

# Probabilistic Reasoning: Graphical Models

Christian Borgelt

Intelligent Data Analysis and Graphical Models Research Unit  
European Center for Soft Computing  
c/ Gonzalo Gutiérrez Quirós s/n, 33600 Mieres (Asturias), Spain

`christian.borgelt@softcomputing.es`

`http://www.softcomputing.es/`

`http://www.borgelt.net/`

# Overview

- **Graphical Models: Core Ideas and Notions**
- **A Simple Example: How does it work in principle?**
- **Conditional Independence Graphs**
  - conditional independence and the graphoid axioms
  - separation in (directed and undirected) graphs
  - decomposition/factorization of distributions
- **Evidence Propagation in Graphical Models**
- **Building Graphical Models**
- **Learning Graphical Models from Data**
  - quantitative (parameter) and qualitative (structure) learning
  - evaluation measures and search methods
  - learning by measuring the strength of marginal dependences
  - learning by conditional independence tests
- **Summary**

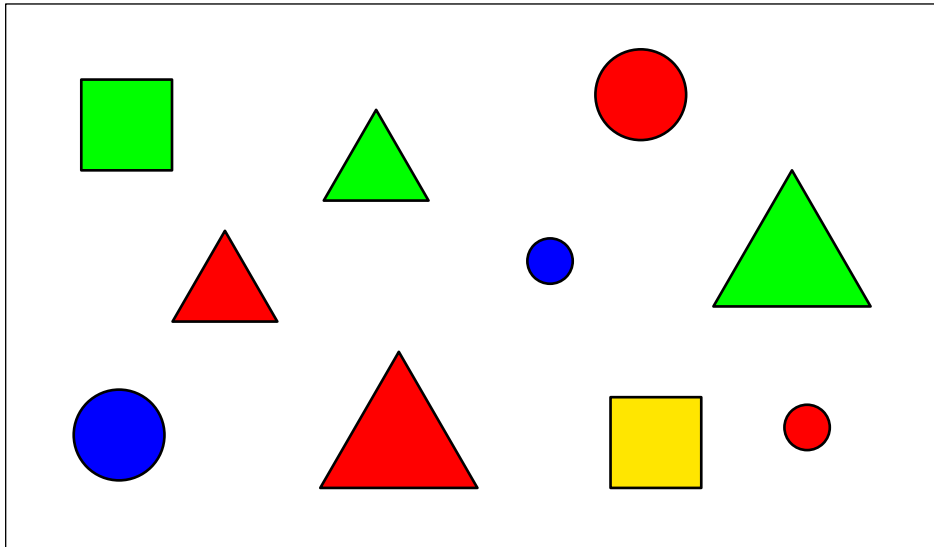
# Graphical Models: Core Ideas and Notions

- **Decomposition:** Under certain conditions a distribution  $\delta$  (e.g. a probability distribution) on a multi-dimensional domain, which encodes *prior* or *generic knowledge* about this domain, can be decomposed into a set  $\{\delta_1, \dots, \delta_s\}$  of (usually overlapping) distributions on lower-dimensional subspaces.
- **Simplified Reasoning:** If such a decomposition is possible, it is sufficient to know the distributions on the subspaces to draw all inferences in the domain under consideration that can be drawn using the original distribution  $\delta$ .
- Such a decomposition can nicely be represented as a **graph** (in the sense of graph theory), and therefore it is called a **Graphical Model**.
- The graphical representation
  - encodes **conditional independences** that hold in the distribution,
  - describes a **factorization** of the probability distribution,
  - indicates how **evidence propagation** has to be carried out.

# A Simple Example: The Relational Case

# A Simple Example

## Example Domain

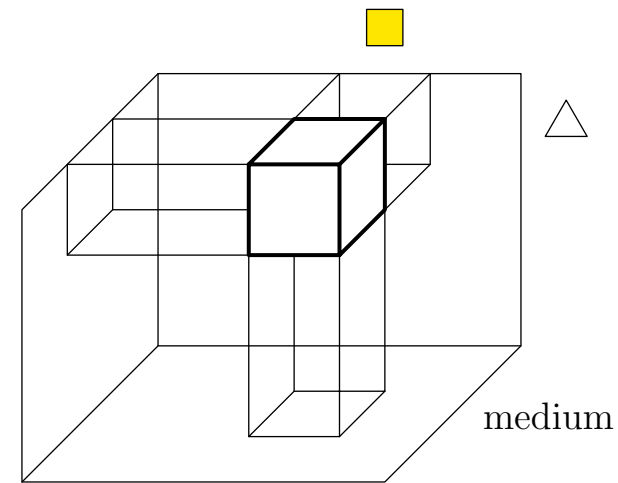
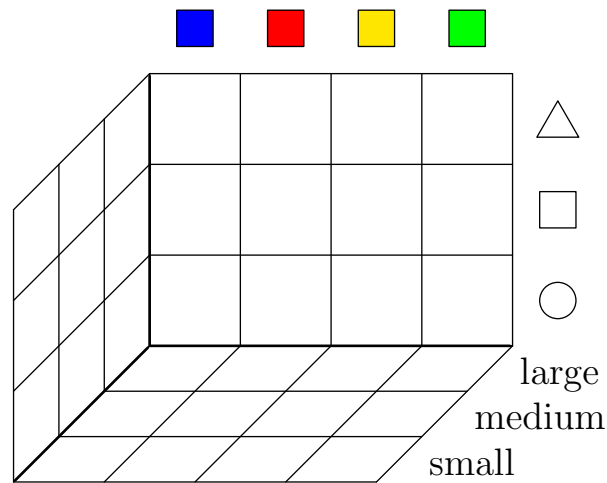


## Relation

color	shape	size
■	○	small
■	○	medium
■	○	small
■	○	medium
■	△	medium
■	△	large
■	□	medium
■	□	medium
■	△	medium
■	△	large

- 10 simple geometrical objects, 3 attributes.
- One object is chosen at random and examined.
- Inferences are drawn about the unobserved attributes.

# The Reasoning Space



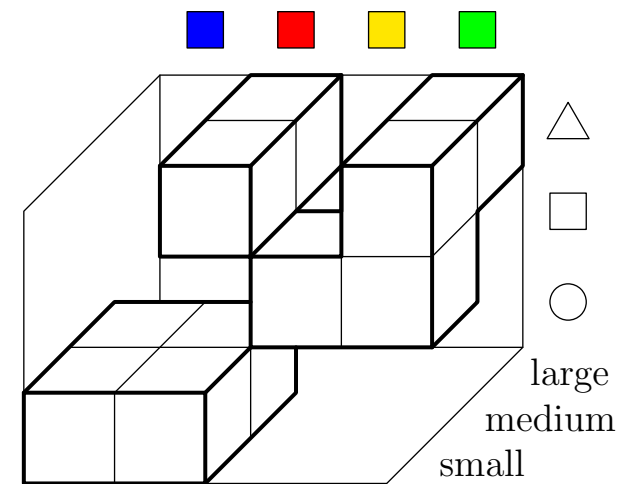
- The reasoning space consists of a finite set  $\Omega$  of states.
- The states are described by a set of  $n$  attributes  $A_i$ ,  $i = 1, \dots, n$ , whose domains  $\{a_1^{(i)}, \dots, a_{n_i}^{(i)}\}$  can be seen as sets of propositions or events.
- The events in a domain are mutually exclusive and exhaustive.
- The reasoning space is assumed to contain the true, but unknown state  $\omega_0$ .
- Technically, the attributes  $A_i$  are random variables.

# The Relation in the Reasoning Space

## Relation

color	shape	size
■	○	small
■	○	medium
■	○	small
■	○	medium
■	△	medium
■	△	large
■	□	medium
■	□	medium
■	△	medium
■	△	large

## Relation in the Reasoning Space

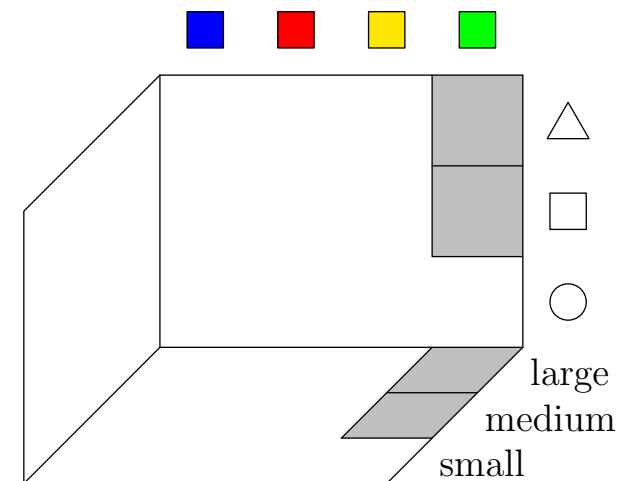
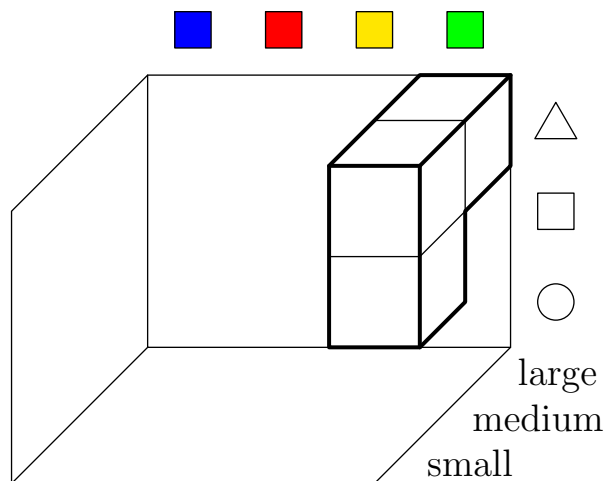


Each cube represents one tuple.

- The spatial representation helps to understand the decomposition mechanism.
- However, in practice graphical models refer to (many) more than three attributes.

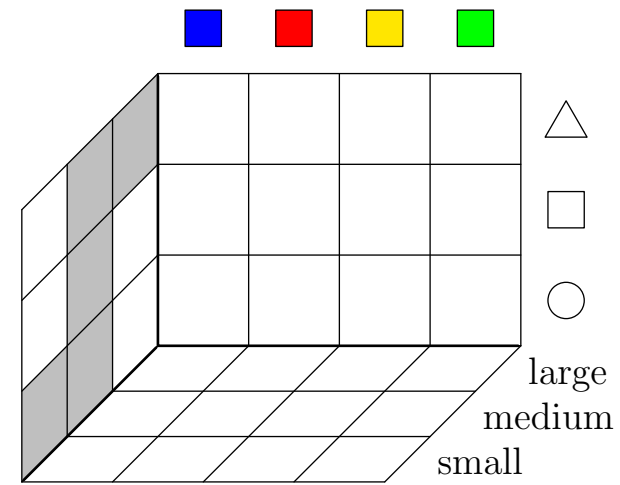
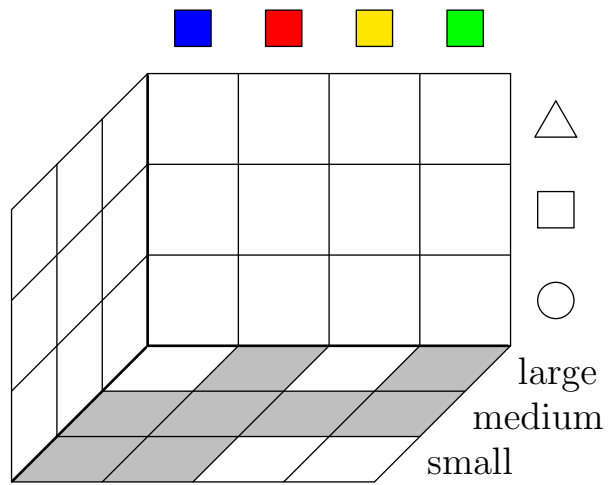
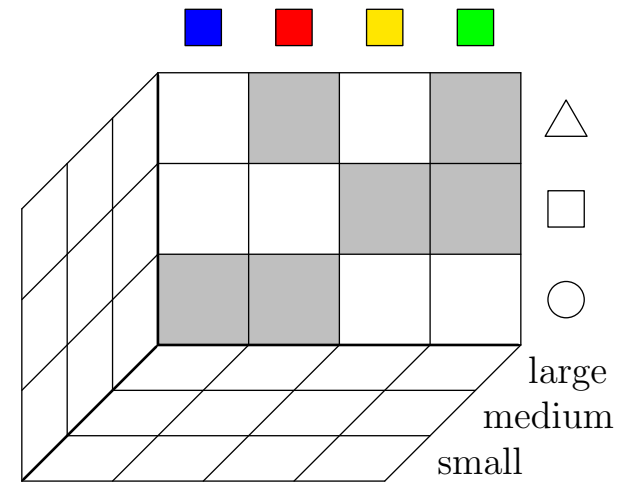
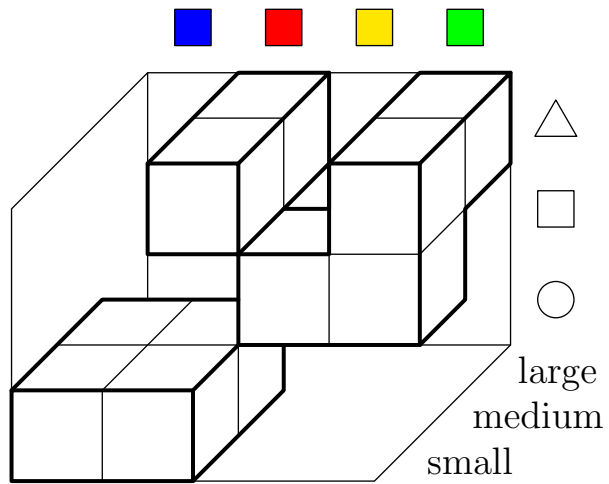
# Reasoning

- Let it be known (e.g. from an observation) that the given object is green. This information considerably reduces the space of possible value combinations.
- From the prior knowledge it follows that the given object must be
  - either a triangle or a square and
  - either medium or large.

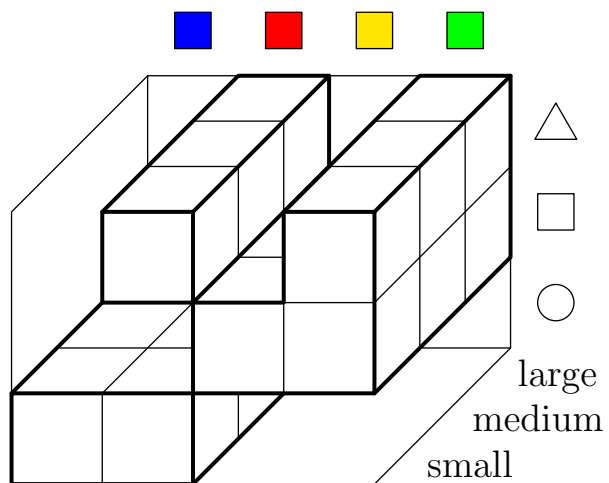




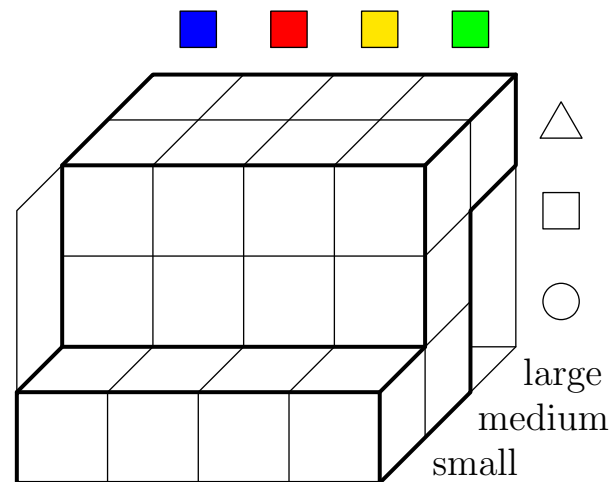
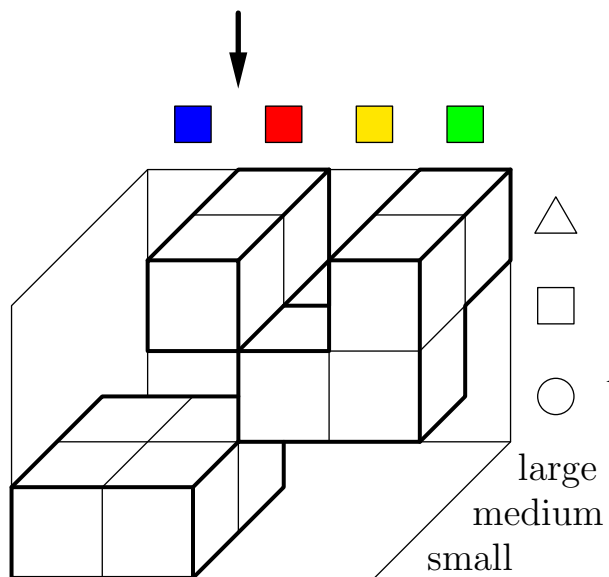
# Prior Knowledge and Its Projections



# Cylindrical Extensions and Their Intersection

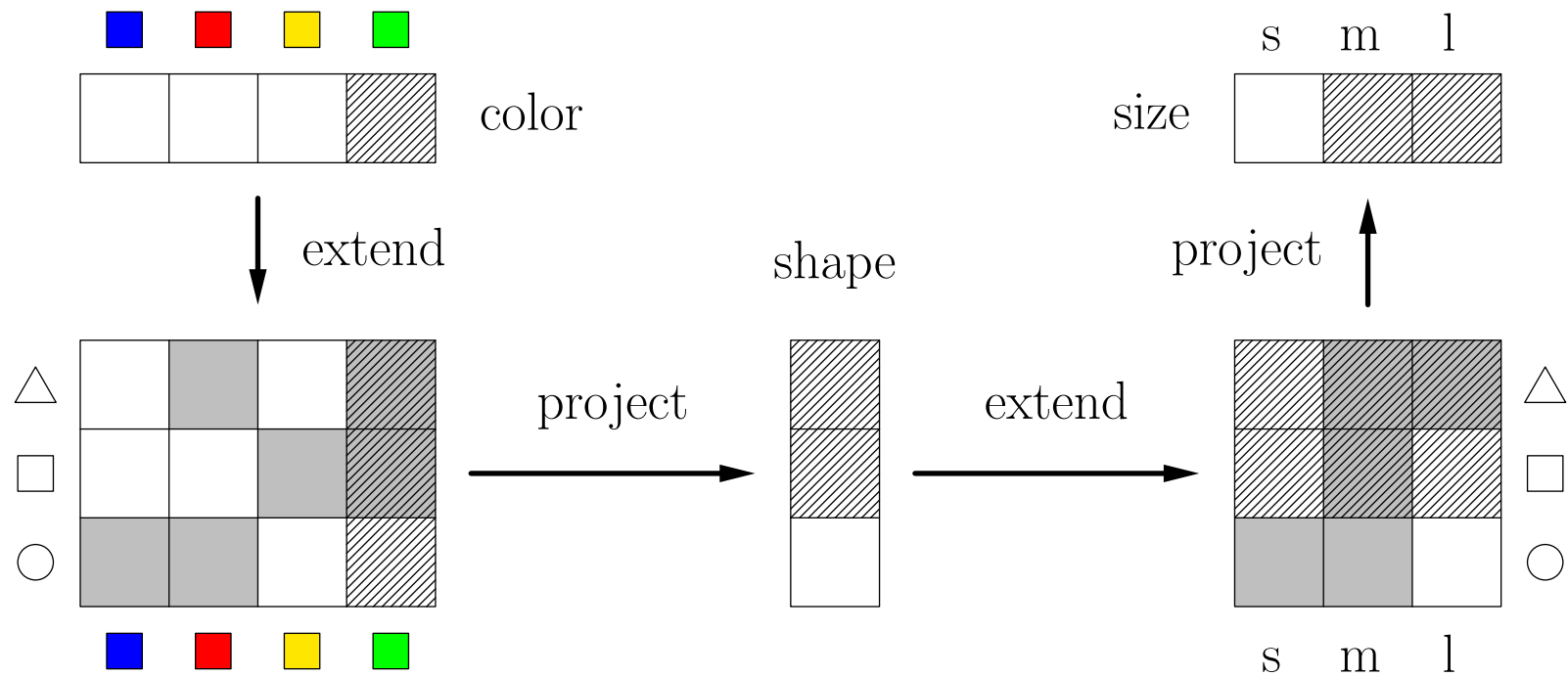


Intersecting the cylindrical extensions of the projection to the subspace spanned by color and shape and of the projection to the subspace spanned by shape and size yields the original three-dimensional relation.



# Reasoning with Projections

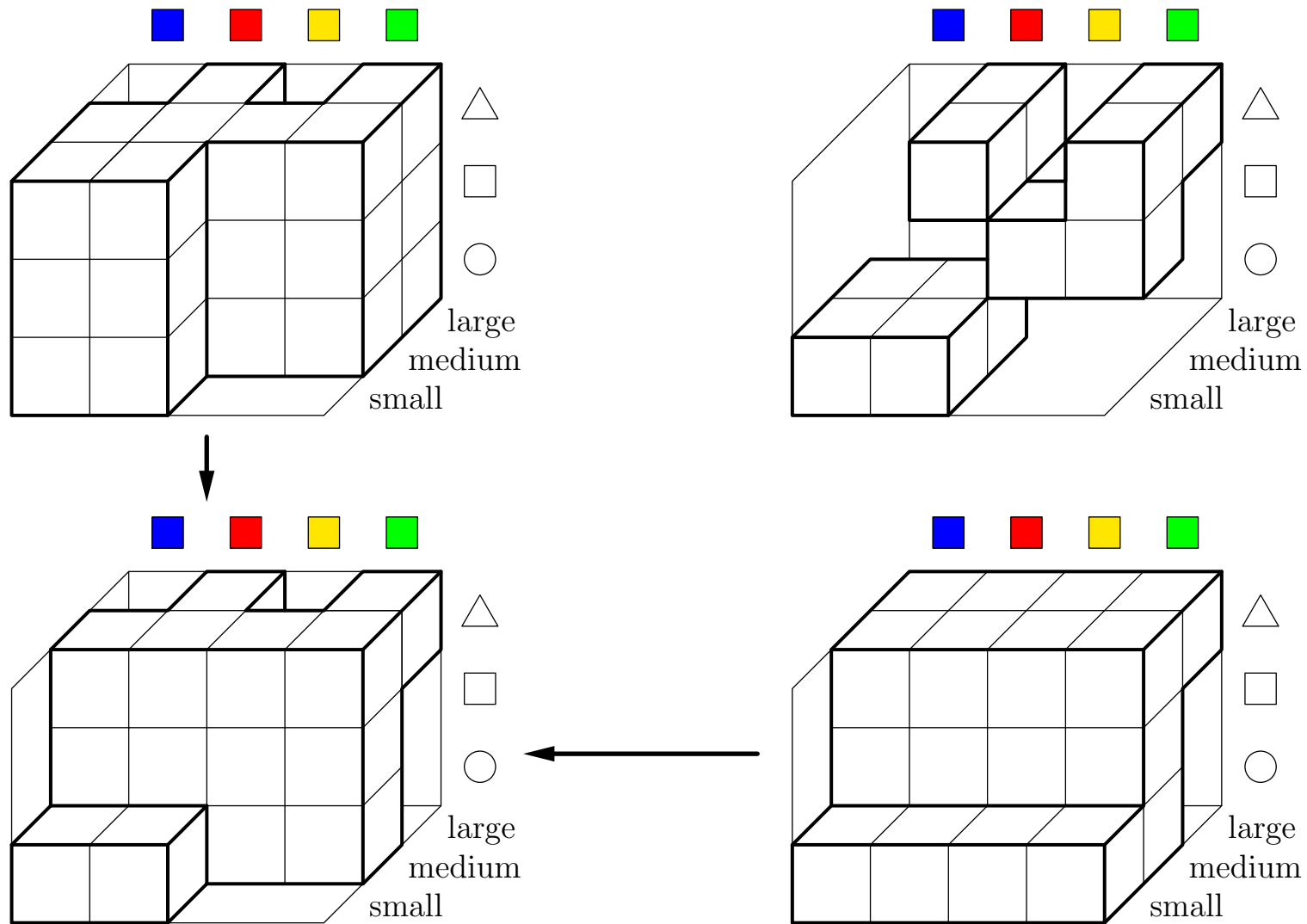
The reasoning result can be obtained using only the projections to the subspaces without reconstructing the original three-dimensional relation:



