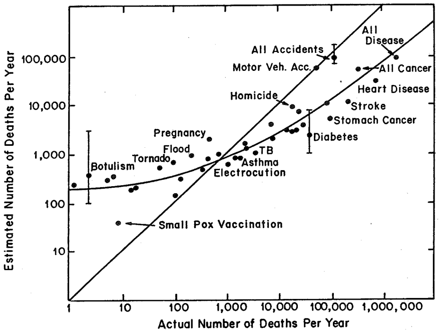
- prior probabilities, or base rates, are insufficiently taken into account;
- assessments are based upon insufficient samples;
- weights of the characteristics of the stereotype outcome are insufficiently taken into consideration;
- ...

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

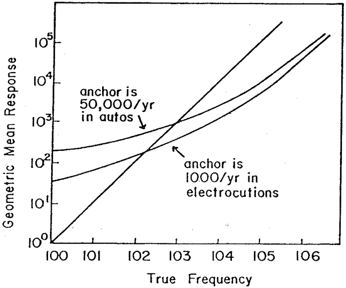


Pitfalls — cntd.

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



- . . .

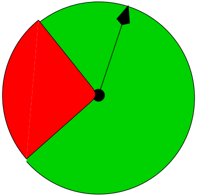
Probability assessment tools

For eliciting probabilities from experts, various tools are available from the field of decision analysis:

- probability wheels;
- betting models;
- lottery models;
- probability scales.

Probability wheels

A probability wheel is composed of two coloured faces and a hand:

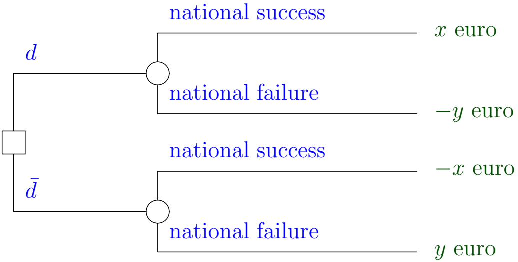


The expert is asked to adjust the area of the red face so that the probability of the hand stopping there, equals the probability of interest.

Betting 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 bets:



- if the expert is indifferent between d and $\bar{d}$, then

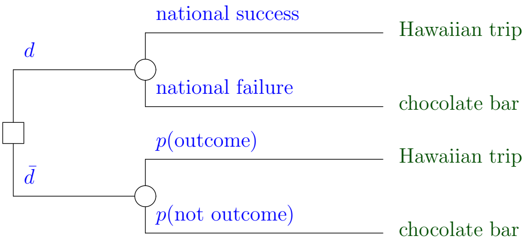
$$x \cdot \Pr(n) - y \cdot (1 - \Pr(n)) = y \cdot (1 - \Pr(n)) - x \cdot \Pr(n)$$

from which we find $\Pr(n) = \dfrac{y}{x + y}$.

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:



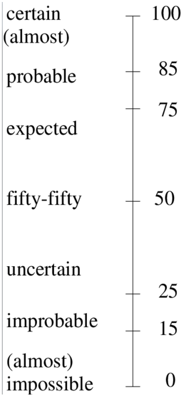
- if the expert is indifferent between d and $\bar{d}$, then $\Pr(n) = p(\text{outcome})$.

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.

Conjunctivitis | Mucositis (1)

Consider a pig *without an infection of the mucous*.
How likely is it that this pig shows a *conjunctivitis* ?



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.