This justifies a graph representation:

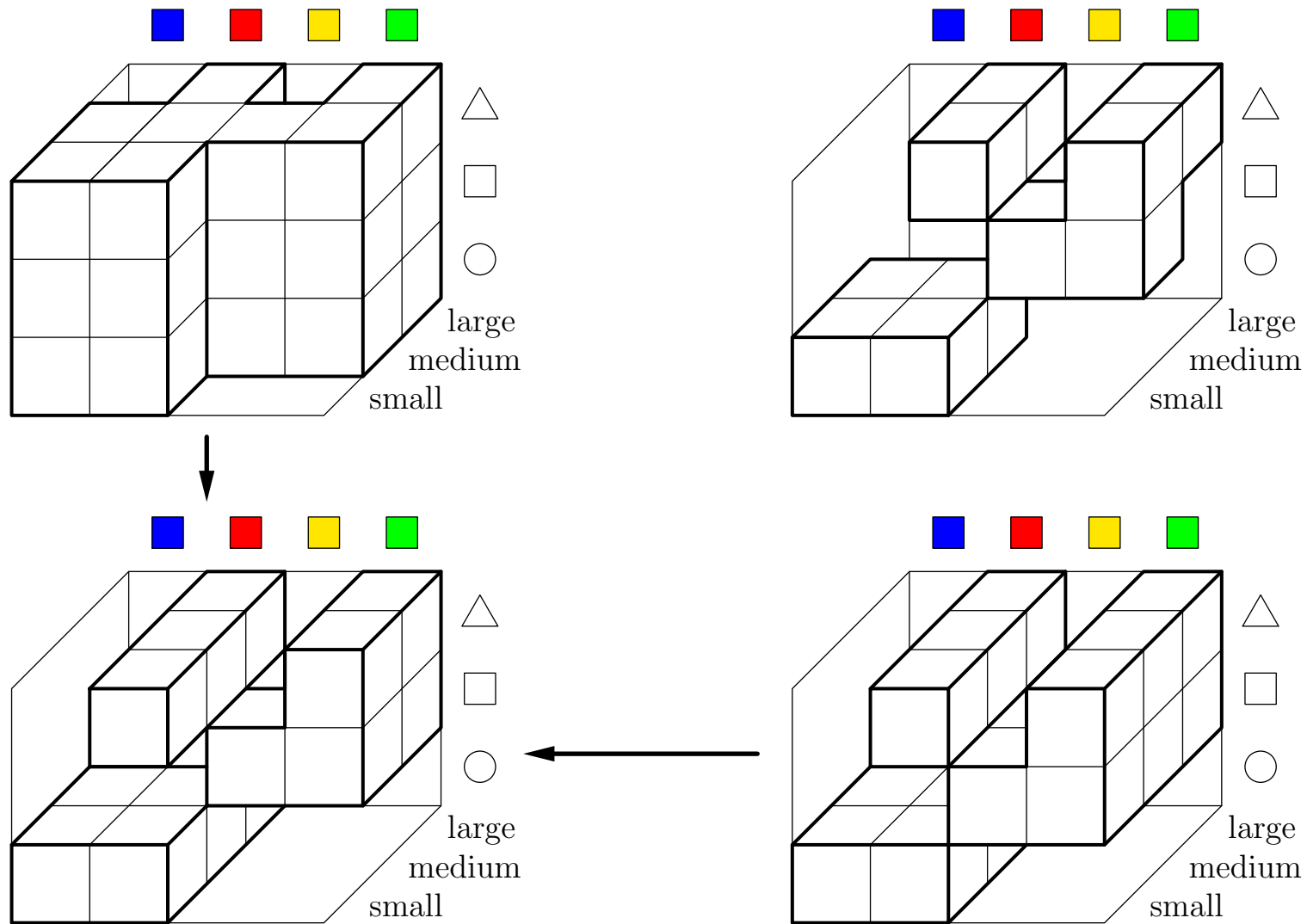


# Using other Projections 1



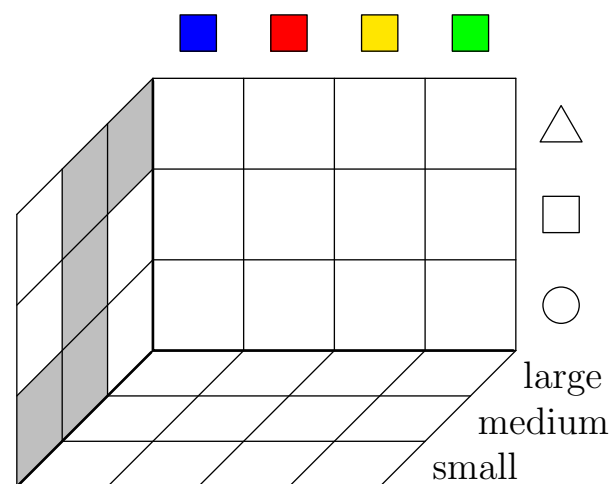
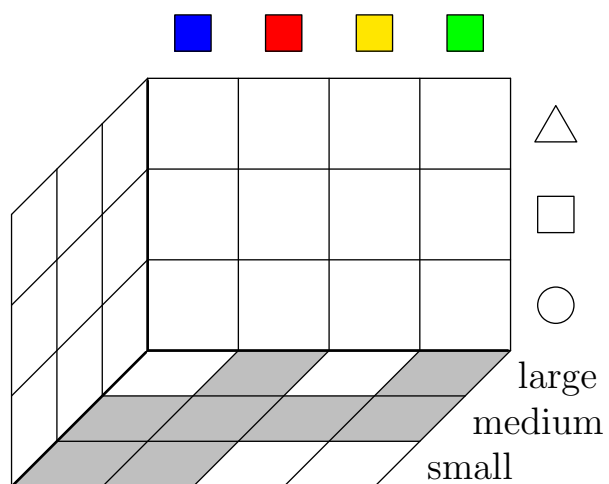
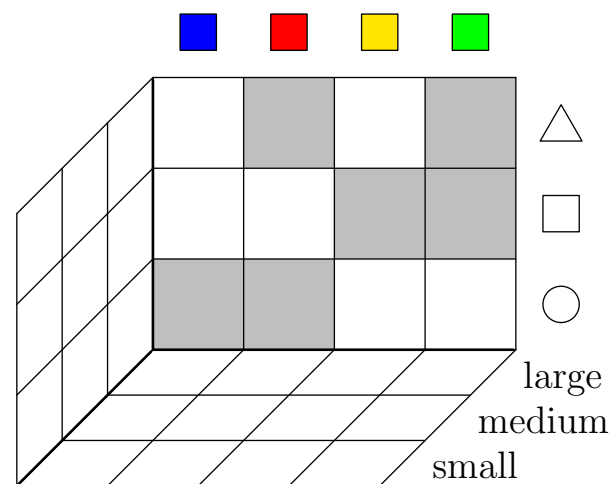
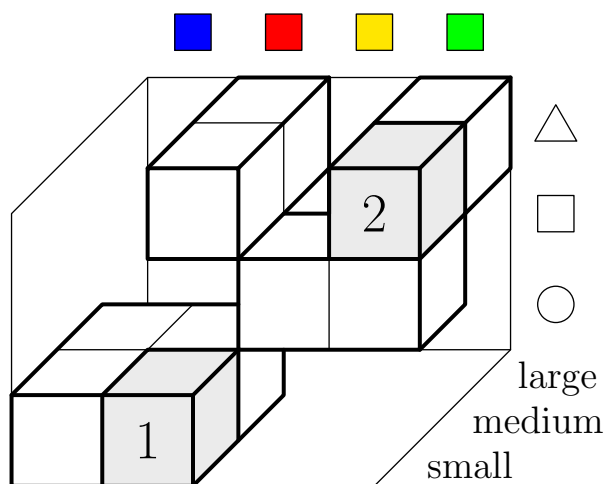
- This choice of subspaces does not yield a decomposition.

# Using other Projections 2



- This choice of subspaces does not yield a decomposition.

# Is Decomposition Always Possible?



- A modified relation (without tuples 1 or 2) may not possess a decomposition.

# Relational Graphical Models: Formalization

# Possibility-Based Formalization

**Definition:** Let  $\Omega$  be a (finite) sample space.

A **discrete possibility measure**  $R$  on  $\Omega$  is a function  $R : 2^\Omega \rightarrow \{0, 1\}$  satisfying

1.  $R(\emptyset) = 0$  and
  2.  $\forall E_1, E_2 \subseteq \Omega : R(E_1 \cup E_2) = \max\{R(E_1), R(E_2)\}$ .
- Similar to Kolmogorov's axioms of probability theory.
  - If an event  $E$  can occur (if it is possible), then  $R(E) = 1$ , otherwise (if  $E$  cannot occur/is impossible) then  $R(E) = 0$ .
  - $R(\Omega) = 1$  is not required, because this would exclude the empty relation.
  - From the axioms it follows  $R(E_1 \cap E_2) \leq \min\{R(E_1), R(E_2)\}$ .
  - Attributes are introduced as random variables (as in probability theory).
  - $R(A = a)$  and  $R(a)$  are abbreviations of  $R(\{\omega \mid A(\omega) = a\})$ .



# Possibility-Based Formalization (continued)

**Definition:** Let  $U = \{A_1, \dots, A_n\}$  be a set of attributes defined on a (finite) sample space  $\Omega$  with respective domains  $\text{dom}(A_i)$ ,  $i = 1, \dots, n$ . A **relation**  $r_U$  over  $U$  is the restriction of a discrete possibility measure  $R$  on  $\Omega$  to the set of all events that can be defined by stating values for all attributes in  $U$ . That is,  $r_U = R|_{\mathcal{E}_U}$ , where

$$\begin{aligned}\mathcal{E}_U &= \left\{ E \in 2^\Omega \mid \exists a_1 \in \text{dom}(A_1) : \dots \exists a_n \in \text{dom}(A_n) : \right. \\ &\quad \left. E \cong \bigwedge_{A_j \in U} A_j = a_j \right\} \\ &= \left\{ E \in 2^\Omega \mid \exists a_1 \in \text{dom}(A_1) : \dots \exists a_n \in \text{dom}(A_n) : \right. \\ &\quad \left. E = \left\{ \omega \in \Omega \mid \bigwedge_{A_j \in U} A_j(\omega) = a_j \right\} \right\}.\end{aligned}$$

- A relation corresponds to the notion of a probability distribution.
- Advantage of this formalization: No index transformation functions are needed for projections, there are just fewer terms in the conjunctions.

# Possibility-Based Formalization (continued)

**Definition:** Let  $U = \{A_1, \dots, A_n\}$  be a set of attributes and  $r_U$  a relation over  $U$ . Furthermore, let  $\mathcal{M} = \{M_1, \dots, M_m\} \subseteq 2^U$  be a set of nonempty (but not necessarily disjoint) subsets of  $U$  satisfying

$$\bigcup_{M \in \mathcal{M}} M = U.$$

$r_U$  is called **decomposable** w.r.t.  $\mathcal{M}$  iff

$$\forall a_1 \in \text{dom}(A_1) : \dots \forall a_n \in \text{dom}(A_n) : \\ r_U \left( \bigwedge_{A_i \in U} A_i = a_i \right) = \min_{M \in \mathcal{M}} \left\{ r_M \left( \bigwedge_{A_i \in M} A_i = a_i \right) \right\}.$$

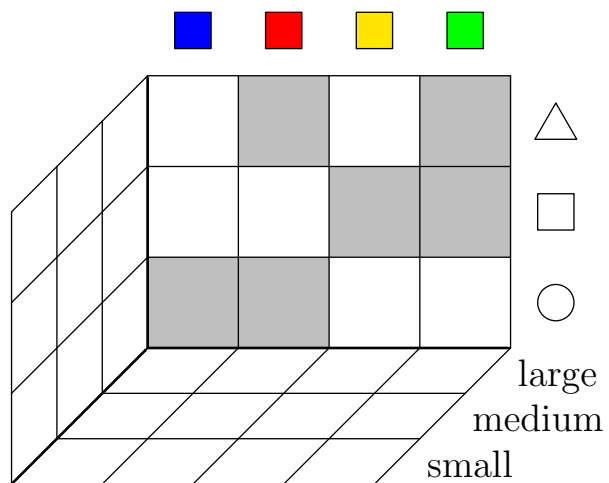
If  $r_U$  is decomposable w.r.t.  $\mathcal{M}$ , the set of relations

$$\mathcal{R}_{\mathcal{M}} = \{r_{M_1}, \dots, r_{M_m}\} = \{r_M \mid M \in \mathcal{M}\}$$

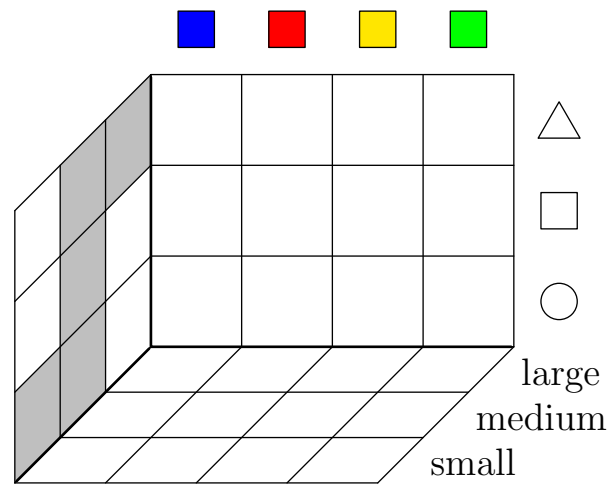
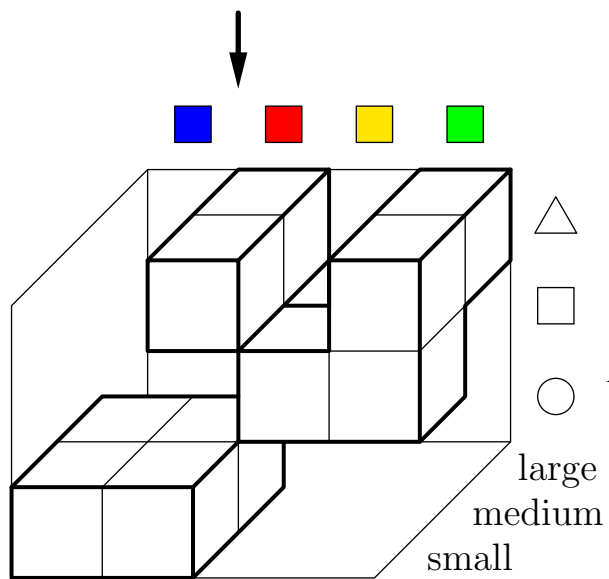
is called the **decomposition** of  $r_U$ .

- Equivalent to **join decomposability** in database theory (natural join).

# Relational Decomposition: Simple Example



Taking the minimum of the projection to the subspace spanned by color and shape and of the projection to the subspace spanned by shape and size yields the original three-dimensional relation.



# Conditional Possibility and Independence

**Definition:** Let  $\Omega$  be a (finite) sample space,  $R$  a discrete possibility measure on  $\Omega$ , and  $E_1, E_2 \subseteq \Omega$  events. Then

$$R(E_1 \mid E_2) = R(E_1 \cap E_2)$$

is called the **conditional possibility** of  $E_1$  given  $E_2$ .

**Definition:** Let  $\Omega$  be a (finite) sample space,  $R$  a discrete possibility measure on  $\Omega$ , and  $A, B$ , and  $C$  attributes with respective domains  $\text{dom}(A)$ ,  $\text{dom}(B)$ , and  $\text{dom}(C)$ .  $A$  and  $B$  are called **conditionally relationally independent** given  $C$ , written  $A \perp\!\!\!\perp_R B \mid C$ , iff

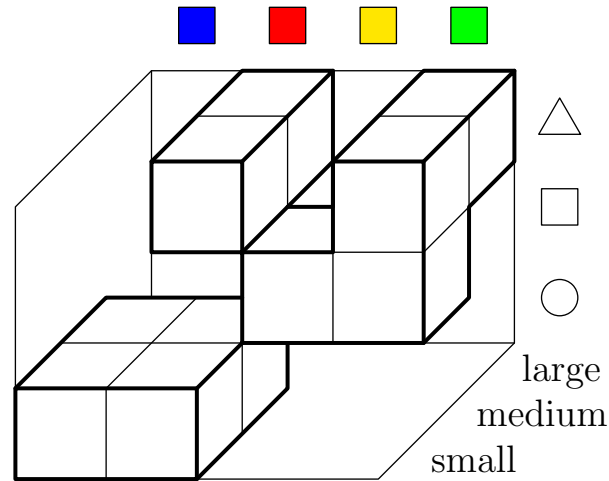
$$\forall a \in \text{dom}(A) : \forall b \in \text{dom}(B) : \forall c \in \text{dom}(C) :$$

$$R(A = a, B = b \mid C = c) = \min\{R(A = a \mid C = c), R(B = b \mid C = c)\},$$

$$\Leftrightarrow R(A = a, B = b, C = c) = \min\{R(A = a, C = c), R(B = b, C = c)\}.$$

- Similar to the corresponding notions of probability theory.

# Conditional Independence: Simple Example

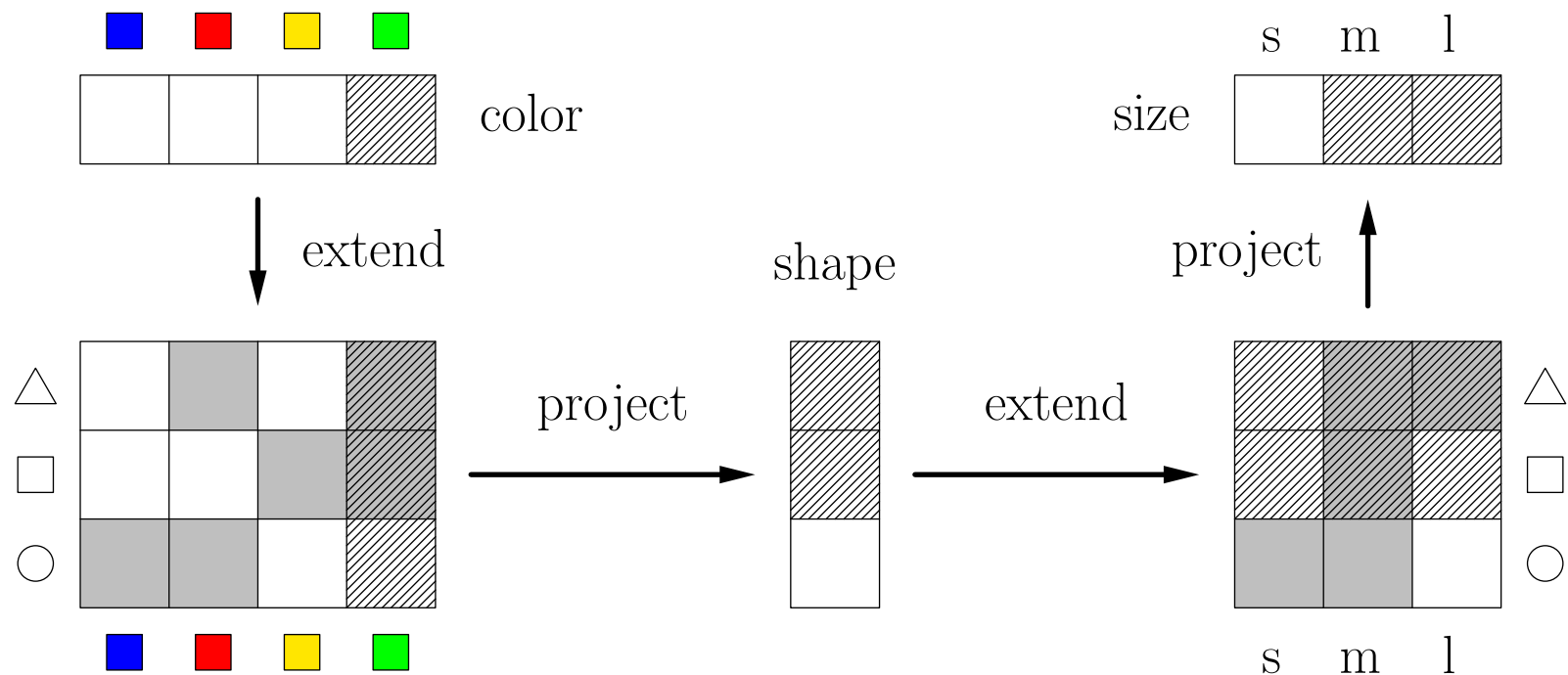


Example relation describing ten simple geometric objects by three attributes: color, shape, and size.

- In this example relation, the color of an object is conditionally relationally independent of its size given its shape.
- Intuitively: if we fix the shape, the colors and sizes that are possible together with this shape can be combined freely.
- Alternative view: once we know the shape, the color does not provide additional information about the size (and vice versa).

# Relational Evidence Propagation

Due to the fact that color and size are conditionally independent given the shape, the reasoning result can be obtained using only the projections to the subspaces:



This reasoning scheme can be formally justified with discrete possibility measures.

# Relational Evidence Propagation, Step 1

$A$ :	color
$B$ :	shape
$C$ :	size

$$\begin{aligned}
 & R(B = b \mid A = a_{\text{obs}}) \\
 &= R\left(\bigvee_{a \in \text{dom}(A)} A = a, B = b, \bigvee_{c \in \text{dom}(C)} C = c \mid A = a_{\text{obs}}\right) \\
 &\stackrel{(1)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C)} \{R(A = a, B = b, C = c \mid A = a_{\text{obs}})\} \right\} \\
 &\stackrel{(2)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C)} \left\{ \min\{R(A = a, B = b, C = c), R(A = a \mid A = a_{\text{obs}})\} \right\} \right\} \\
 &\stackrel{(3)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C)} \left\{ \min\{R(A = a, B = b), R(B = b, C = c), R(A = a \mid A = a_{\text{obs}})\} \right\} \right\} \\
 &= \max_{a \in \text{dom}(A)} \left\{ \min\{R(A = a, B = b), R(A = a \mid A = a_{\text{obs}}), \underbrace{\max_{c \in \text{dom}(C)} \{R(B = b, C = c)\}}_{=R(B=b) \geq R(A=a, B=b)}\} \right\} \\
 &= \max_{a \in \text{dom}(A)} \left\{ \min\{R(A = a, B = b), R(A = a \mid A = a_{\text{obs}})\} \right\}.
 \end{aligned}$$

# Relational Evidence Propagation, Step 1 (continued)

- (1) holds because of the second axiom a discrete possibility measure has to satisfy.
- (3) holds because of the fact that the relation  $R_{ABC}$  can be decomposed w.r.t. the set  $\mathcal{M} = \{\{A, B\}, \{B, C\}\}$ . ( $A$ : color,  $B$ : shape,  $C$ : size)
- (2) holds, since in the first place

$$\begin{aligned} R(A = a, B = b, C = c | A = a_{obs}) &= R(A = a, B = b, C = c, A = a_{obs}) \\ &= \begin{cases} R(A = a, B = b, C = c), & \text{if } a = a_{obs}, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

and secondly

$$\begin{aligned} R(A = a | A = a_{obs}) &= R(A = a, A = a_{obs}) \\ &= \begin{cases} R(A = a), & \text{if } a = a_{obs}, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

and therefore, since trivially  $R(A = a) \geq R(A = a, B = b, C = c)$ ,

$$\begin{aligned} R(A = a, B = b, C = c | A = a_{obs}) \\ &= \min\{R(A = a, B = b, C = c), R(A = a | A = a_{obs})\}. \end{aligned}$$



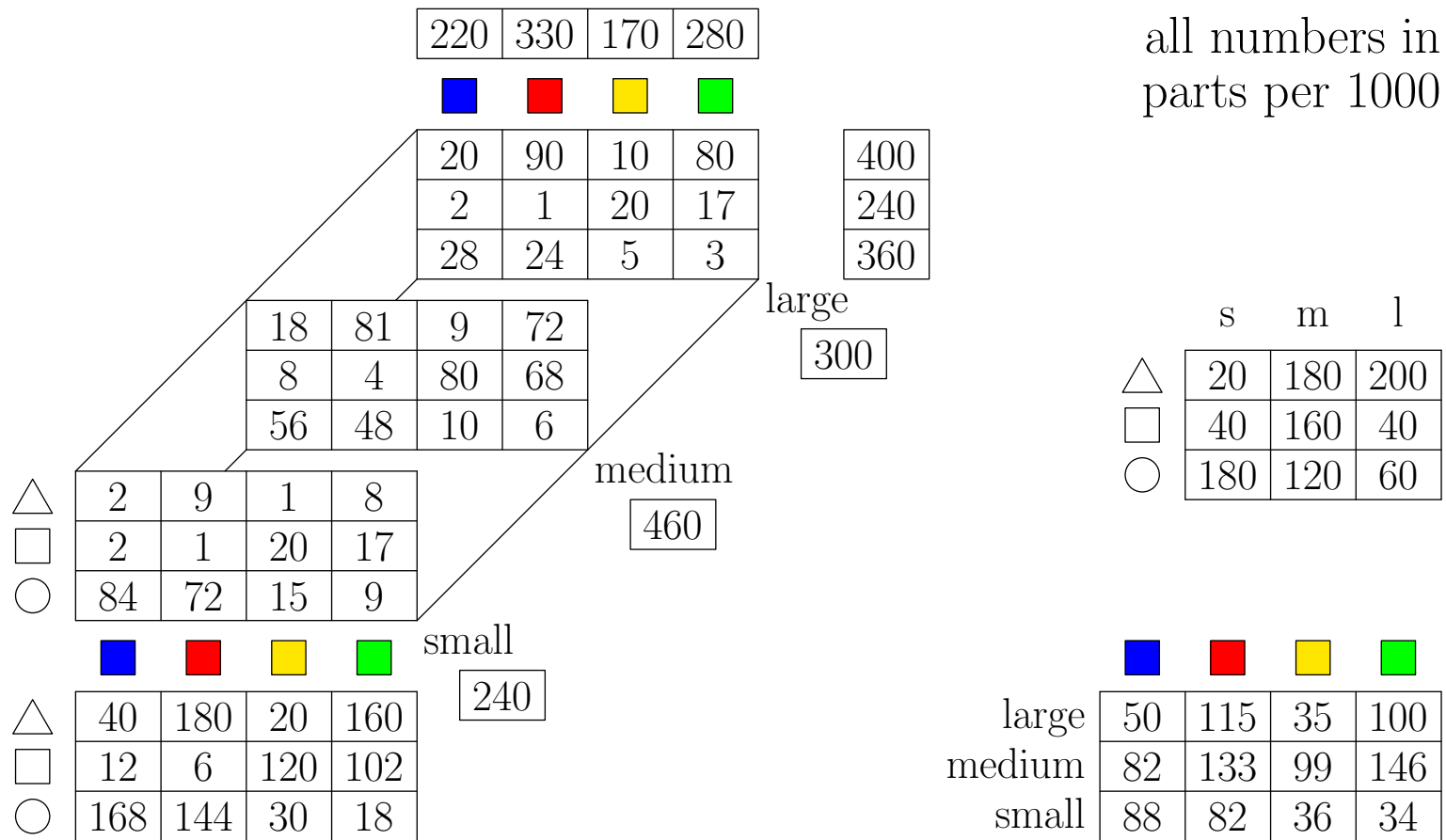
# Relational Evidence Propagation, Step 2

$A$ :	color
$B$ :	shape
$C$ :	size

$$\begin{aligned}
 & R(C = c \mid A = a_{\text{obs}}) \\
 &= R\left(\bigvee_{a \in \text{dom}(A)} A = a, \bigvee_{b \in \text{dom}(B)} B = b, C = c \mid A = a_{\text{obs}}\right) \\
 &\stackrel{(1)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{b \in \text{dom}(B)} \{R(A = a, B = b, C = c \mid A = a_{\text{obs}})\} \right\} \\
 &\stackrel{(2)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{b \in \text{dom}(B)} \left\{ \min\{R(A = a, B = b, C = c), R(A = a \mid A = a_{\text{obs}})\} \right\} \right\} \\
 &\stackrel{(3)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{b \in \text{dom}(B)} \left\{ \min\{R(A = a, B = b), R(B = b, C = c), \right. \right. \\
 &\quad \left. \left. R(A = a \mid A = a_{\text{obs}})\} \right\} \right\} \\
 &= \max_{b \in \text{dom}(B)} \left\{ \min\{R(B = b, C = c), \right. \\
 &\quad \left. \underbrace{\max_{a \in \text{dom}(A)} \{ \min\{R(A = a, B = b), R(A = a \mid A = a_{\text{obs}})\} \}}_{=R(B=b \mid A=a_{\text{obs}})} \right\} \\
 &= \max_{b \in \text{dom}(B)} \left\{ \min\{R(B = b, C = c), R(B = b \mid A = a_{\text{obs}})\} \right\}.
 \end{aligned}$$

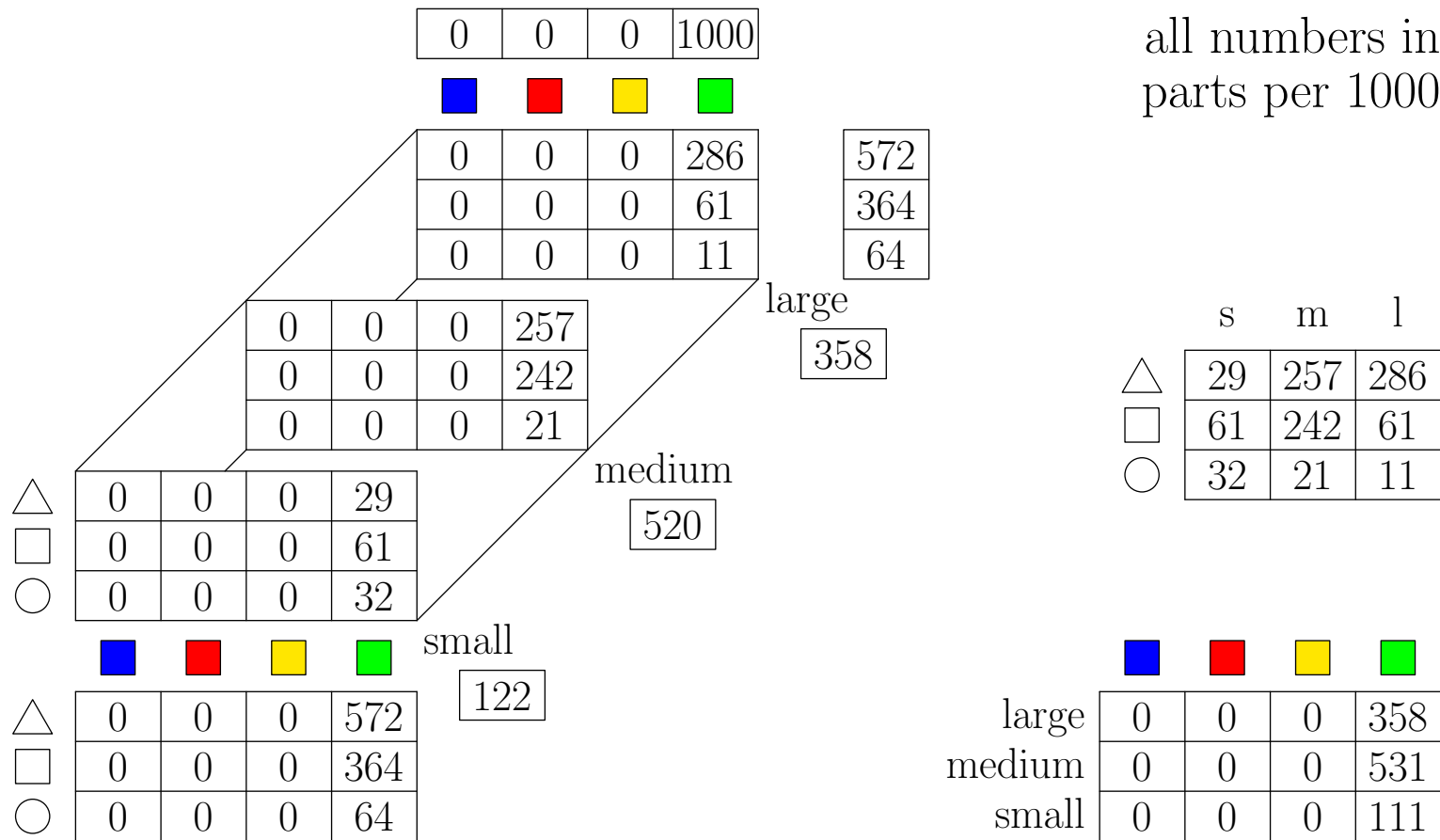
# A Simple Example: The Probabilistic Case

# A Probability Distribution



The numbers state the probability of the corresponding value combination.  
Compared to the example relation, the possible combinations are now frequent.

# Reasoning: Computing Conditional Probabilities



Using the information that the given object is green:  
 The observed color has a posterior probability of 1.

# Probabilistic Decomposition: Simple Example

- As for relational graphical models, the three-dimensional probability distribution can be decomposed into projections to subspaces, namely the marginal distribution on the subspace spanned by color and shape and the marginal distribution on the subspace spanned by shape and size.
- The original probability distribution can be reconstructed from the marginal distributions using the following formulae  $\forall i, j, k$  :

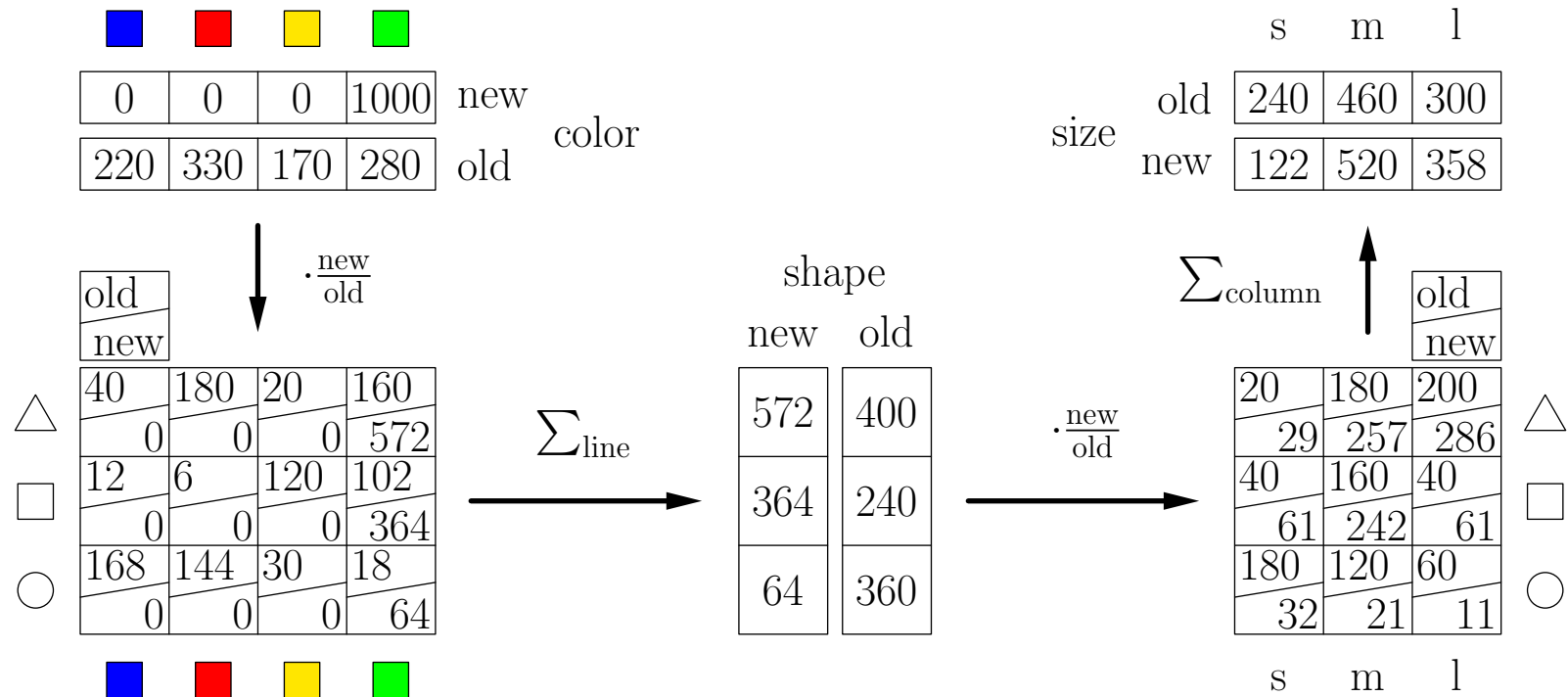
$$\begin{aligned} P\left(a_i^{(\text{color})}, a_j^{(\text{shape})}, a_k^{(\text{size})}\right) &= P\left(a_i^{(\text{color})}, a_j^{(\text{shape})}\right) \cdot P\left(a_k^{(\text{size})} \mid a_j^{(\text{shape})}\right) \\ &= \frac{P\left(a_i^{(\text{color})}, a_j^{(\text{shape})}\right) \cdot P\left(a_j^{(\text{shape})}, a_k^{(\text{size})}\right)}{P\left(a_j^{(\text{shape})}\right)} \end{aligned}$$

- These equations express the *conditional independence* of attributes *color* and *size* given the attribute *shape*, since they only hold if  $\forall i, j, k$  :

$$P\left(a_k^{(\text{size})} \mid a_j^{(\text{shape})}\right) = P\left(a_k^{(\text{size})} \mid a_i^{(\text{color})}, a_j^{(\text{shape})}\right)$$

# Reasoning with Projections

Again the same result can be obtained using only projections to subspaces (marginal probability distributions):



This justifies a graph representation:



# Probabilistic Graphical Models: Formalization

# Probabilistic Decomposition

**Definition:** Let  $U = \{A_1, \dots, A_n\}$  be a set of attributes and  $p_U$  a probability distribution over  $U$ . Furthermore, let  $\mathcal{M} = \{M_1, \dots, M_m\} \subseteq 2^U$  be a set of nonempty (but not necessarily disjoint) subsets of  $U$  satisfying

$$\bigcup_{M \in \mathcal{M}} M = U.$$

$p_U$  is called **decomposable** or **factorizable** w.r.t.  $\mathcal{M}$  iff it can be written as a product of  $m$  nonnegative functions  $\phi_M : \mathcal{E}_M \rightarrow \mathbb{R}_0^+$ ,  $M \in \mathcal{M}$ , i.e., iff

$$\forall a_1 \in \text{dom}(A_1) : \dots \forall a_n \in \text{dom}(A_n) : \\ p_U \left( \bigwedge_{A_i \in U} A_i = a_i \right) = \prod_{M \in \mathcal{M}} \phi_M \left( \bigwedge_{A_i \in M} A_i = a_i \right).$$

If  $p_U$  is decomposable w.r.t.  $\mathcal{M}$  the set of functions

$$\Phi_{\mathcal{M}} = \{\phi_{M_1}, \dots, \phi_{M_m}\} = \{\phi_M \mid M \in \mathcal{M}\}$$

is called the **decomposition** or the **factorization** of  $p_U$ .

The functions in  $\Phi_{\mathcal{M}}$  are called the **factor potentials** of  $p_U$ .



# Conditional Independence

**Definition:** Let  $\Omega$  be a (finite) sample space,  $P$  a probability measure on  $\Omega$ , and  $A$ ,  $B$ , and  $C$  attributes with respective domains  $\text{dom}(A)$ ,  $\text{dom}(B)$ , and  $\text{dom}(C)$ .  $A$  and  $B$  are called **conditionally probabilistically independent** given  $C$ , written  $A \perp\!\!\!\perp_P B \mid C$ , iff

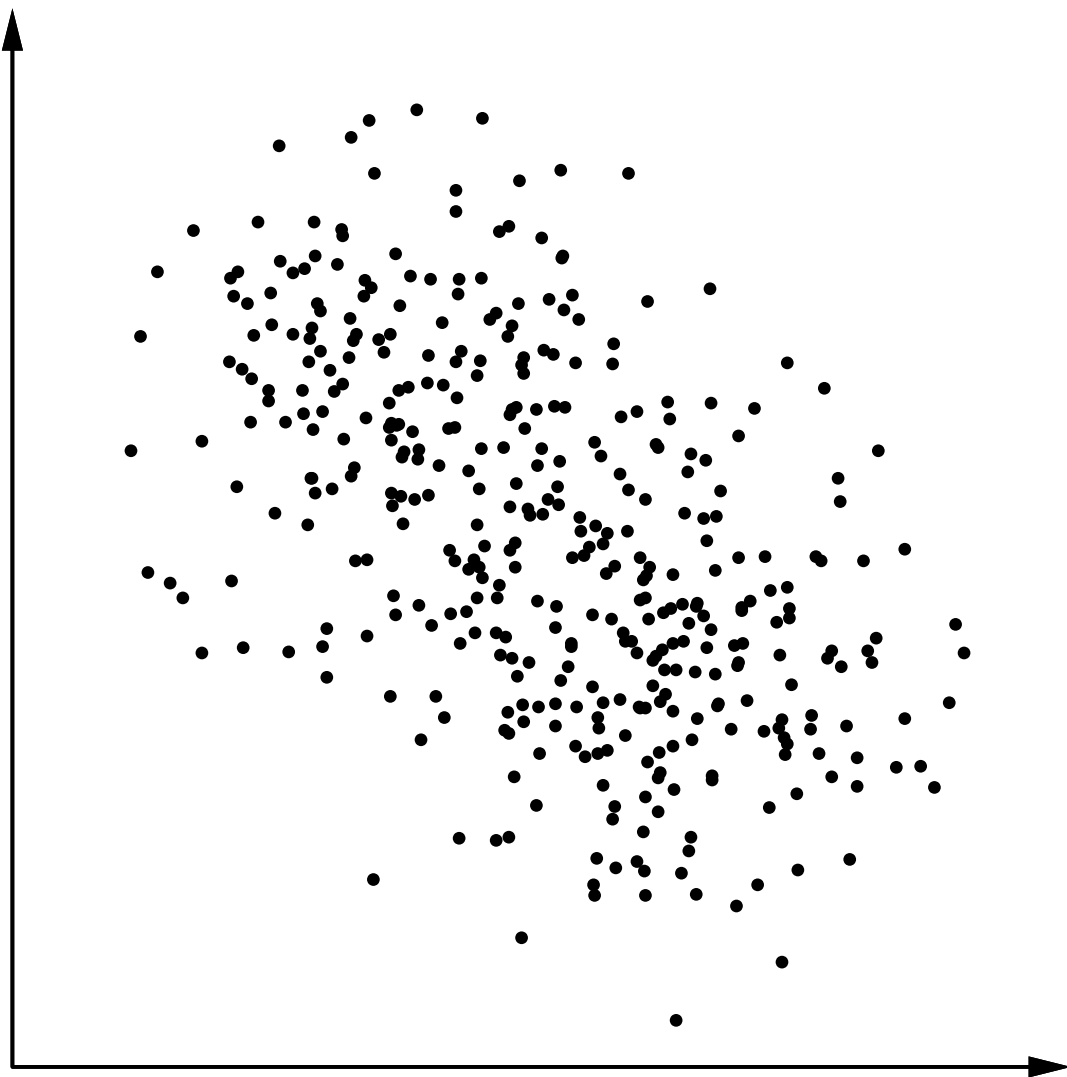
$$\forall a \in \text{dom}(A) : \forall b \in \text{dom}(B) : \forall c \in \text{dom}(C) : \\ P(A = a, B = b \mid C = c) = P(A = a \mid C = c) \cdot P(B = b \mid C = c)$$

Equivalent formula (sometimes more convenient):

$$\forall a \in \text{dom}(A) : \forall b \in \text{dom}(B) : \forall c \in \text{dom}(C) : \\ P(A = a \mid B = b, C = c) = P(A = a \mid C = c)$$

- Conditional independences make it possible to consider parts of a probability distribution independent of others.
- Therefore it is plausible that a set of conditional independences may enable a decomposition of a joint probability distribution.

# Conditional Independence: An Example



Dependence (fictitious) between smoking and life expectancy.

Each dot represents one person.

$x$ -axis: age at death

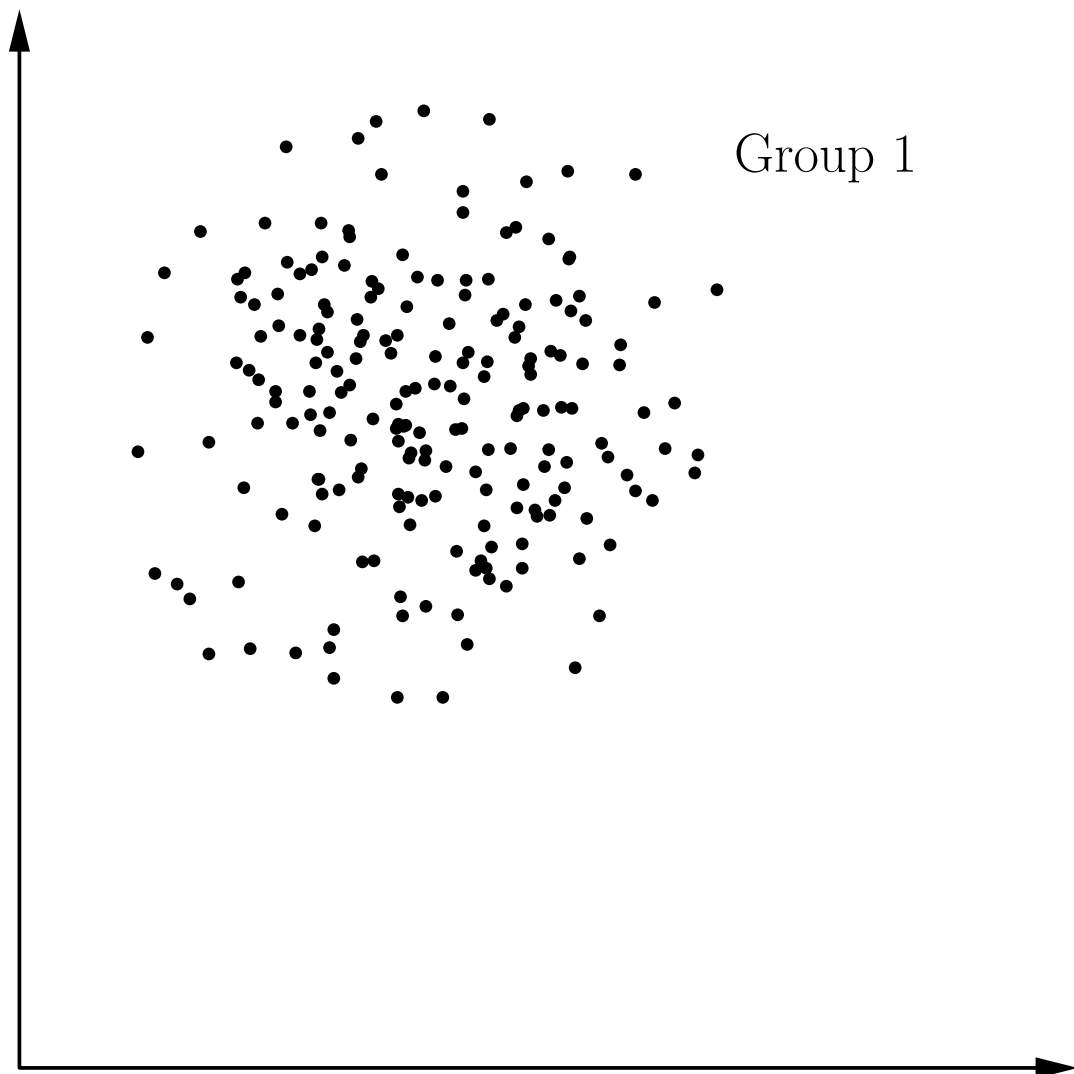
$y$ -axis: average number of cigarettes per day

Weak, but clear dependence:

The more cigarettes are smoked, the lower the life expectancy.

(Note that this data is artificial and thus should not be seen as revealing an actual dependence.)

# Conditional Independence: An Example



Group 1

Conjectured explanation:

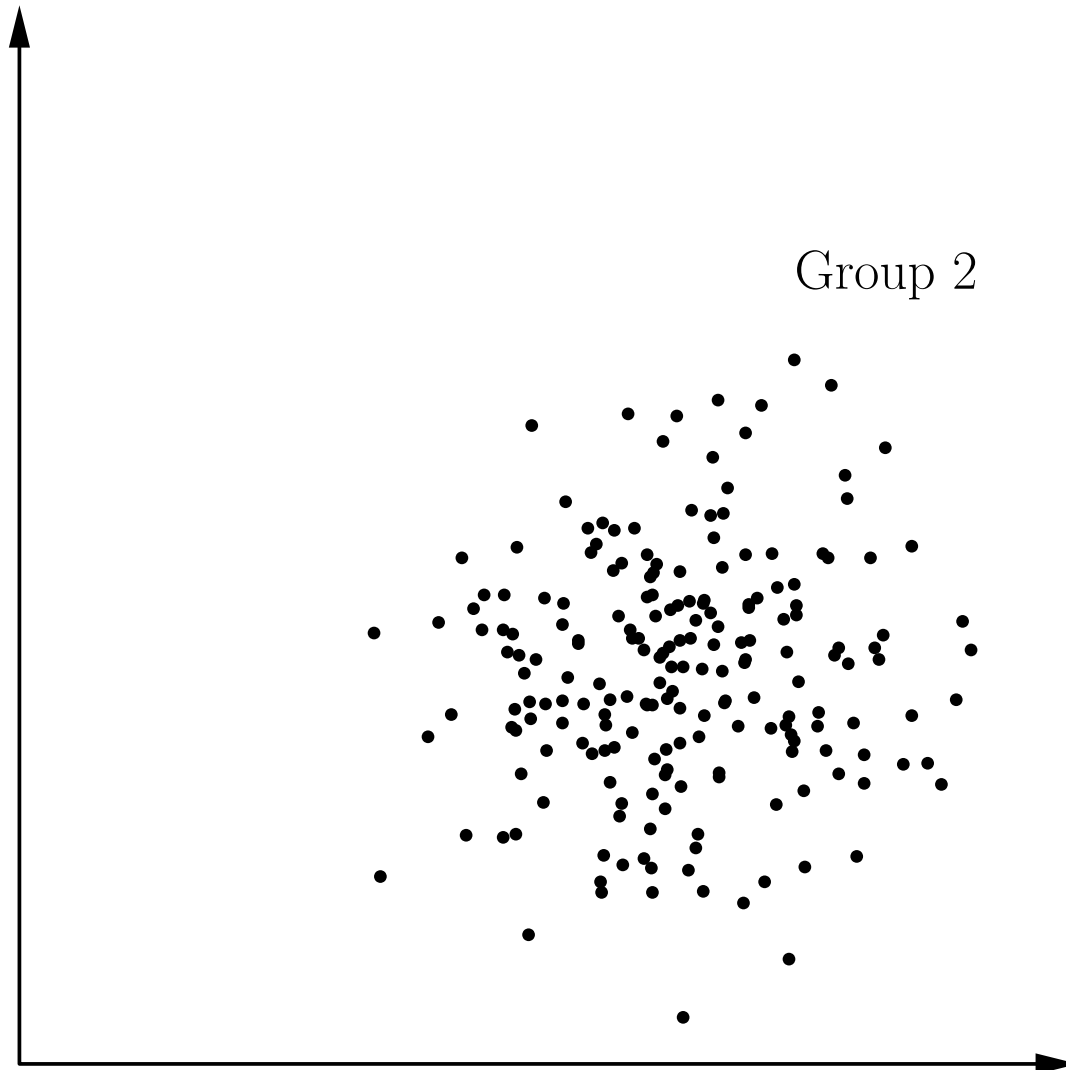
There is a common cause, namely whether the person is exposed to stress at work.

If this were correct, splitting the data should remove the dependence.

Group 1:  
exposed to stress at work

(Note that this data is artificial and therefore should not be seen as an argument against health hazards caused by smoking.)

# Conditional Independence: An Example



Conjectured explanation:

There is a common cause, namely whether the person is exposed to stress at work.

If this were correct, splitting the data should remove the dependence.

Group 2:  
**not** exposed to stress at work

(Note that this data is artificial and therefore should not be seen as an argument against health hazards caused by smoking.)

# Probabilistic Decomposition (continued)

## Chain Rule of Probability:

$\forall a_1 \in \text{dom}(A_1) : \dots \forall a_n \in \text{dom}(A_n) :$

$$P\left(\bigwedge_{i=1}^n A_i = a_i\right) = \prod_{i=1}^n P\left(A_i = a_i \mid \bigwedge_{j=1}^{i-1} A_j = a_j\right)$$

- The chain rule of probability is valid in general (or at least for strictly positive distributions).

## Chain Rule Factorization:

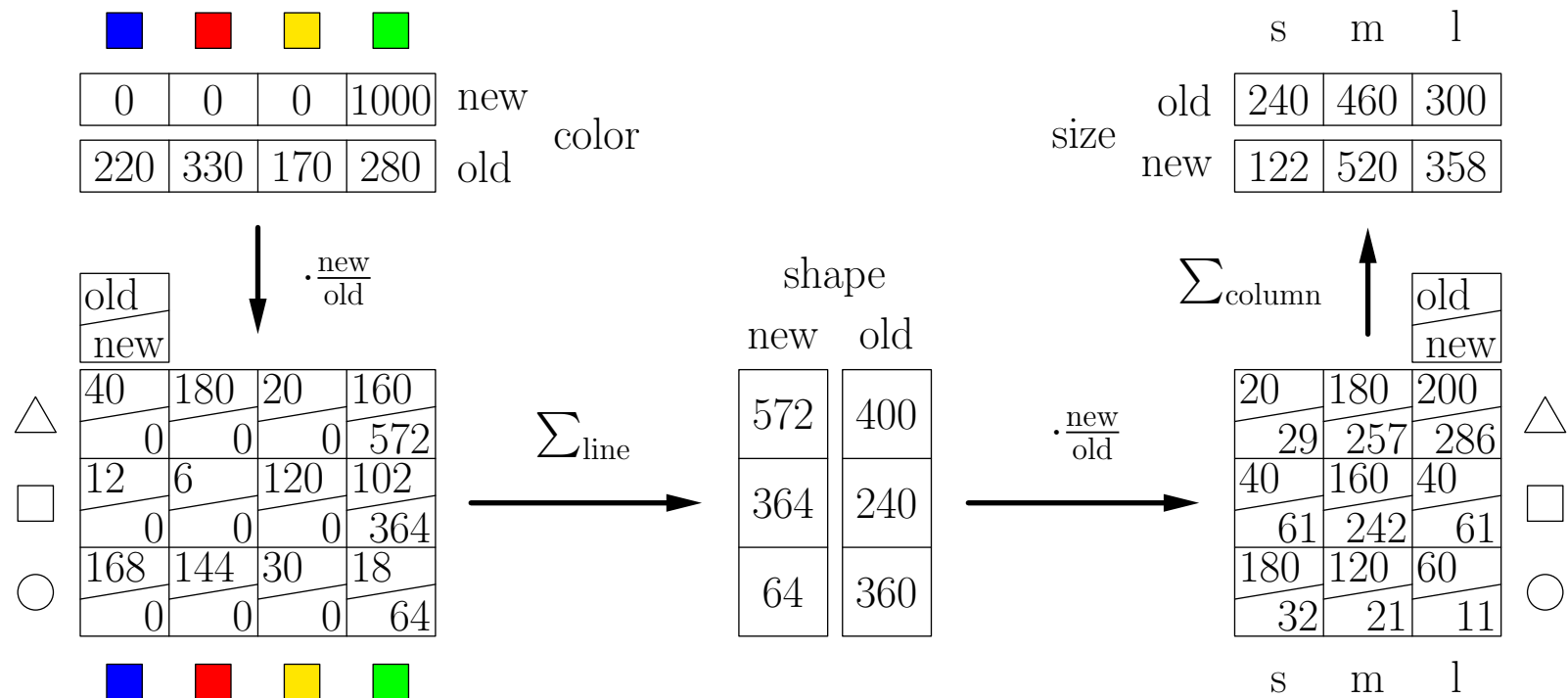
$\forall a_1 \in \text{dom}(A_1) : \dots \forall a_n \in \text{dom}(A_n) :$

$$P\left(\bigwedge_{i=1}^n A_i = a_i\right) = \prod_{i=1}^n P\left(A_i = a_i \mid \bigwedge_{A_j \in \text{parents}(A_i)} A_j = a_j\right)$$

- Conditional independence statements are used to “cancel” conditions.

# Reasoning with Projections

Due to the fact that color and size are conditionally independent given the shape, the reasoning result can be obtained using only the projections to the subspaces:



This reasoning scheme can be formally justified with probability measures.

# Probabilistic Evidence Propagation, Step 1

$$\begin{aligned} & P(B = b \mid A = a_{\text{obs}}) \\ &= P\left(\bigvee_{a \in \text{dom}(A)} A = a, B = b, \bigvee_{c \in \text{dom}(C)} C = c \mid A = a_{\text{obs}}\right) \\ &\stackrel{(1)}{=} \sum_{a \in \text{dom}(A)} \sum_{c \in \text{dom}(C)} P(A = a, B = b, C = c \mid A = a_{\text{obs}}) \\ &\stackrel{(2)}{=} \sum_{a \in \text{dom}(A)} \sum_{c \in \text{dom}(C)} P(A = a, B = b, C = c) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)} \\ &\stackrel{(3)}{=} \sum_{a \in \text{dom}(A)} \sum_{c \in \text{dom}(C)} \frac{P(A = a, B = b)P(B = b, C = c)}{P(B = b)} \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)} \\ &= \sum_{a \in \text{dom}(A)} P(A = a, B = b) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)} \underbrace{\sum_{c \in \text{dom}(C)} P(C = c \mid B = b)}_{=1} \\ &= \sum_{a \in \text{dom}(A)} P(A = a, B = b) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)}. \end{aligned}$$

$A$ :	color
$B$ :	shape
$C$ :	size

# Probabilistic Evidence Propagation, Step 1 (continued)

(1) holds because of Kolmogorov's axioms.

(3) holds because of the fact that the distribution  $p_{ABC}$  can be decomposed w.r.t. the set  $\mathcal{M} = \{\{A, B\}, \{B, C\}\}$ . (A: color, B: shape, C: size)

(2) holds, since in the first place

$$P(A = a, B = b, C = c | A = a_{obs}) = \frac{P(A = a, B = b, C = c, A = a_{obs})}{P(A = a_{obs})}$$
$$= \begin{cases} \frac{P(A = a, B = b, C = c)}{P(A = a_{obs})}, & \text{if } a = a_{obs}, \\ 0, & \text{otherwise,} \end{cases}$$

and secondly

$$P(A = a, A = a_{obs}) = \begin{cases} P(A = a), & \text{if } a = a_{obs}, \\ 0, & \text{otherwise,} \end{cases}$$

and therefore

$$P(A = a, B = b, C = c | A = a_{obs})$$
$$= P(A = a, B = b, C = c) \cdot \frac{P(A = a | A = a_{obs})}{P(A = a)}.$$



# Probabilistic Evidence Propagation, Step 2

$$P(C = c \mid A = a_{\text{obs}}) = P\left(\bigvee_{a \in \text{dom}(A)} A = a, \bigvee_{b \in \text{dom}(B)} B = b, C = c \mid A = a_{\text{obs}}\right)$$

A:	color
B:	shape
C:	size

$$\stackrel{(1)}{=} \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} P(A = a, B = b, C = c \mid A = a_{\text{obs}})$$

$$\stackrel{(2)}{=} \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} P(A = a, B = b, C = c) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)}$$

$$\stackrel{(3)}{=} \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} \frac{P(A = a, B = b)P(B = b, C = c)}{P(B = b)} \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)}$$

$$= \sum_{b \in \text{dom}(B)} \frac{P(B = b, C = c)}{P(B = b)} \underbrace{\sum_{a \in \text{dom}(A)} P(A = a, B = b) \cdot \frac{P(A = a \mid A = a_{\text{obs}})}{P(A = a)}}_{=P(B=b \mid A=a_{\text{obs}})}$$

$$= \sum_{b \in \text{dom}(B)} P(B = b, C = c) \cdot \frac{P(B = b \mid A = a_{\text{obs}})}{P(B = b)}$$

# Excursion: Possibility Theory

# Possibility Theory

- The best-known calculus for handling uncertainty is, of course, **probability theory**. [Laplace 1812]
- An less well-known, but noteworthy alternative is **possibility theory**. [Dubois and Prade 1988]
- In the interpretation we consider here, possibility theory can handle **uncertain and imprecise information**, while probability theory, at least in its basic form, was only designed to handle *uncertain information*.
- Types of **imperfect information**:
  - **Imprecision**: disjunctive or set-valued information about the obtaining state, which is certain: the true state is contained in the disjunction or set.
  - **Uncertainty**: precise information about the obtaining state (single case), which is not certain: the true state may differ from the stated one.
  - **Vagueness**: meaning of the information is in doubt: the interpretation of the given statements about the obtaining state may depend on the user.

# Possibility Theory: Axiomatic Approach

**Definition:** Let  $\Omega$  be a (finite) sample space.

A **possibility measure**  $\Pi$  on  $\Omega$  is a function  $\Pi : 2^\Omega \rightarrow [0, 1]$  satisfying

1.  $\Pi(\emptyset) = 0$       and
2.  $\forall E_1, E_2 \subseteq \Omega : \Pi(E_1 \cup E_2) = \max\{\Pi(E_1), \Pi(E_2)\}$ .

- Similar to Kolmogorov's axioms of probability theory.
- From the axioms follows  $\Pi(E_1 \cap E_2) \leq \min\{\Pi(E_1), \Pi(E_2)\}$ .
- Attributes are introduced as random variables (as in probability theory).
- $\Pi(A = a)$  is an abbreviation of  $\Pi(\{\omega \in \Omega \mid A(\omega) = a\})$
- If an event  $E$  is possible without restriction, then  $\Pi(E) = 1$ .  
If an event  $E$  is impossible, then  $\Pi(E) = 0$ .

# Possibility Theory and the Context Model

## Interpretation of Degrees of Possibility

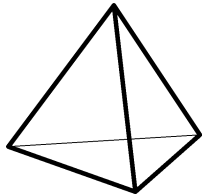
[Gebhardt and Kruse 1993]

- Let  $\Omega$  be the (nonempty) set of all possible states of the world,  $\omega_0$  the actual (but unknown) state.
- Let  $C = \{c_1, \dots, c_n\}$  be a set of contexts (observers, frame conditions etc.) and  $(C, 2^C, P)$  a finite probability space (context weights).
- Let  $\Gamma : C \rightarrow 2^\Omega$  be a set-valued mapping, which assigns to each context the **most specific correct set-valued specification of  $\omega_0$** . The sets  $\Gamma(c)$  are called the **focal sets** of  $\Gamma$ .
- $\Gamma$  is a **random set** (i.e., a set-valued random variable) [Nguyen 1978]. The **basic possibility assignment** induced by  $\Gamma$  is the mapping

$$\begin{aligned}\pi : \Omega &\rightarrow [0, 1] \\ \pi(\omega) &\mapsto P(\{c \in C \mid \omega \in \Gamma(c)\}).\end{aligned}$$

# Example: Dice and Shakers

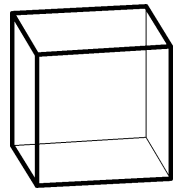
shaker 1



tetrahedron

1 - 4

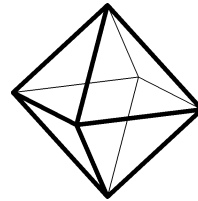
shaker 2



hexahedron

1 - 6

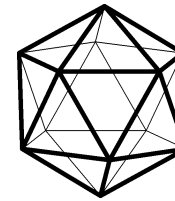
shaker 3



octahedron

1 - 8

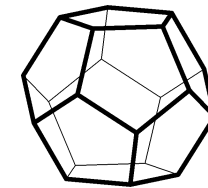
shaker 4



icosahedron

1 - 10

shaker 5



dodecahedron

1 - 12

numbers	degree of possibility
1 - 4	$\frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} = 1$
5 - 6	$\frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} = \frac{4}{5}$
7 - 8	$\frac{1}{5} + \frac{1}{5} + \frac{1}{5} = \frac{3}{5}$
9 - 10	$\frac{1}{5} + \frac{1}{5} = \frac{2}{5}$
11 - 12	$\frac{1}{5} = \frac{1}{5}$

# From the Context Model to Possibility Measures

**Definition:** Let  $\Gamma : C \rightarrow 2^\Omega$  be a random set.

The **possibility measure** induced by  $\Gamma$  is the mapping

$$\begin{aligned} \Pi : 2^\Omega &\rightarrow [0, 1], \\ E &\mapsto P(\{c \in C \mid E \cap \Gamma(c) \neq \emptyset\}). \end{aligned}$$

**Problem:** From the given interpretation it follows only:

$$\forall E \subseteq \Omega : \max_{\omega \in E} \pi(\omega) \leq \Pi(E) \leq \min \left\{ 1, \sum_{\omega \in E} \pi(\omega) \right\}.$$

	1	2	3	4	5
$c_1 : \frac{1}{2}$			•		
$c_2 : \frac{1}{4}$		•	•	•	
$c_3 : \frac{1}{4}$	•	•	•	•	•
$\pi$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{4}$

	1	2	3	4	5
$c_1 : \frac{1}{2}$			•		
$c_2 : \frac{1}{4}$	•	•			
$c_3 : \frac{1}{4}$				•	•
$\pi$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

# From the Context Model to Possibility Measures (cont.)

Attempts to solve the indicated problem:

- Require the focal sets to be **consonant**:

**Definition:** Let  $\Gamma : C \rightarrow 2^\Omega$  be a random set with  $C = \{c_1, \dots, c_n\}$ . The focal sets  $\Gamma(c_i)$ ,  $1 \leq i \leq n$ , are called **consonant**, iff there exists a sequence  $c_{i_1}, c_{i_2}, \dots, c_{i_n}$ ,  $1 \leq i_1, \dots, i_n \leq n$ ,  $\forall 1 \leq j < k \leq n : i_j \neq i_k$ , so that

$$\Gamma(c_{i_1}) \subseteq \Gamma(c_{i_2}) \subseteq \dots \subseteq \Gamma(c_{i_n}).$$

→ mass assignment theory [Baldwin *et al.* 1995]

**Problem:** The “voting model” is not sufficient to justify consonance.

- Use the lower bound as the “most pessimistic” choice. [Gebhardt 1997]

**Problem:** Basic possibility assignments represent negative information, the lower bound is actually the *most optimistic* choice.

- Justify the lower bound from decision making purposes.  
[Borgelt 1995, Borgelt 2000]



# From the Context Model to Possibility Measures (cont.)

- Assume that in the end we have to decide on a single event.
- Each event is described by the values of a set of attributes.
- Then it can be useful to assign to a set of events the degree of possibility of the “most possible” event in the set.

Example:

$\Sigma$	36	18	18	28	
28	0	0	0	28	28
18	18	0	0	0	18
18	18	0	0	0	18
36	0	18	18	0	18
	18	18	18	28	max

0	40	0	40
40	0	0	40
0	0	20	20
40	40	20	max

# Possibility Distributions

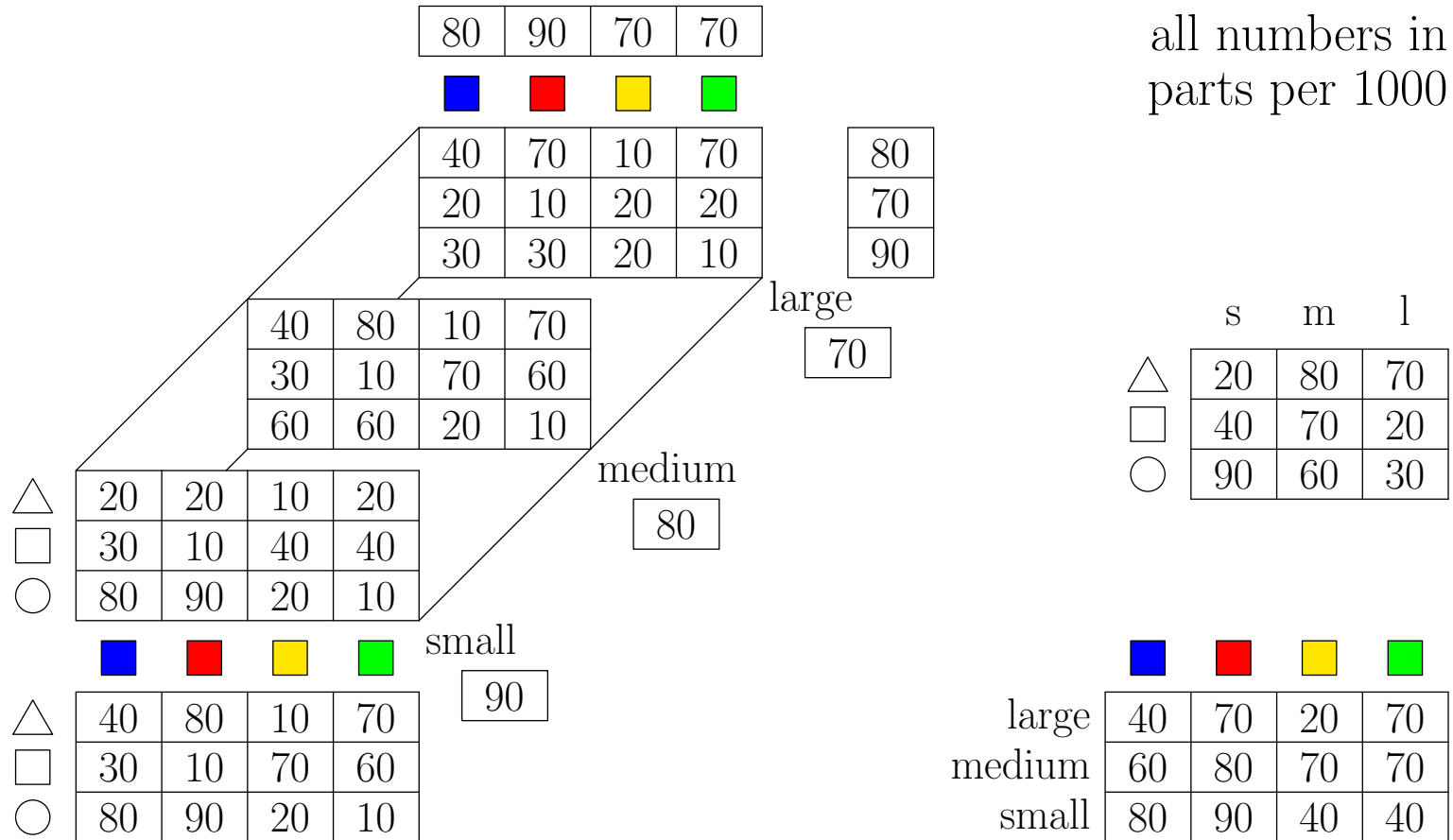
**Definition:** Let  $X = \{A_1, \dots, A_n\}$  be a set of attributes defined on a (finite) sample space  $\Omega$  with respective domains  $\text{dom}(A_i)$ ,  $i = 1, \dots, n$ . A **possibility distribution**  $\pi_X$  over  $X$  is the restriction of a possibility measure  $\Pi$  on  $\Omega$  to the set of all events that can be defined by stating values for all attributes in  $X$ . That is,  $\pi_X = \Pi|_{\mathcal{E}_X}$ , where

$$\begin{aligned}\mathcal{E}_X &= \left\{ E \in 2^\Omega \mid \exists a_1 \in \text{dom}(A_1) : \dots \exists a_n \in \text{dom}(A_n) : \right. \\ &\quad \left. E \cong \bigwedge_{A_j \in X} A_j = a_j \right\} \\ &= \left\{ E \in 2^\Omega \mid \exists a_1 \in \text{dom}(A_1) : \dots \exists a_n \in \text{dom}(A_n) : \right. \\ &\quad \left. E = \left\{ \omega \in \Omega \mid \bigwedge_{A_j \in X} A_j(\omega) = a_j \right\} \right\}.\end{aligned}$$

- Corresponds to the notion of a probability distribution.
- Advantage of this formalization: No index transformation functions are needed for projections, there are just fewer terms in the conjunctions.

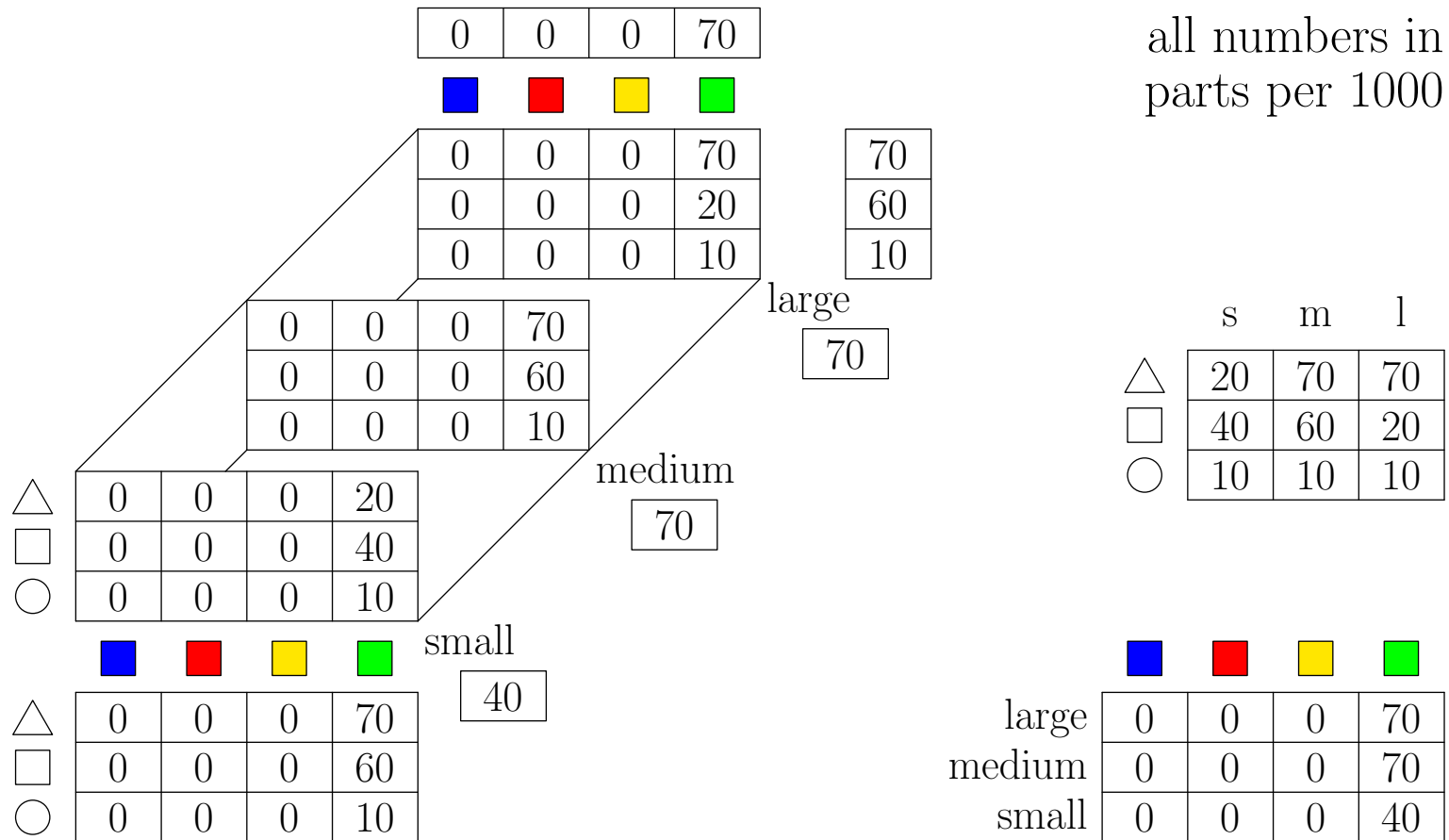
# A Simple Example: The Possibilistic Case

# A Possibility Distribution



- The numbers state the degrees of possibility of the corresp. value combination.

# Reasoning



- Using the information that the given object is green.

# Possibilistic Decomposition

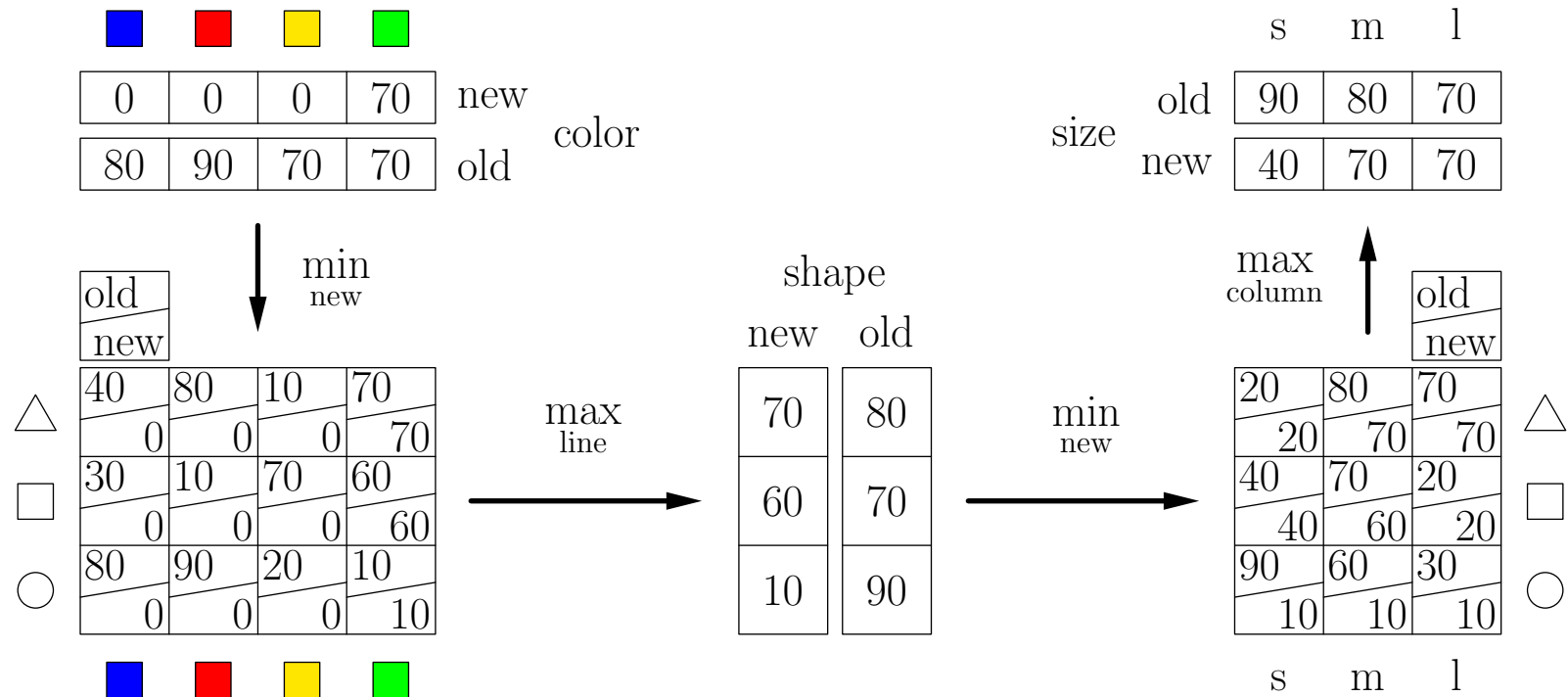
- As for relational and probabilistic networks, the three-dimensional possibility distribution can be decomposed into projections to subspaces, namely:
  - the maximum projection to the subspace color  $\times$  shape and
  - the maximum projection to the subspace shape  $\times$  size.
- It can be reconstructed using the following formula:

$$\begin{aligned}\forall i, j, k : \pi \left( a_i^{(\text{color})}, a_j^{(\text{shape})}, a_k^{(\text{size})} \right) \\ &= \min \left\{ \pi \left( a_i^{(\text{color})}, a_j^{(\text{shape})} \right), \pi \left( a_j^{(\text{shape})}, a_k^{(\text{size})} \right) \right\} \\ &= \min \left\{ \max_k \pi \left( a_i^{(\text{color})}, a_j^{(\text{shape})}, a_k^{(\text{size})} \right), \right. \\ &\quad \left. \max_i \pi \left( a_i^{(\text{color})}, a_j^{(\text{shape})}, a_k^{(\text{size})} \right) \right\}\end{aligned}$$

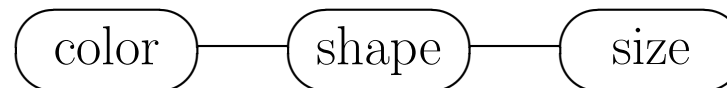
- Note the analogy to the probabilistic reconstruction formulas.

# Reasoning with Projections

Again the same result can be obtained using only projections to subspaces (maximal degrees of possibility):



This justifies a graph representation:



# Possibilistic Graphical Models: Formalization



# Conditional Possibility and Independence

**Definition:** Let  $\Omega$  be a (finite) sample space,  $\Pi$  a possibility measure on  $\Omega$ , and  $E_1, E_2 \subseteq \Omega$  events. Then

$$\Pi(E_1 \mid E_2) = \Pi(E_1 \cap E_2)$$

is called the **conditional possibility** of  $E_1$  given  $E_2$ .

**Definition:** Let  $\Omega$  be a (finite) sample space,  $\Pi$  a possibility measure on  $\Omega$ , and  $A, B$ , and  $C$  attributes with respective domains  $\text{dom}(A)$ ,  $\text{dom}(B)$ , and  $\text{dom}(C)$ .  $A$  and  $B$  are called **conditionally possibilistically independent** given  $C$ , written  $A \perp\!\!\!\perp_{\Pi} B \mid C$ , iff

$$\forall a \in \text{dom}(A) : \forall b \in \text{dom}(B) : \forall c \in \text{dom}(C) :$$

$$\Pi(A = a, B = b \mid C = c) = \min\{\Pi(A = a \mid C = c), \Pi(B = b \mid C = c)\}.$$

- Similar to the corresponding notions of probability theory.

# Possibilistic Evidence Propagation, Step 1

$A$ :	color
$B$ :	shape
$C$ :	size

$$\begin{aligned}
 & \pi(B = b \mid A = a_{\text{obs}}) \\
 &= \pi\left(\bigvee_{a \in \text{dom}(A)} A = a, B = b, \bigvee_{c \in \text{dom}(C)} C = c \mid A = a_{\text{obs}}\right) \\
 &\stackrel{(1)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C)} \left\{ \pi(A = a, B = b, C = c \mid A = a_{\text{obs}}) \right\} \right\} \\
 &\stackrel{(2)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C)} \left\{ \min\left\{ \pi(A = a, B = b, C = c), \pi(A = a \mid A = a_{\text{obs}}) \right\} \right\} \right\} \\
 &\stackrel{(3)}{=} \max_{a \in \text{dom}(A)} \left\{ \max_{c \in \text{dom}(C)} \left\{ \min\left\{ \pi(A = a, B = b), \pi(B = b, C = c), \right. \right. \right. \\
 &\quad \left. \left. \left. \pi(A = a \mid A = a_{\text{obs}}) \right\} \right\} \right\} \\
 &= \max_{a \in \text{dom}(A)} \left\{ \min\left\{ \pi(A = a, B = b), \pi(A = a \mid A = a_{\text{obs}}), \right. \right. \\
 &\quad \left. \left. \underbrace{\max_{c \in \text{dom}(C)} \left\{ \pi(B = b, C = c) \right\}}_{=\pi(B=b) \geq \pi(A=a, B=b)} \right\} \right\} \\
 &= \max_{a \in \text{dom}(A)} \left\{ \min\left\{ \pi(A = a, B = b), \pi(A = a \mid A = a_{\text{obs}}) \right\} \right\}
 \end{aligned}$$

# Graphical Models: The General Theory

# (Semi-)Graphoid Axioms

**Definition:** Let  $V$  be a set of (mathematical) objects and  $(\cdot \perp\!\!\!\perp \cdot \mid \cdot)$  a three-place relation of subsets of  $V$ . Furthermore, let  $W, X, Y,$  and  $Z$  be four disjoint subsets of  $V$ . The four statements

symmetry:  $(X \perp\!\!\!\perp Y \mid Z) \Rightarrow (Y \perp\!\!\!\perp X \mid Z)$

decomposition:  $(W \cup X \perp\!\!\!\perp Y \mid Z) \Rightarrow (W \perp\!\!\!\perp Y \mid Z) \wedge (X \perp\!\!\!\perp Y \mid Z)$

weak union:  $(W \cup X \perp\!\!\!\perp Y \mid Z) \Rightarrow (X \perp\!\!\!\perp Y \mid Z \cup W)$

contraction:  $(X \perp\!\!\!\perp Y \mid Z \cup W) \wedge (W \perp\!\!\!\perp Y \mid Z) \Rightarrow (W \cup X \perp\!\!\!\perp Y \mid Z)$

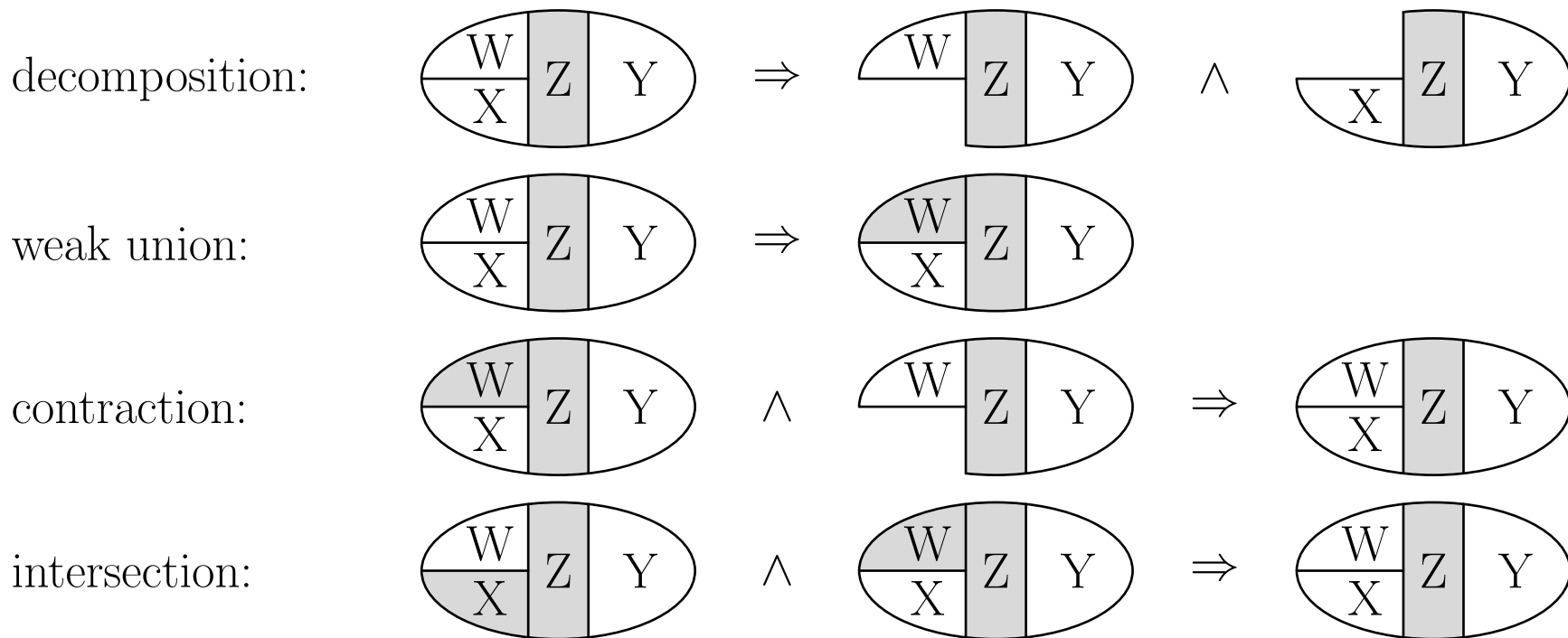
are called the **semi-graphoid axioms**. A three-place relation  $(\cdot \perp\!\!\!\perp \cdot \mid \cdot)$  that satisfies the semi-graphoid axioms for all  $W, X, Y,$  and  $Z$  is called a **semi-graphoid**.

The above four statements together with

intersection:  $(W \perp\!\!\!\perp Y \mid Z \cup X) \wedge (X \perp\!\!\!\perp Y \mid Z \cup W) \Rightarrow (W \cup X \perp\!\!\!\perp Y \mid Z)$

are called the **graphoid axioms**. A three-place relation  $(\cdot \perp\!\!\!\perp \cdot \mid \cdot)$  that satisfies the graphoid axioms for all  $W, X, Y,$  and  $Z$  is called a **graphoid**.

# Illustration of the (Semi-)Graphoid Axioms



- Similar to the properties of **separation in graphs**.
- Idea: **Represent conditional independence by separation in graphs.**

# Separation in Graphs

**Definition:** Let  $G = (V, E)$  be an undirected graph and  $X, Y$ , and  $Z$  three disjoint subsets of nodes.  $Z$  **u-separates**  $X$  and  $Y$  in  $G$ , written  $\langle X \mid Z \mid Y \rangle_G$ , iff all paths from a node in  $X$  to a node in  $Y$  contain a node in  $Z$ . A path that contains a node in  $Z$  is called **blocked** (by  $Z$ ), otherwise it is called **active**.

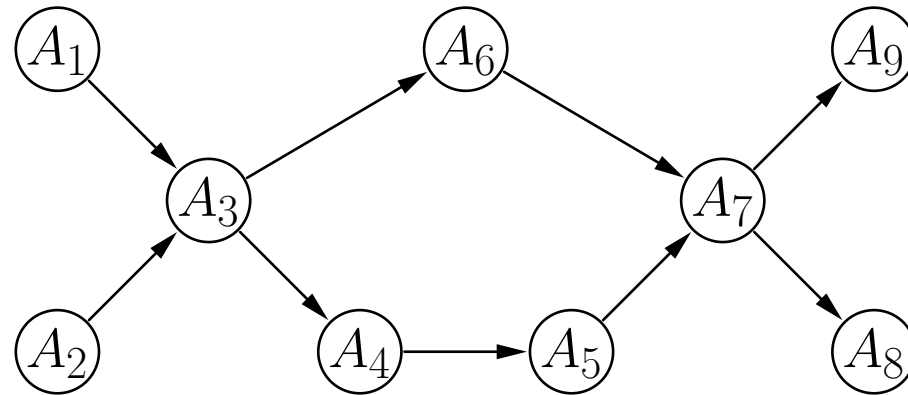
**Definition:** Let  $\vec{G} = (V, \vec{E})$  be a directed acyclic graph and  $X, Y$ , and  $Z$  three disjoint subsets of nodes.  $Z$  **d-separates**  $X$  and  $Y$  in  $\vec{G}$ , written  $\langle X \mid Z \mid Y \rangle_{\vec{G}}$ , iff there is *no* path from a node in  $X$  to a node in  $Y$  along which the following two conditions hold:

1. every node with converging edges either is in  $Z$  or has a descendant in  $Z$ ,
2. every other node is not in  $Z$ .

A path satisfying the two conditions above is said to be **active**, otherwise it is said to be **blocked** (by  $Z$ ).

# Separation in Directed Acyclic Graphs

Example Graph:



Valid Separations:

$$\langle \{A_1\} \mid \{A_3\} \mid \{A_4\} \rangle$$

$$\langle \{A_3\} \mid \{A_4, A_6\} \mid \{A_7\} \rangle$$

$$\langle \{A_8\} \mid \{A_7\} \mid \{A_9\} \rangle$$

$$\langle \{A_1\} \mid \emptyset \mid \{A_2\} \rangle$$

Invalid Separations:

$$\langle \{A_1\} \mid \{A_4\} \mid \{A_2\} \rangle$$

$$\langle \{A_4\} \mid \{A_3, A_7\} \mid \{A_6\} \rangle$$

$$\langle \{A_1\} \mid \{A_6\} \mid \{A_7\} \rangle$$

$$\langle \{A_1\} \mid \{A_4, A_9\} \mid \{A_5\} \rangle$$

# Conditional (In)Dependence Graphs

**Definition:** Let  $(\cdot \perp\!\!\!\perp_{\delta} \cdot \mid \cdot)$  be a three-place relation representing the set of conditional independence statements that hold in a given distribution  $\delta$  over a set  $U$  of attributes. An undirected graph  $G = (U, E)$  over  $U$  is called a **conditional dependence graph** or a **dependence map** w.r.t.  $\delta$ , iff for all disjoint subsets  $X, Y, Z \subseteq U$  of attributes

$$X \perp\!\!\!\perp_{\delta} Y \mid Z \Rightarrow \langle X \mid Z \mid Y \rangle_G,$$

i.e., if  $G$  captures by  $u$ -separation all (conditional) independences that hold in  $\delta$  and thus represents only valid (conditional) dependences. Similarly,  $G$  is called a **conditional independence graph** or an **independence map** w.r.t.  $\delta$ , iff for all disjoint subsets  $X, Y, Z \subseteq U$  of attributes

$$\langle X \mid Z \mid Y \rangle_G \Rightarrow X \perp\!\!\!\perp_{\delta} Y \mid Z,$$

i.e., if  $G$  captures by  $u$ -separation only (conditional) independences that are valid in  $\delta$ .  $G$  is said to be a **perfect map** of the conditional (in)dependences in  $\delta$ , if it is both a dependence map and an independence map.



# Conditional (In)Dependence Graphs

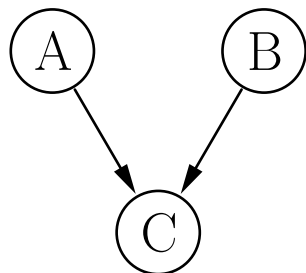
**Definition:** A conditional dependence graph is called **maximal** w.r.t. a distribution  $\delta$  (or, in other words, a **maximal dependence map** w.r.t.  $\delta$ ) iff no edge can be added to it so that the resulting graph is still a conditional dependence graph w.r.t. the distribution  $\delta$ .

**Definition:** A conditional independence graph is called **minimal** w.r.t. a distribution  $\delta$  (or, in other words, a **minimal independence map** w.r.t.  $\delta$ ) iff no edge can be removed from it so that the resulting graph is still a conditional independence graph w.r.t. the distribution  $\delta$ .

- Conditional independence graphs are sometimes required to be minimal.
- However, this requirement is not necessary for a conditional independence graph to be usable for evidence propagation.
- The disadvantage of a non-minimal conditional independence graph is that evidence propagation may be more costly computationally than necessary.

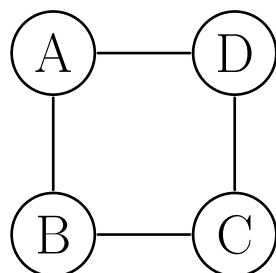
# Limitations of Graph Representations

Perfect directed map, no perfect undirected map:



$p_{ABC}$	$A = a_1$		$A = a_2$	
	$B = b_1$	$B = b_2$	$B = b_1$	$B = b_2$
$C = c_1$	$\frac{4}{24}$	$\frac{3}{24}$	$\frac{3}{24}$	$\frac{2}{24}$
$C = c_2$	$\frac{2}{24}$	$\frac{3}{24}$	$\frac{3}{24}$	$\frac{4}{24}$

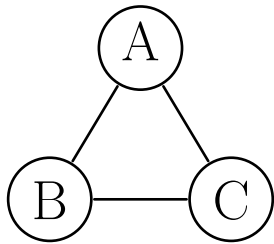
Perfect undirected map, no perfect directed map:



$p_{ABCD}$		$A = a_1$		$A = a_2$	
		$B = b_1$	$B = b_2$	$B = b_1$	$B = b_2$
$C = c_1$	$D = d_1$	$\frac{1}{47}$	$\frac{1}{47}$	$\frac{1}{47}$	$\frac{2}{47}$
	$D = d_2$	$\frac{1}{47}$	$\frac{1}{47}$	$\frac{2}{47}$	$\frac{4}{47}$
$C = c_2$	$D = d_1$	$\frac{1}{47}$	$\frac{2}{47}$	$\frac{1}{47}$	$\frac{4}{47}$
	$D = d_2$	$\frac{2}{47}$	$\frac{4}{47}$	$\frac{4}{47}$	$\frac{16}{47}$

# Limitations of Graph Representations

- There are also probability distributions for which there exists neither a directed nor an undirected perfect map:



$p_{ABC}$	$A = a_1$		$A = a_2$	
	$B = b_1$	$B = b_2$	$B = b_1$	$B = b_2$
$C = c_1$	$2/12$	$1/12$	$1/12$	$2/12$
$C = c_2$	$1/12$	$2/12$	$2/12$	$1/12$

- In such cases either not all dependences or not all independences can be captured by a graph representation.
- In such a situation one usually decides to neglect some of the independence information, that is, to use only a (minimal) conditional independence graph.
- This is sufficient for correct evidence propagation, the existence of a perfect map is not required.

# Markov Properties of Undirected Graphs

**Definition:** An undirected graph  $G = (U, E)$  over a set  $U$  of attributes is said to have (w.r.t. a distribution  $\delta$ ) the

**pairwise Markov property,**

iff in  $\delta$  any pair of attributes which are nonadjacent in the graph are conditionally independent given all remaining attributes, i.e., iff

$$\forall A, B \in U, A \neq B : (A, B) \notin E \Rightarrow A \perp\!\!\!\perp_{\delta} B \mid U - \{A, B\},$$

**local Markov property,**

iff in  $\delta$  any attribute is conditionally independent of all remaining attributes given its neighbors, i.e., iff

$$\forall A \in U : A \perp\!\!\!\perp_{\delta} U - \text{closure}(A) \mid \text{boundary}(A),$$

**global Markov property,**

iff in  $\delta$  any two sets of attributes which are  $u$ -separated by a third are conditionally independent given the attributes in the third set, i.e., iff

$$\forall X, Y, Z \subseteq U : \langle X \mid Z \mid Y \rangle_G \Rightarrow X \perp\!\!\!\perp_{\delta} Y \mid Z.$$

# Markov Properties of Directed Acyclic Graphs

**Definition:** A directed acyclic graph  $\vec{G} = (U, \vec{E})$  over a set  $U$  of attributes is said to have (w.r.t. a distribution  $\delta$ ) the

**pairwise Markov property,**

iff in  $\delta$  any attribute is conditionally independent of any non-descendant not among its parents given all remaining non-descendants, i.e., iff

$$\forall A, B \in U : B \in \text{nondescs}(A) - \text{parents}(A) \Rightarrow A \perp\!\!\!\perp_{\delta} B \mid \text{nondescs}(A) - \{B\},$$

**local Markov property,**

iff in  $\delta$  any attribute is conditionally independent of all remaining non-descendants given its parents, i.e., iff

$$\forall A \in U : A \perp\!\!\!\perp_{\delta} \text{nondescs}(A) - \text{parents}(A) \mid \text{parents}(A),$$

**global Markov property,**

iff in  $\delta$  any two sets of attributes which are  $d$ -separated by a third are conditionally independent given the attributes in the third set, i.e., iff

$$\forall X, Y, Z \subseteq U : \langle X \mid Z \mid Y \rangle_{\vec{G}} \Rightarrow X \perp\!\!\!\perp_{\delta} Y \mid Z.$$

# Equivalence of Markov Properties

**Theorem:** If a three-place relation  $(\cdot \perp\!\!\!\perp_{\delta} \cdot \mid \cdot)$  representing the set of conditional independence statements that hold in a given joint distribution  $\delta$  over a set  $U$  of attributes satisfies the graphoid axioms, then the pairwise, the local, and the global Markov property of an undirected graph  $G = (U, E)$  over  $U$  are equivalent.

**Theorem:** If a three-place relation  $(\cdot \perp\!\!\!\perp_{\delta} \cdot \mid \cdot)$  representing the set of conditional independence statements that hold in a given joint distribution  $\delta$  over a set  $U$  of attributes satisfies the semi-graphoid axioms, then the local and the global Markov property of a directed acyclic graph  $\vec{G} = (U, \vec{E})$  over  $U$  are equivalent.

If  $(\cdot \perp\!\!\!\perp_{\delta} \cdot \mid \cdot)$  satisfies the graphoid axioms, then the pairwise, the local, and the global Markov property are equivalent.

# Markov Equivalence of Graphs

- Can two distinct graphs represent the exactly the same set of conditional independence statements?
- The answer is relevant for learning graphical models from data, because it determines whether we can expect a unique graph as a learning result or not.

**Definition:** Two (directed or undirected) graphs  $G_1 = (U, E_1)$  and  $G_2 = (U, E_2)$  with the same set  $U$  of nodes are called **Markov equivalent** iff they satisfy the same set of node separation statements (with  $d$ -separation for directed graphs and  $u$ -separation for undirected graphs), or formally, iff

$$\forall X, Y, Z \subseteq U : \langle X \mid Z \mid Y \rangle_{G_1} \Leftrightarrow \langle X \mid Z \mid Y \rangle_{G_2}.$$

- No two different *undirected* graphs can be Markov equivalent.
- The reason is that these two graphs, in order to be different, have to differ in at least one edge. However, the graph lacking this edge satisfies a node separation (and thus expresses a conditional independence) that is not satisfied (expressed) by the graph possessing the edge.

# Markov Equivalence of Graphs

**Definition:** Let  $\vec{G} = (U, \vec{E})$  be a directed graph.

The **skeleton** of  $\vec{G}$  is the undirected graph  $G = (V, E)$  where  $E$  contains the same edges as  $\vec{E}$ , but with their directions removed, or formally:

$$E = \{(A, B) \in U \times U \mid (A, B) \in \vec{E} \vee (B, A) \in \vec{E}\}.$$

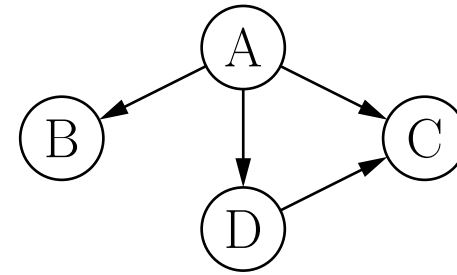
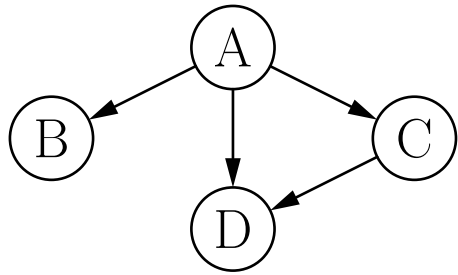
**Definition:** Let  $\vec{G} = (U, \vec{E})$  be a directed graph and  $A, B, C \in U$  three nodes of  $\vec{G}$ . The triple  $(A, B, C)$  is called a **v-structure** of  $\vec{G}$  iff  $(A, B) \in \vec{E}$  and  $(C, B) \in \vec{E}$ , but neither  $(A, C) \in \vec{E}$  nor  $(C, A) \in \vec{E}$ , that is, iff  $\vec{G}$  has converging edges from  $A$  and  $C$  at  $B$ , but  $A$  and  $C$  are unconnected.

**Theorem:** Let  $\vec{G}_1 = (U, \vec{E}_1)$  and  $\vec{G}_2 = (U, \vec{E}_2)$  be two directed acyclic graphs with the same node set  $U$ . The graphs  $\vec{G}_1$  and  $\vec{G}_2$  are Markov equivalent iff they possess the same skeleton and the same set of v-structures.

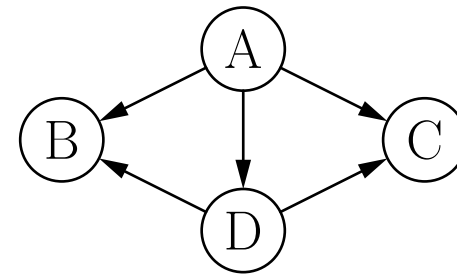
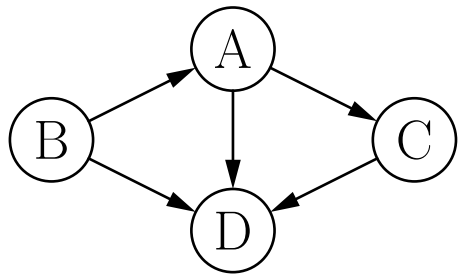
- Intuitively:  
Edge directions may be reversed if this does not change the set of v-structures.



# Markov Equivalence of Graphs



Graphs with the same skeleton, but converging edges at different nodes, which start from *connected* nodes, can be Markov equivalent.



Of several edges that converge at a node only a subset may actually represent a v-structure. This v-structure, however, is relevant.

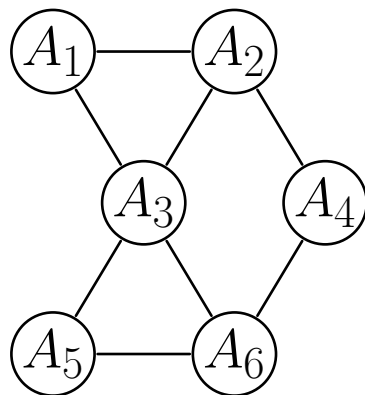
# Undirected Graphs and Decompositions

**Definition:** A probability distribution  $p_V$  over a set  $V$  of variables is called **decomposable** or **factorizable w.r.t. an undirected graph**  $G = (V, E)$  iff it can be written as a product of nonnegative functions on the maximal cliques of  $G$ .

That is, let  $\mathcal{M}$  be a family of subsets of variables, such that the subgraphs of  $G$  induced by the sets  $M \in \mathcal{M}$  are the maximal cliques of  $G$ . Then there exist functions  $\phi_M : \mathcal{E}_M \rightarrow \mathbb{R}_0^+$ ,  $M \in \mathcal{M}$ ,  $\forall a_1 \in \text{dom}(A_1) : \dots \forall a_n \in \text{dom}(A_n)$  :

$$p_V\left(\bigwedge_{A_i \in V} A_i = a_i\right) = \prod_{M \in \mathcal{M}} \phi_M\left(\bigwedge_{A_i \in M} A_i = a_i\right).$$

**Example:**



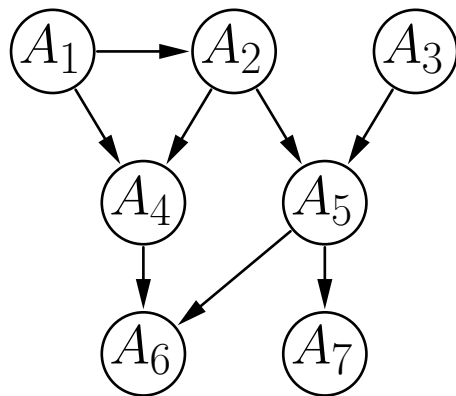
$$\begin{aligned} p_V(A_1 = a_1, \dots, A_6 = a_6) \\ &= \phi_{A_1 A_2 A_3}(A_1 = a_1, A_2 = a_2, A_3 = a_3) \\ &\cdot \phi_{A_3 A_5 A_6}(A_3 = a_3, A_5 = a_5, A_6 = a_6) \\ &\cdot \phi_{A_2 A_4}(A_2 = a_2, A_4 = a_4) \\ &\cdot \phi_{A_4 A_6}(A_4 = a_4, A_6 = a_6). \end{aligned}$$

# Directed Acyclic Graphs and Decompositions

**Definition:** A probability distribution  $p_U$  over a set  $U$  of attributes is called **decomposable** or **factorizable w.r.t. a directed acyclic graph**  $\vec{G} = (U, \vec{E})$  over  $U$ , iff it can be written as a product of the conditional probabilities of the attributes given their parents in  $\vec{G}$ , i.e., iff

$$\forall a_1 \in \text{dom}(A_1) : \dots \forall a_n \in \text{dom}(A_n) : \\ p_U \left( \bigwedge_{A_i \in U} A_i = a_i \right) = \prod_{A_i \in U} P \left( A_i = a_i \mid \bigwedge_{A_j \in \text{parents}_{\vec{G}}(A_i)} A_j = a_j \right).$$

**Example:**



$$\begin{aligned} &P(A_1 = a_1, \dots, A_7 = a_7) \\ &= P(A_1 = a_1) \cdot P(A_2 = a_2 \mid A_1 = a_1) \cdot P(A_3 = a_3) \\ &\quad \cdot P(A_4 = a_4 \mid A_1 = a_1, A_2 = a_2) \\ &\quad \cdot P(A_5 = a_5 \mid A_2 = a_2, A_3 = a_3) \\ &\quad \cdot P(A_6 = a_6 \mid A_4 = a_4, A_5 = a_5) \\ &\quad \cdot P(A_7 = a_7 \mid A_5 = a_5). \end{aligned}$$

# Conditional Independence Graphs and Decompositions

## Core Theorem of Graphical Models:

Let  $p_V$  be a strictly positive probability distribution on a set  $V$  of (discrete) variables. A directed or undirected graph  $G = (V, E)$  is a conditional independence graph w.r.t.  $p_V$  if and only if  $p_V$  is factorizable w.r.t.  $G$ .

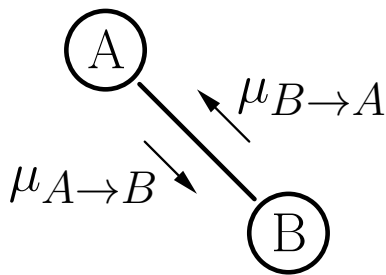
**Definition:** A **Markov network** is an undirected conditional independence graph of a probability distribution  $p_V$  together with the family of positive functions  $\phi_M$  of the factorization induced by the graph.

**Definition:** A **Bayesian network** is a directed conditional independence graph of a probability distribution  $p_U$  together with the family of conditional probabilities of the factorization induced by the graph.

- Sometimes the conditional independence graph is required to be minimal, if it is to be used as the graph underlying a Markov or Bayesian network.
- For correct evidence propagation it is not required that the graph is minimal. Evidence propagation may just be less efficient than possible.

# Probabilistic Graphical Models: Evidence Propagation in Undirected Trees

# Evidence Propagation in Undirected Trees



Node processors communicating by message passing. The messages represent information collected in the corresponding subgraphs.

## Derivation of the Propagation Formulae

Computation of Marginal Distribution:

$$P(A_g = a_g) = \sum_{a_k \in \text{dom}(A_k)}^{\forall A_k \in U - \{A_g\}:} P\left(\bigwedge_{A_i \in U} A_i = a_i\right),$$

Factor Potential Decomposition w.r.t. Undirected Tree:

$$P(A_g = a_g) = \sum_{a_k \in \text{dom}(A_k)}^{\forall A_k \in U - \{A_g\}:} \prod_{(A_i, A_j) \in E} \phi_{A_i A_j}(a_i, a_j).$$

# Evidence Propagation in Undirected Trees

- All factor potentials have only two arguments, because we deal with a tree: the maximal cliques of a tree are simply its edges, as there are no cycles.
- In addition, a tree has the convenient property that by removing an edge it is split into two disconnected subgraphs.
- In order to be able to refer to such subgraphs, we define:

$$U_B^A = \{A\} \cup \{C \in U \mid A \underset{G'}{\approx} C, G' = (U, E - \{(A, B), (B, A)\})\},$$

that is,  $U_B^A$  is the set of those attributes that can still be reached from the attribute  $A$  if the edge  $A-B$  is removed.

- Similarly, we introduce a notation for the edges in these subgraphs, namely

$$E_B^A = E \cap (U_B^A \times U_B^A).$$

- Thus  $G_B^A = (U_B^A, E_B^A)$  is the subgraph containing all attributes that can be reached from the attribute  $B$  through its neighbor  $A$  (including  $A$  itself).

# Evidence Propagation in Undirected Trees

- In the next step we split the product over all edges into individual factors w.r.t. the neighbors of the goal attribute: we write one factor for each neighbor.
- Each of these factors captures the part of the factorization that refers to the subgraph consisting of the attributes that can be reached from the goal attribute through this neighbor, including the factor potential of the edge that connects the neighbor to the goal attribute.
- That is, we write:

$$\begin{aligned} & P(A_g = a_g) \\ &= \sum_{\substack{\forall A_k \in U - \{A_g\}: \\ a_k \in \text{dom}(A_k)}} \prod_{A_h \in \text{neighbors}(A_g)} \left( \phi_{A_g A_h}(a_g, a_h) \prod_{(A_i, A_j) \in E_{A_g}^{A_h}} \phi_{A_i A_j}(a_i, a_j) \right). \end{aligned}$$

- Note that indeed each factor of the outer product in the above formula refers only to attributes in the subgraph that can be reached from the attribute  $A_g$  through the neighbor attribute  $A_h$  defining the factor.



# Evidence Propagation in Undirected Trees

- In the third step it is exploited that terms that are independent of a summation variable can be moved out of the corresponding sum.
- In addition we make use of 
$$\sum_i \sum_j a_i b_j = \left( \sum_i a_i \right) \left( \sum_j b_j \right).$$
- This yields a decomposition of the expression for  $P(A_g = a_g)$  into factors:

$$\begin{aligned} & P(A_g = a_g) \\ &= \prod_{A_h \in \text{neighbors}(A_g)} \left( \sum_{\substack{\forall A_k \in U_{A_g}^{A_h}: \\ a_k \in \text{dom}(A_k)}} \phi_{A_g A_h}(a_g, a_h) \prod_{(A_i, A_j) \in E_{A_g}^{A_h}} \phi_{A_i A_j}(a_i, a_j) \right) \\ &= \prod_{A_h \in \text{neighbors}(A_g)} \mu_{A_h \rightarrow A_g}(A_g = a_g). \end{aligned}$$

- Each factor represents the probabilistic influence of the subgraph that can be reached through the corresponding neighbor  $A_h \in \text{neighbors}(A_g)$ .
- Thus it can be interpreted as a message about this influence sent from  $A_h$  to  $A_g$ .

# Evidence Propagation in Undirected Trees

- With this formula the propagation formula can now easily be derived.
- The key is to consider a single factor of the above product and to compare it to the expression for  $P(A_h = a_h)$  for the corresponding neighbor  $A_h$ , that is, to

$$P(A_h = a_h) = \sum_{a_k \in \text{dom}(A_k)}^{\forall A_k \in U - \{A_h\}:} \prod_{(A_i, A_j) \in E} \phi_{A_i A_j}(a_i, a_j).$$

- Note that this formula is completely analogous to the formula for  $P(A_g = a_g)$  after the first step, that is, after the application of the factorization formula, with the only difference that this formula refers to  $A_h$  instead of  $A_g$ :

$$P(A_g = a_g) = \sum_{a_k \in \text{dom}(A_k)}^{\forall A_k \in U - \{A_g\}:} \prod_{(A_i, A_j) \in E} \phi_{A_i A_j}(a_i, a_j).$$

- We now identify terms that occur in both formulas.

# Evidence Propagation in Undirected Trees

- Exploiting that obviously  $U = U_{A_g}^{A_h} \cup U_{A_h}^{A_g}$  and drawing on the distributive law again, we can easily rewrite this expression as a product with two factors:

$$\begin{aligned}
 P(A_h = a_h) &= \left( \sum_{\substack{\forall A_k \in U_{A_g}^{A_h} - \{A_h\}: \\ a_k \in \text{dom}(A_k)}} \prod_{(A_i, A_j) \in E_{A_g}^{A_h}} \phi_{A_i A_j}(a_i, a_j) \right) \\
 &\cdot \underbrace{\left( \sum_{a_k \in \text{dom}(A_k)} \phi_{A_g A_h}(a_g, a_h) \prod_{(A_i, A_j) \in E_{A_h}^{A_g}} \phi_{A_i A_j}(a_i, a_j) \right)}_{= \mu_{A_g \rightarrow A_h}(A_h = a_h)}.
 \end{aligned}$$

# Evidence Propagation in Undirected Trees

- As a consequence, we obtain the simple expression

$$\begin{aligned} & \mu_{A_h \rightarrow A_g}(A_g = a_g) \\ &= \sum_{a_h \in \text{dom}(A_h)} \left( \phi_{A_g A_h}(a_g, a_h) \cdot \frac{P(A_h = a_h)}{\mu_{A_g \rightarrow A_h}(A_h = a_h)} \right) \\ &= \sum_{a_h \in \text{dom}(A_h)} \left( \phi_{A_g A_h}(a_g, a_h) \prod_{A_i \in \text{neighbors}(A_h) - \{A_g\}} \mu_{A_i \rightarrow A_h}(A_h = a_h) \right). \end{aligned}$$

- This formula is very intuitive:
  - In the upper form it says that all information collected at  $A_k$  (expressed as  $P(A_k = a_k)$ ) should be transferred to  $A_g$ , with the exception of the information that was received from  $A_g$ .
  - In the lower form the formula says that everything coming in through edges other than  $A_g - A_k$  has to be combined and then passed on to  $A_g$ .

# Evidence Propagation in Undirected Trees

- The second form of this formula also provides us with a means to start the message computations.
- Obviously, the value of the message  $\mu_{A_h \rightarrow A_g}(A_g = a_g)$  can immediately be computed if  $A_h$  is a leaf node of the tree. In this case the product has no factors and thus the equation reduces to

$$\mu_{A_h \rightarrow A_g}(A_g = a_g) = \sum_{a_h \in \text{dom}(A_h)} \phi_{A_g A_h}(a_g, a_h).$$

- After all leaves have computed these messages, there must be at least one node, for which messages from all but one neighbor are known.
- This enables this node to compute the message to the neighbor it did not receive a message from.
- After that, there must again be at least one node, which has received messages from all but one neighbor. Hence it can send a message and so on, until all messages have been computed.

# Evidence Propagation in Undirected Trees

- Up to now we have assumed that no evidence has been added to the network, that is, that no attributes have been instantiated.
- However, if attributes are instantiated, the formulae change only slightly.
- We have to add to the joint probability distribution an *evidence factor* for each instantiated attribute:  
if  $U_{\text{obs}}$  is the set of observed (instantiated) attributes, we compute

$$P(A_g = a_g \mid \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})})$$
$$= \alpha \sum_{\forall A_k \in U - \{A_g\}: a_k \in \text{dom}(A_k)} P\left(\bigwedge_{A_i \in U} A_i = a_i\right) \prod_{A_o \in U_{\text{obs}}} \overbrace{\frac{P(A_o = a_o \mid A_o = a_o^{(\text{obs})})}{P(A_o = a_o)}}^{\text{evidence factor for } A_o},$$

where the  $a_o^{(\text{obs})}$  are the observed values and  $\alpha$  is a normalization constant,

$$\alpha = \beta \cdot \prod_{A_j \in U_{\text{obs}}} P(A_j = a_j^{(\text{obs})}) \quad \text{with} \quad \beta = P\left(\bigwedge_{A_j \in U_{\text{obs}}} A_j = a_j^{(\text{obs})}\right)^{-1}.$$

# Evidence Propagation in Undirected Trees

- The justification for this formula is analogous to the justification for the introduction of similar *evidence factors* for the observed attributes in the simple three-attribute example (color/shape/size):

$$\begin{aligned} & P\left(\bigwedge_{A_i \in U} A_i = a_i \mid \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})}\right) \\ &= \beta P\left(\bigwedge_{A_i \in U} A_i = a_i, \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})}\right) \\ &= \begin{cases} \beta P\left(\bigwedge_{A_i \in U} A_i = a_i\right), & \text{if } \forall A_i \in U_{\text{obs}} : a_i = a_i^{(\text{obs})}, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

with  $\beta$  as defined above,

$$\beta = P\left(\bigwedge_{A_j \in U_{\text{obs}}} A_j = a_j^{(\text{obs})}\right)^{-1}.$$

# Evidence Propagation in Undirected Trees

- In addition, it is clear that

$$\forall A_j \in U_{\text{obs}} : P\left(A_j = a_j \mid A_j = a_j^{(\text{obs})}\right) = \begin{cases} 1, & \text{if } a_j = a_j^{(\text{obs})}, \\ 0, & \text{otherwise,} \end{cases}$$

- Therefore we have

$$\prod_{A_j \in U_{\text{obs}}} P\left(A_j = a_j \mid A_j = a_j^{(\text{obs})}\right) = \begin{cases} 1, & \text{if } \forall A_j \in U_{\text{obs}} : a_j = a_j^{(\text{obs})}, \\ 0, & \text{otherwise.} \end{cases}$$

- Combining these equations, we arrive at the formula stated above:

$$\begin{aligned} & P(A_g = a_g \mid \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})}) \\ &= \alpha \sum_{\substack{\forall A_k \in U - \{A_g\}: \\ a_k \in \text{dom}(A_k)}} P\left(\bigwedge_{A_i \in U} A_i = a_i\right) \prod_{A_o \in U_{\text{obs}}} \underbrace{\frac{P(A_o = a_o \mid A_o = a_o^{(\text{obs})})}{P(A_o = a_o)}}_{\text{evidence factor for } A_o}, \end{aligned}$$



# Evidence Propagation in Undirected Trees

- Note that we can neglect the normalization factor  $\alpha$ , because it can always be recovered from the fact that a probability distribution, whether marginal or conditional, must be normalized.
- That is, instead of trying to determine  $\alpha$  beforehand in order to compute  $P(A_g = a_g \mid \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})})$  directly, we confine ourselves to computing  $\frac{1}{\alpha}P(A_g = a_g \mid \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})})$  for all  $a_g \in \text{dom}(A_g)$ .
- Then we determine  $\alpha$  indirectly with the equation

$$\sum_{a_g \in \text{dom}(A_g)} P(A_g = a_g \mid \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})}) = 1.$$

- In other words, the computed values  $\frac{1}{\alpha}P(A_g = a_g \mid \bigwedge_{A_o \in U_{\text{obs}}} A_o = a_o^{(\text{obs})})$  are simply normalized to sum 1 to compute the desired probabilities.

# Evidence Propagation in Undirected Trees

- If the derivation is redone with the modified initial formula for the probability of a value of some goal attribute  $A_g$ , the *evidence factors*  $P(A_o = a_o | A_o = a_o^{(\text{obs})}) / P(A_o = a_o)$  directly influence only the formula for the messages that are sent out from the instantiated attributes.
- Therefore we obtain the following formula for the messages that are sent from an instantiated attribute  $A_o$ :

$$\begin{aligned} & \mu_{A_o \rightarrow A_i}(A_i = a_i) \\ &= \sum_{a_o \in \text{dom}(A_o)} \left( \phi_{A_i A_o}(a_i, a_o) \frac{P(A_o = a_o)}{\mu_{A_i \rightarrow A_o}(A_o = a_o)} \right) \frac{P(A_o = a_o | A_o = a_o^{(\text{obs})})}{P(A_o = a_o)} \\ &= \begin{cases} \gamma \cdot \phi_{A_i A_o}(a_i, a_o^{(\text{obs})}), & \text{if } a_o = a_o^{(\text{obs})}, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

where  $\gamma = 1 / \mu_{A_i \rightarrow A_o}(A_o = a_o^{(\text{obs})})$ .

# Evidence Propagation in Undirected Trees

This formula is again very intuitive:

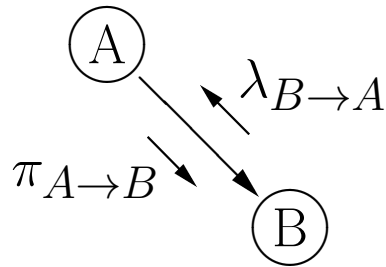
- In an undirected tree, any attribute  $A_o$   $u$ -separates all attributes in a subgraph reached through one of its neighbors from all attributes in a subgraph reached through any other of its neighbors.
- Consequently, if  $A_o$  is instantiated, all paths through  $A_o$  are blocked and thus no information should be passed from one neighbor to any other.
- Note that in an implementation we can neglect  $\gamma$ , because it is the same for all values  $a_i \in \text{dom}(A_i)$  and thus can be incorporated into the constant  $\alpha$ .

Rewriting the Propagation Formulae in Vector Form:

- We need to determine the probability of all values of the goal attribute and we have to evaluate the messages for all values of the attributes that are arguments.
- Therefore it is convenient to write the equations in vector form, with a vector for each attribute that has as many elements as the attribute has values. The factor potentials can then be represented as matrices.

# Probabilistic Graphical Models: Evidence Propagation in Polytrees

# Evidence Propagation in Polytrees



**Idea:** Node processors communicating by message passing:  $\pi$ -messages are sent from parent to child and  $\lambda$ -messages are sent from child to parent.

## Derivation of the Propagation Formulae

Computation of Marginal Distribution:

$$P(A_g = a_g) = \sum_{\substack{\forall A_i \in U - \{A_g\}: \\ a_i \in \text{dom}(A_i)}} P\left(\bigwedge_{A_j \in U} A_j = a_j\right)$$

Chain Rule Factorization w.r.t. the Polytree:

$$P(A_g = a_g) = \sum_{\substack{\forall A_i \in U - \{A_g\}: \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U} P\left(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j\right)$$

# Evidence Propagation in Polytrees (continued)

Decomposition w.r.t. Subgraphs:

$$\begin{aligned}
 P(A_g = a_g) &= \sum_{\substack{\forall A_i \in U - \{A_g\}: \\ a_i \in \text{dom}(A_i)}} \left( P(A_g = a_g \mid \bigwedge_{A_j \in \text{parents}(A_g)} A_j = a_j) \right. \\
 &\quad \cdot \prod_{A_k \in U_+(A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \\
 &\quad \left. \cdot \prod_{A_k \in U_-(A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right).
 \end{aligned}$$

Attribute sets underlying subgraphs:

$$\begin{aligned}
 U_B^A(C) &= \{C\} \cup \{D \in U \mid D \overset{\vec{G}'}{\approx} C, \vec{G}' = (U, E - \{(A, B)\})\}, \\
 U_+(A) &= \bigcup_{C \in \text{parents}(A)} U_A^C(C), & U_+(A, B) &= \bigcup_{C \in \text{parents}(A) - \{B\}} U_A^C(C), \\
 U_-(A) &= \bigcup_{C \in \text{children}(A)} U_C^A(C), & U_-(A, B) &= \bigcup_{C \in \text{children}(A) - \{B\}} U_A^C(C).
 \end{aligned}$$

# Evidence Propagation in Polytrees (continued)

Terms that are independent of a summation variable can be moved out of the corresponding sum. This yields a decomposition into two main factors:

$$\begin{aligned}
 P(A_g = a_g) &= \left( \sum_{\substack{\forall A_i \in \text{parents}(A_g): \\ a_i \in \text{dom}(A_i)}} P(A_g = a_g \mid \bigwedge_{A_j \in \text{parents}(A_g)} A_j = a_j) \right. \\
 &\quad \cdot \left[ \sum_{\substack{\forall A_i \in U_+^*(A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_+(A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \Big) \\
 &\quad \cdot \left[ \sum_{\substack{\forall A_i \in U_-(A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \\
 &= \pi(A_g = a_g) \cdot \lambda(A_g = a_g),
 \end{aligned}$$

where  $U_+^*(A_g) = U_+(A_g) - \text{parents}(A_g)$ .

# Evidence Propagation in Polytrees (continued)

$$\begin{aligned}
 & \sum_{\substack{\forall A_i \in U_+^*(A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_+(A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \\
 &= \prod_{A_p \in \text{parents}(A_g)} \left( \sum_{\substack{\forall A_i \in \text{parents}(A_p): \\ a_i \in \text{dom}(A_i)}} P(A_p = a_p \mid \bigwedge_{A_j \in \text{parents}(A_p)} A_j = a_j) \right. \\
 & \quad \cdot \left[ \sum_{\substack{\forall A_i \in U_+^*(A_p): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_+(A_p)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \\
 & \quad \cdot \left[ \sum_{\substack{\forall A_i \in U_-(A_p, A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_p, A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \\
 &= \prod_{A_p \in \text{parents}(A_g)} \pi(A_p = a_p) \\
 & \quad \cdot \left[ \sum_{\substack{\forall A_i \in U_-(A_p, A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_p, A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right]
 \end{aligned}$$



# Evidence Propagation in Polytrees (continued)

$$\begin{aligned}
 & \sum_{\substack{\forall A_i \in U_+^*(A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_+(A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \\
 &= \prod_{A_p \in \text{parents}(A_g)} \pi(A_p = a_p) \\
 & \cdot \left[ \sum_{\substack{\forall A_i \in U_-(A_p, A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_p, A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \\
 &= \prod_{A_p \in \text{parents}(A_g)} \pi_{A_p \rightarrow A_g}(A_p = a_p)
 \end{aligned}$$

$$\begin{aligned}
 \pi(A_g = a_g) &= \sum_{\substack{\forall A_i \in \text{parents}(A_g): \\ a_i \in \text{dom}(A_i)}} P(A_g = a_g \mid \bigwedge_{A_j \in \text{parents}(A_g)} A_j = a_j) \\
 & \cdot \prod_{A_p \in \text{parents}(A_g)} \pi_{A_p \rightarrow A_g}(A_p = a_p)
 \end{aligned}$$

# Evidence Propagation in Polytrees (continued)

$$\begin{aligned}
 \lambda(A_g = a_g) &= \sum_{\substack{\forall A_i \in U_-(A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \\
 &= \prod_{A_c \in \text{children}(A_g)} \sum_{a_c \in \text{dom}(A_c)} \left( \sum_{\substack{\forall A_i \in \text{parents}(A_c) - \{A_g\}: \\ a_i \in \text{dom}(A_i)}} P(A_c = a_c \mid \bigwedge_{A_j \in \text{parents}(A_c)} A_j = a_j) \right) \\
 &\quad \cdot \left[ \sum_{\substack{\forall A_i \in U_+^*(A_c, A_g): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_+(A_c, A_g)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \\
 &\quad \cdot \left[ \sum_{\substack{\forall A_i \in U_-(A_c): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_c)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \\
 &\hspace{15em} \underbrace{\hspace{15em}} = \lambda(A_c = a_c) \\
 &= \prod_{A_c \in \text{children}(A_g)} \lambda_{A_c \rightarrow A_g}(A_g = a_g)
 \end{aligned}$$

# Propagation Formulae without Evidence

$$\begin{aligned}
 & \pi_{A_p \rightarrow A_c}(A_p = a_p) \\
 &= \pi(A_p = a_p) \cdot \left[ \sum_{\substack{\forall A_i \in U_-(A_p, A_c): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_p, A_c)} P(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j) \right] \\
 &= \frac{P(A_p = a_p)}{\lambda_{A_c \rightarrow A_p}(A_p = a_p)}
 \end{aligned}$$

$$\begin{aligned}
 & \lambda_{A_c \rightarrow A_p}(A_p = a_p) \\
 &= \sum_{a_c \in \text{dom}(A_c)} \lambda(A_c = a_c) \sum_{\substack{\forall A_i \in \text{parents}(A_c) - \{A_p\}: \\ a_i \in \text{dom}(A_k)}} P(A_c = a_c \mid \bigwedge_{A_j \in \text{parents}(A_c)} A_j = a_j) \\
 & \quad \cdot \prod_{A_k \in \text{parents}(A_c) - \{A_p\}} \pi_{A_k \rightarrow A_p}(A_k = a_k)
 \end{aligned}$$

# Evidence Propagation in Polytrees (continued)

**Evidence:** The attributes in a set  $X_{\text{obs}}$  are observed.

$$\begin{aligned} & P\left(A_g = a_g \mid \bigwedge_{A_k \in X_{\text{obs}}} A_k = a_k^{(\text{obs})}\right) \\ &= \sum_{\substack{\forall A_i \in U - \{A_g\}: \\ a_i \in \text{dom}(A_i)}} P\left(\bigwedge_{A_j \in U} A_j = a_j \mid \bigwedge_{A_k \in X_{\text{obs}}} A_k = a_k^{(\text{obs})}\right) \\ &= \alpha \sum_{\substack{\forall A_i \in U - \{A_g\}: \\ a_i \in \text{dom}(A_i)}} P\left(\bigwedge_{A_j \in U} A_j = a_j\right) \prod_{A_k \in X_{\text{obs}}} P\left(A_k = a_k \mid A_k = a_k^{(\text{obs})}\right), \end{aligned}$$

where  $\alpha = \frac{1}{P\left(\bigwedge_{A_k \in X_{\text{obs}}} A_k = a_k^{(\text{obs})}\right)}$

# Propagation Formulae with Evidence

$$\begin{aligned} & \pi_{A_p \rightarrow A_c}(A_p = a_p) \\ &= P\left(A_p = a_p \mid A_p = a_p^{(\text{obs})}\right) \cdot \pi(A_p = a_p) \\ & \cdot \left[ \sum_{\substack{\forall A_i \in U_-(A_p, A_c): \\ a_i \in \text{dom}(A_i)}} \prod_{A_k \in U_-(A_p, A_c)} P\left(A_k = a_k \mid \bigwedge_{A_j \in \text{parents}(A_k)} A_j = a_j\right) \right] \\ &= \begin{cases} \beta, & \text{if } a_p = a_p^{(\text{obs})}, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

- The value of  $\beta$  is not explicitly determined. Usually a value of 1 is used and the correct value is implicitly determined later by normalizing the resulting probability distribution for  $A_g$ .

# Propagation Formulae with Evidence

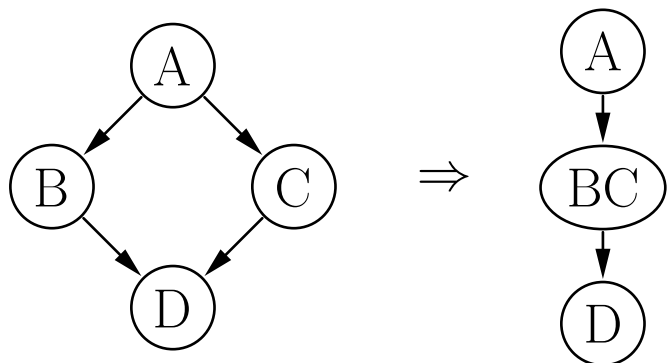
$$\begin{aligned} & \lambda_{A_c \rightarrow A_p}(A_p = a_p) \\ &= \sum_{a_c \in \text{dom}(A_c)} P\left(A_c = a_c \mid A_c = a_c^{(\text{obs})}\right) \cdot \lambda(A_c = a_c) \\ & \quad \cdot \sum_{\substack{\forall A_i \in \text{parents}(A_c) - \{A_p\}: \\ a_i \in \text{dom}(A_k)}} P\left(A_c = a_c \mid \bigwedge_{A_j \in \text{parents}(A_c)} A_j = a_j\right) \\ & \quad \cdot \prod_{A_k \in \text{parents}(A_c) - \{A_p\}} \pi_{A_k \rightarrow A_c}(A_k = a_k) \end{aligned}$$

# Probabilistic Graphical Models: Evidence Propagation in Multiply Connected Networks

# Propagation in Multiply Connected Networks

- Multiply connected networks pose a problem:
  - There are several ways on which information can travel from one attribute (node) to another.
  - As a consequence, the same evidence may be used twice to update the probability distribution of an attribute.
  - Since probabilistic update is not idempotent, multiple inclusion of the same evidence usually invalidates the result.
- General idea to solve this problem:

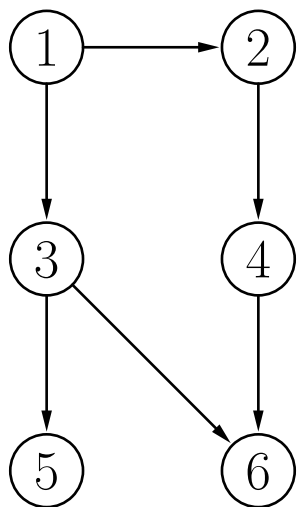
**Transform network into a singly connected structure.**



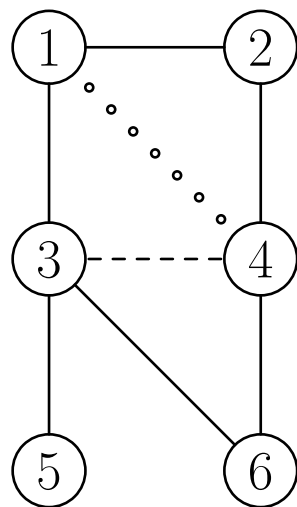
Merging attributes can make the polytree algorithm applicable in multiply connected networks.



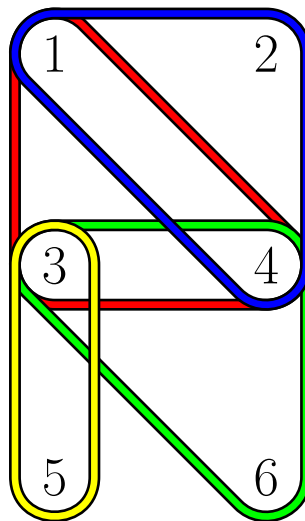
# Triangulation and Join Tree Construction



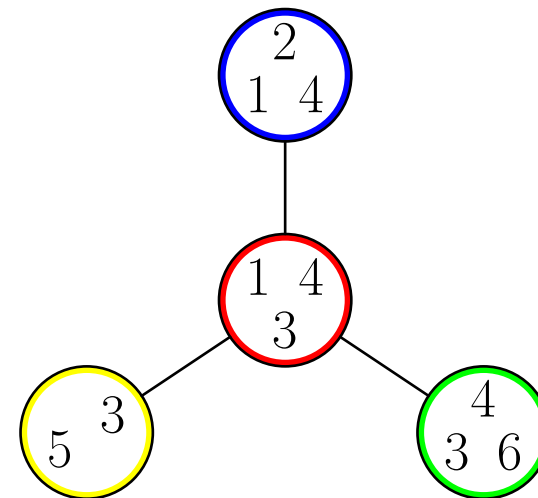
original graph



triangulated moral graph



maximal cliques



join tree

- A singly connected structure is obtained by triangulating the graph and then forming a tree of maximal cliques, the so-called **join tree**.
- For evidence propagation a join tree is enhanced by so-called **separators** on the edges, which are intersection of the connected nodes → **junction tree**.

# Graph Triangulation

**Algorithm:** Graph Triangulation

**Input:** An undirected graph  $G = (V, E)$ .

**Output:** A triangulated undirected graph  $G' = (V, E')$  with  $E' \supseteq E$ .

1. Compute an ordering of the nodes of the graph using *maximum cardinality search*. That is, number the nodes from 1 to  $n = |V|$ , in increasing order, always assigning the next number to the node having the largest set of previously numbered neighbors (breaking ties arbitrarily).
2. From  $i = n = |V|$  to  $i = 1$  recursively fill in edges between any nonadjacent neighbors of the node numbered  $i$  that have lower ranks than  $i$  (including neighbors linked to the node numbered  $i$  in previous steps). If no edges are added to the graph  $G$ , then the original graph  $G$  is triangulated; otherwise the new graph (with the added edges) is triangulated.

# Join Tree Construction

**Algorithm:** Join Tree Construction

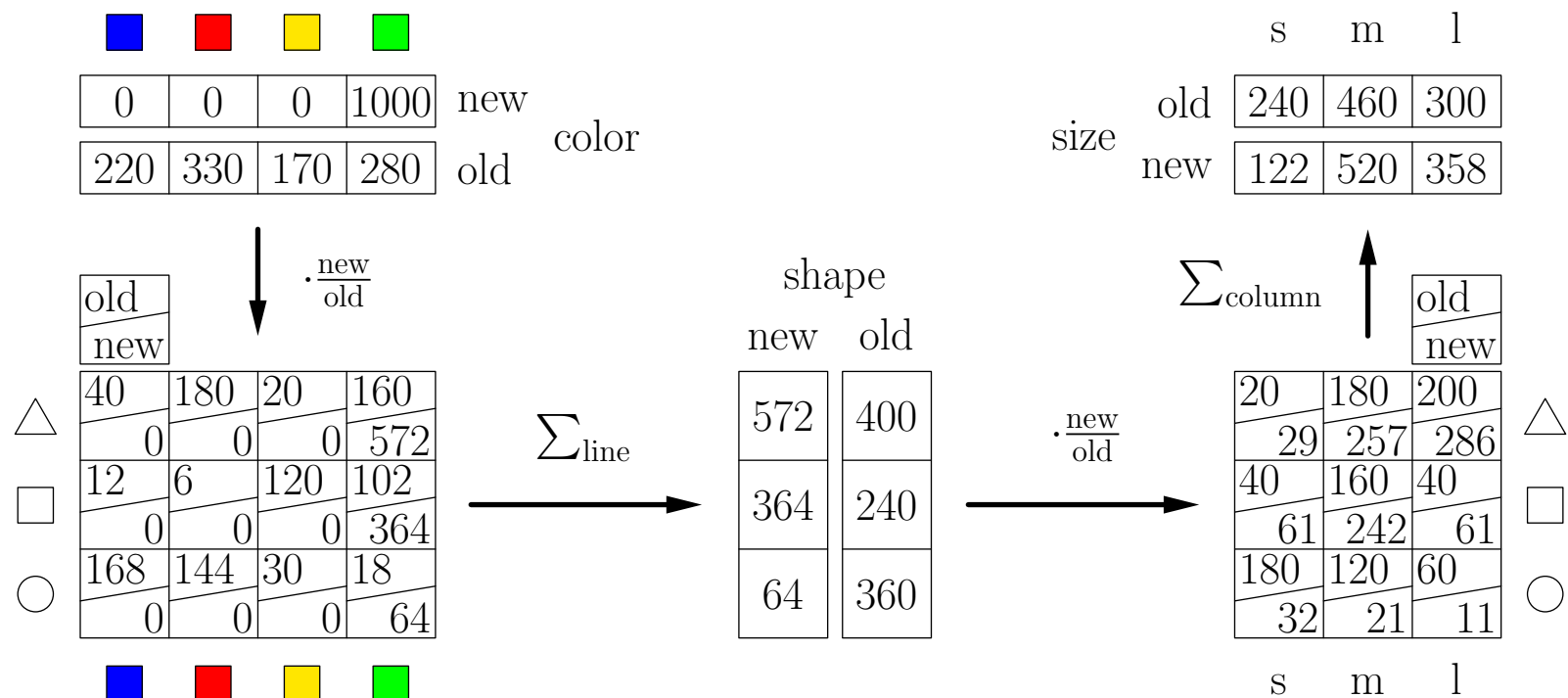
**Input:** A triangulated undirected graph  $G = (V, E)$ .

**Output:** A join tree  $G' = (V', E')$  for  $G$ .

1. Find all maximal cliques  $C_1, \dots, C_k$  of the input graph  $G$  and thus form the set  $V'$  of vertices of the graph  $G'$  (each maximal clique is a node).
2. Form the set  $E^* = \{(C_i, C_j) \mid C_i \cap C_j \neq \emptyset\}$  of candidate edges and assign to each edge the size of the intersection of the connected maximal cliques as a weight, that is, set  $w((C_i, C_j)) = |C_i \cap C_j|$ .
3. Form a maximum spanning tree from the edges in  $E^*$  w.r.t. the weight  $w$ , using, for example, the algorithms proposed by [Kruskal 1956, Prim 1957]. The edges of this maximum spanning tree are the edges in  $E'$ .

# Reasoning in Join/Junction Trees

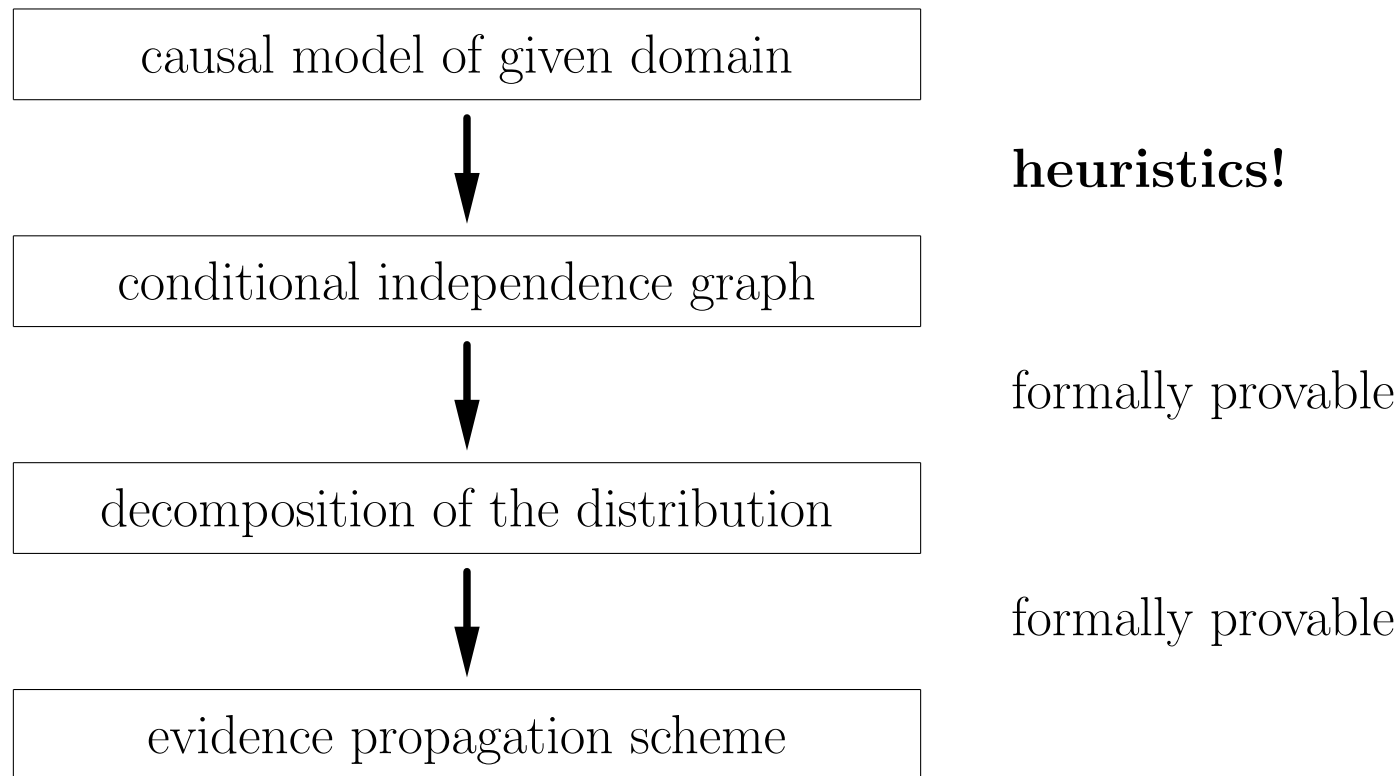
- Reasoning in join trees follows the same lines as for undirected trees.
- Multiple pieces of evidence from different branches may be incorporated into a distribution before continuing by summing/marginalizing.



# Graphical Models: Manual Model Building

# Building Graphical Models: Causal Modeling

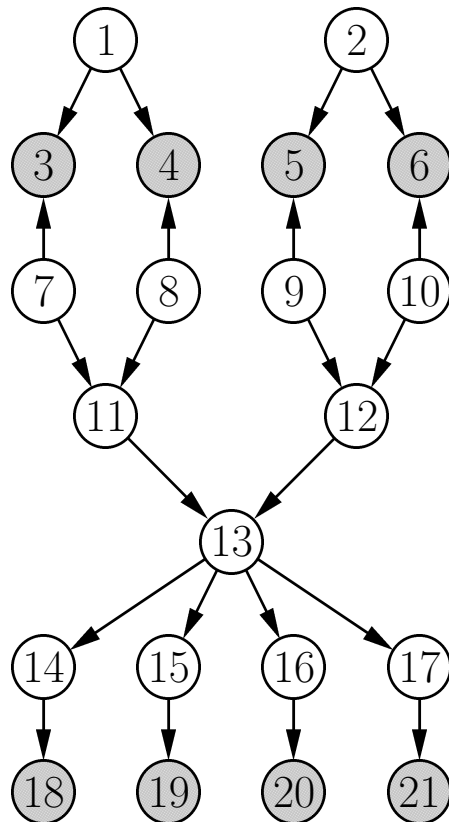
Manual creation of a reasoning system based on a graphical model:



- Problem: strong assumptions about the statistical effects of causal relations.
- Nevertheless this approach often yields usable graphical models.

# Probabilistic Graphical Models: An Example

## Danish Jersey Cattle Blood Type Determination



21 attributes:

- |                          |                         |
|--------------------------|-------------------------|
| 1 – dam correct?         | 11 – offspring ph.gr. 1 |
| 2 – sire correct?        | 12 – offspring ph.gr. 2 |
| 3 – stated dam ph.gr. 1  | 13 – offspring genotype |
| 4 – stated dam ph.gr. 2  | 14 – factor 40          |
| 5 – stated sire ph.gr. 1 | 15 – factor 41          |
| 6 – stated sire ph.gr. 2 | 16 – factor 42          |
| 7 – true dam ph.gr. 1    | 17 – factor 43          |
| 8 – true dam ph.gr. 2    | 18 – lysis 40           |
| 9 – true sire ph.gr. 1   | 19 – lysis 41           |
| 10 – true sire ph.gr. 2  | 20 – lysis 42           |
|                          | 21 – lysis 43           |

The grey nodes correspond to observable attributes.

- This graph was specified by human domain experts, based on knowledge about (causal) dependences of the variables.

# Probabilistic Graphical Models: An Example

## Danish Jersey Cattle Blood Type Determination

- Full 21-dimensional domain has  $2^6 \cdot 3^{10} \cdot 6 \cdot 8^4 = 92\,876\,046\,336$  possible states.
- Bayesian network requires only 306 conditional probabilities.
- Example of a conditional probability table (attributes 2, 9, and 5):

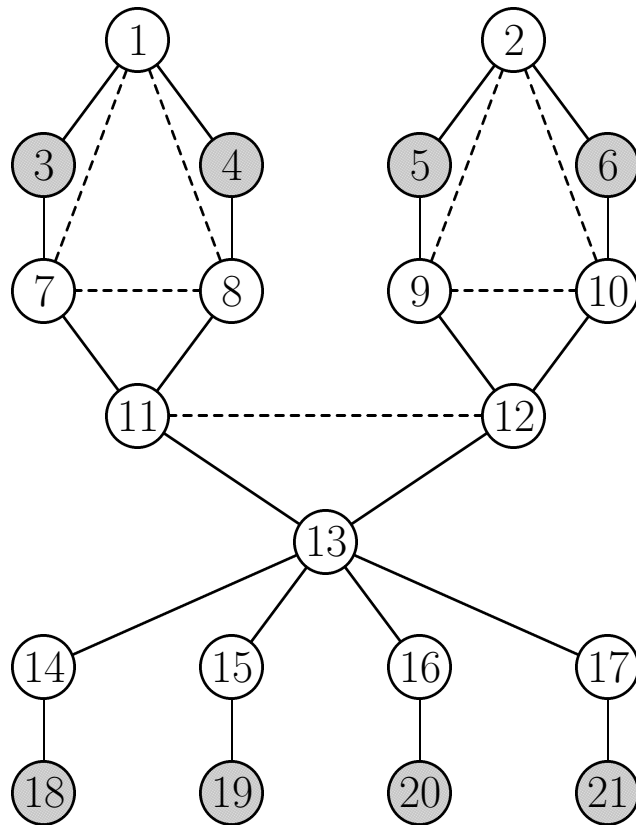
sire correct	true sire phenogroup 1	stated sire phenogroup 1		
		F1	V1	V2
yes	F1	1	0	0
yes	V1	0	1	0
yes	V2	0	0	1
no	F1	0.58	0.10	0.32
no	V1	0.58	0.10	0.32
no	V2	0.58	0.10	0.32

- The probabilities are acquired from human domain experts or estimated from historical data.

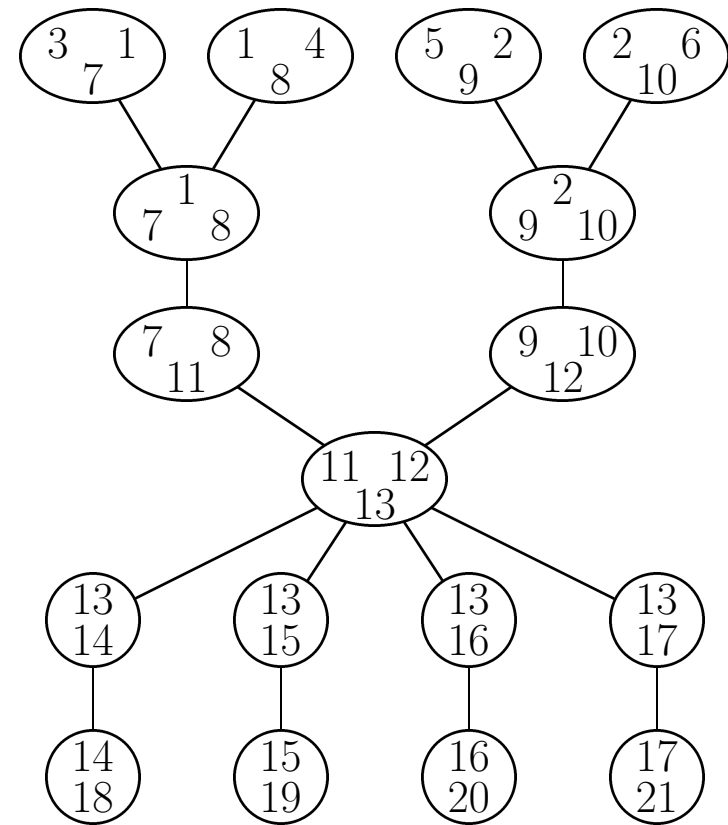


# Probabilistic Graphical Models: An Example

## Danish Jersey Cattle Blood Type Determination



**moral graph**  
(already triangulated)



**join tree**

# Graphical Models and Causality

# Graphical Models and Causality



causal chain

Example:

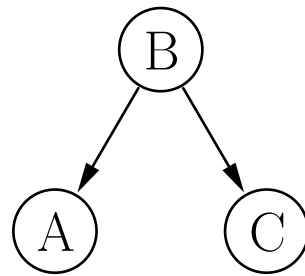
A – accelerator pedal

B – fuel supply

C – engine speed

$$A \not\perp C \mid \emptyset$$

$$A \perp C \mid B$$



common cause

Example:

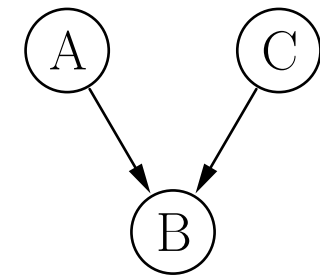
A – ice cream sales

B – temperature

C – bathing accidents

$$A \not\perp C \mid \emptyset$$

$$A \perp C \mid B$$



common effect

Example:

A – influenza

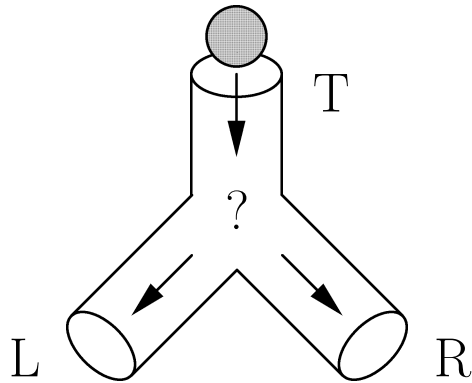
B – fever

C – measles

$$A \perp C \mid \emptyset$$

$$A \not\perp C \mid B$$

# Common Cause Assumption (Causal Markov Assumption)



Y-shaped tube arrangement into which a ball is dropped ( $T$ ). Since the ball can reappear *either* at the left outlet ( $L$ ) *or* the right outlet ( $R$ ) the corresponding variables are dependent.

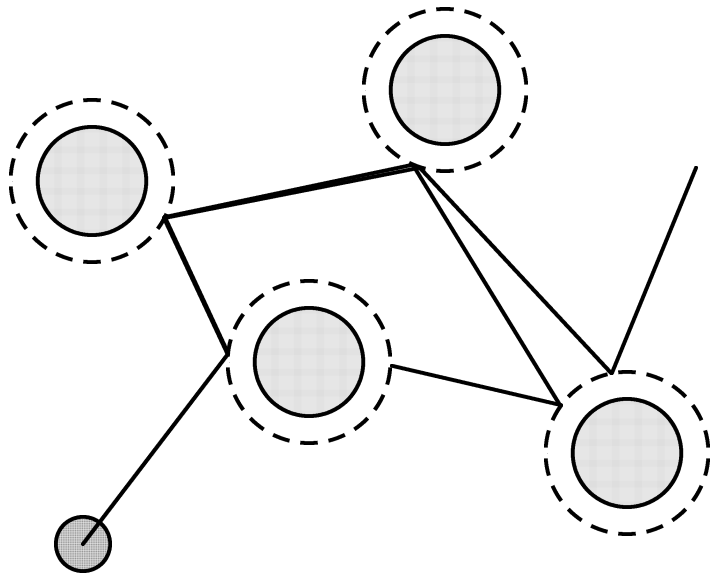
Counter argument: The cause is insufficiently described. If the exact shape, position and velocity of the ball and the tubes are known, the outlet can be determined and the variables become independent.

$t$	$r$	$\bar{r}$	$\Sigma$
$l$	0	$1/2$	$1/2$
$\bar{l}$	$1/2$	0	$1/2$
$\Sigma$	$1/2$	$1/2$	

Counter counter argument: Quantum mechanics states that location and momentum of a particle cannot both at the same time be measured with arbitrary precision.

# Sensitive Dependence on the Initial Conditions

- *Sensitive dependence on the initial conditions* means that a small change of the initial conditions (e.g. a change of the initial position or velocity of a particle) causes a deviation that grows *exponentially* with time.
- Many physical systems show, for arbitrary initial conditions, a sensitive dependence on the initial conditions. Due to this quantum mechanical effects sometimes have macroscopic consequences.



**Example:** Billiard with round (or generally convex) obstacles.

Initial imprecision:  $\approx \frac{1}{100}$  degree

after four collisions:  $\approx 100$  degrees

# Learning Graphical Models from Data

# Learning Graphical Models from Data

Given: A database of sample cases from a domain of interest.

Desired: A (good) graphical model of the domain of interest.

- **Quantitative or Parameter Learning**

- The structure of the conditional independence graph is known.
- Conditional or marginal distributions have to be estimated by standard statistical methods. (*parameter estimation*)

- **Qualitative or Structural Learning**

- The structure of the conditional independence graph is not known.
- A good graph has to be selected from the set of all possible graphs. (*model selection*)
- Tradeoff between model complexity and model accuracy.
- Algorithms consist of a search scheme (*which graphs are considered?*) and a scoring function (*how good is a given graph?*).

# Danish Jersey Cattle Blood Type Determination

A fraction of the database of sample cases:

y	y	f1	v2	f1	v2	f1	v2	f1	v2	v2	v2	v2v2	n	y	n	y	0	6	0	6
y	y	f1	v2	**	**	f1	v2	**	**	**	**	f1v2	y	y	n	y	7	6	0	7
y	y	f1	v2	f1	f1	f1	v2	f1	f1	f1	f1	f1f1	y	y	n	n	7	7	0	0
y	y	f1	v2	f1	f1	f1	v2	f1	f1	f1	f1	f1f1	y	y	n	n	7	7	0	0
y	y	f1	v2	f1	v1	f1	v2	f1	v1	v2	f1	f1v2	y	y	n	y	7	7	0	7
y	y	f1	f1	**	**	f1	f1	**	**	f1	f1	f1f1	y	y	n	n	6	6	0	0
y	y	f1	v1	**	**	f1	v1	**	**	v1	v2	v1v2	n	y	y	y	0	5	4	5
y	y	f1	v2	f1	v1	f1	v2	f1	v1	f1	v1	f1v1	y	y	y	y	7	7	6	7
					⋮											⋮				

- 21 attributes
- 500 real world sample cases
- A lot of missing values (indicated by \*\*)



# Learning Graphical Models from Data: Learning the Parameters

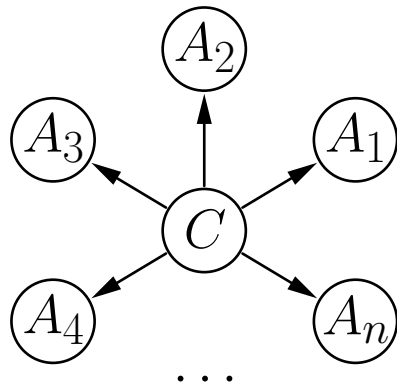
# Learning the Parameters of a Graphical Model

Given: A database of sample cases from a domain of interest.  
The graph underlying a graphical model for the domain.

Desired: Good values for the numeric parameters of the model.

## Example: Naive Bayes Classifiers

- A naive Bayes classifier is a Bayesian network with a star-like structure.
- The class attribute is the only unconditioned attribute.
- All other attributes are conditioned on the class only.



The structure of a naive Bayes classifier is fixed once the attributes have been selected. The only remaining task is to estimate the parameters of the needed probability distributions.

# Probabilistic Classification

- A classifier is an algorithm that assigns a class from a predefined set to a case or object, based on the values of descriptive attributes.
- An optimal classifier maximizes the probability of a correct class assignment.
  - Let  $C$  be a class attribute with  $\text{dom}(C) = \{c_1, \dots, c_{n_C}\}$ , which occur with probabilities  $p_i$ ,  $1 \leq i \leq n_C$ .
  - Let  $q_i$  be the probability with which a classifier assigns class  $c_i$ . ( $q_i \in \{0, 1\}$  for a deterministic classifier)
  - The probability of a correct assignment is

$$P(\text{correct assignment}) = \sum_{i=1}^{n_C} p_i q_i.$$

- Therefore the best choice for the  $q_i$  is

$$q_i = \begin{cases} 1, & \text{if } p_i = \max_{k=1}^{n_C} p_k, \\ 0, & \text{otherwise.} \end{cases}$$

# Probabilistic Classification (continued)

- Consequence: An optimal classifier should assign the **most probable class**.
- This argument does not change if we take descriptive attributes into account.
  - Let  $U = \{A_1, \dots, A_m\}$  be a set of descriptive attributes with domains  $\text{dom}(A_k)$ ,  $1 \leq k \leq m$ .
  - Let  $A_1 = a_1, \dots, A_m = a_m$  be an instantiation of the descriptive attributes.
  - An optimal classifier should assign the class  $c_i$  for which

$$P(C = c_i \mid A_1 = a_1, \dots, A_m = a_m) = \max_{j=1}^{n_C} P(C = c_j \mid A_1 = a_1, \dots, A_m = a_m)$$

- **Problem:** We cannot store a class (or the class probabilities) for every possible instantiation  $A_1 = a_1, \dots, A_m = a_m$  of the descriptive attributes. (The table size grows exponentially with the number of attributes.)
- Therefore: **Simplifying assumptions are necessary.**

# Bayes' Rule and Bayes' Classifiers

- Bayes' rule is a formula that can be used to “invert” conditional probabilities: Let  $X$  and  $Y$  be events,  $P(X) > 0$ . Then

$$P(Y | X) = \frac{P(X | Y) \cdot P(Y)}{P(X)}.$$

- Bayes' rule follows directly from the definition of conditional probability:

$$P(Y | X) = \frac{P(X \cap Y)}{P(X)} \quad \text{and} \quad P(X | Y) = \frac{P(X \cap Y)}{P(Y)}.$$

- Bayes' classifiers: Compute the class probabilities as

$$P(C = c_i | A_1 = a_1, \dots, A_m = a_m) = \frac{P(A_1 = a_1, \dots, A_m = a_m | C = c_i) \cdot P(C = c_i)}{P(A_1 = a_1, \dots, A_m = a_m)}.$$

- Looks unreasonable at first sight: Even more probabilities to store.

# Naive Bayes Classifiers

## Naive Assumption:

*The descriptive attributes are conditionally independent given the class.*

## Bayes' Rule:

$$P(C = c_i | \vec{a}) = \frac{P(A_1 = a_1, \dots, A_m = a_m | C = c_i) \cdot P(C = c_i)}{P(A_1 = a_1, \dots, A_m = a_m)} \quad \leftarrow p_0 = P(\vec{a})$$

## Chain Rule of Probability:

$$P(C = c_i | \vec{a}) = \frac{P(C = c_i)}{p_0} \cdot \prod_{k=1}^m P(A_k = a_k | A_1 = a_1, \dots, A_{k-1} = a_{k-1}, C = c_i)$$

## Conditional Independence Assumption:

$$P(C = c_i | \vec{a}) = \frac{P(C = c_i)}{p_0} \cdot \prod_{k=1}^m P(A_k = a_k | C = c_i)$$

# Naive Bayes Classifiers (continued)

**Consequence:** Manageable amount of data to store.

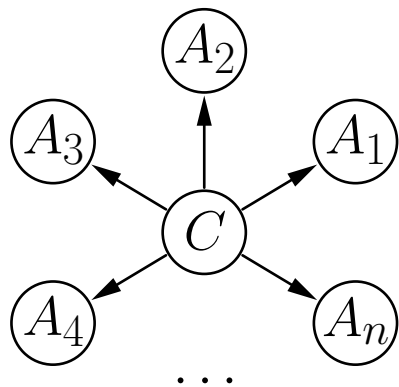
Store distributions  $P(C = c_i)$  and  $\forall 1 \leq j \leq m : P(A_j = a_j | C = c_i)$ .

**Classification:** Compute for all classes  $c_i$

$$P(C = c_i | A_1 = a_1, \dots, A_m = a_m) \cdot p_0 = P(C = c_i) \cdot \prod_{j=1}^n P(A_j = a_j | C = c_i)$$

and predict the class  $c_i$  for which this value is largest.

**Relation to Bayesian Networks:**



Decomposition formula:

$$P(C = c_i, A_1 = a_1, \dots, A_n = a_n) = P(C = c_i) \cdot \prod_{j=1}^n P(A_j = a_j | C = c_i)$$

# Naive Bayes Classifiers: Parameter Estimation

## Estimation of Probabilities:

- **Nominal/Categorical Attributes:**

$$\hat{P}(A_j = a_j \mid C = c_i) = \frac{\#(A_j = a_j, C = c_i) + \gamma}{\#(C = c_i) + n_{A_j}\gamma}$$

$\#(\varphi)$  is the number of example cases that satisfy the condition  $\varphi$ .

$n_{A_j}$  is the number of values of the attribute  $A_j$ .

- $\gamma$  is called **Laplace correction**.

$\gamma = 0$ : Maximum likelihood estimation.

Common choices:  $\gamma = 1$  or  $\gamma = \frac{1}{2}$ .

- Laplace correction helps to avoid problems with attribute values that do not occur with some class in the given data.

It also introduces a bias towards a uniform distribution.



# Naive Bayes Classifiers: Parameter Estimation

## Estimation of Probabilities:

- **Metric/Numeric Attributes:** Assume a normal distribution.

$$P(A_j = a_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_j(c_i)} \exp\left(-\frac{(a_j - \mu_j(c_i))^2}{2\sigma_j^2(c_i)}\right)$$

- Estimate of mean value

$$\hat{\mu}_j(c_i) = \frac{1}{\#(C = c_i)} \sum_{k=1}^{\#(C=c_i)} a_j(k)$$

- Estimate of variance

$$\hat{\sigma}_j^2(c_i) = \frac{1}{\xi} \sum_{j=1}^{\#(C=c_i)} (a_j(k) - \hat{\mu}_j(c_i))^2$$

$\xi = \#(C = c_i)$  : Maximum likelihood estimation

$\xi = \#(C = c_i) - 1$ : Unbiased estimation

# Naive Bayes Classifiers: Simple Example 1

No	Sex	Age	Blood pr.	Drug
1	male	20	normal	A
2	female	73	normal	B
3	female	37	high	A
4	male	33	low	B
5	female	48	high	A
6	male	29	normal	A
7	female	52	normal	B
8	male	42	low	B
9	male	61	normal	B
10	female	30	normal	A
11	female	26	low	B
12	male	54	high	A

$P(\text{Drug})$	$A$	$B$
	0.5	0.5

$P(\text{Sex} \mid \text{Drug})$	$A$	$B$
male	0.5	0.5
female	0.5	0.5

$P(\text{Age} \mid \text{Drug})$	$A$	$B$
$\mu$	36.3	47.8
$\sigma^2$	161.9	311.0

$P(\text{Blood Pr.} \mid \text{Drug})$	$A$	$B$
low	0	0.5
normal	0.5	0.5
high	0.5	0

A simple database and estimated (conditional) probability distributions.

# Naive Bayes Classifiers: Simple Example 1

$$\begin{aligned} &P(\text{Drug A} \mid \text{male}, 61, \text{normal}) \\ &= c_1 \cdot P(\text{Drug A}) \cdot P(\text{male} \mid \text{Drug A}) \cdot P(61 \mid \text{Drug A}) \cdot P(\text{normal} \mid \text{Drug A}) \\ &\approx c_1 \cdot 0.5 \cdot 0.5 \cdot 0.004787 \cdot 0.5 = c_1 \cdot 5.984 \cdot 10^{-4} = 0.219 \end{aligned}$$

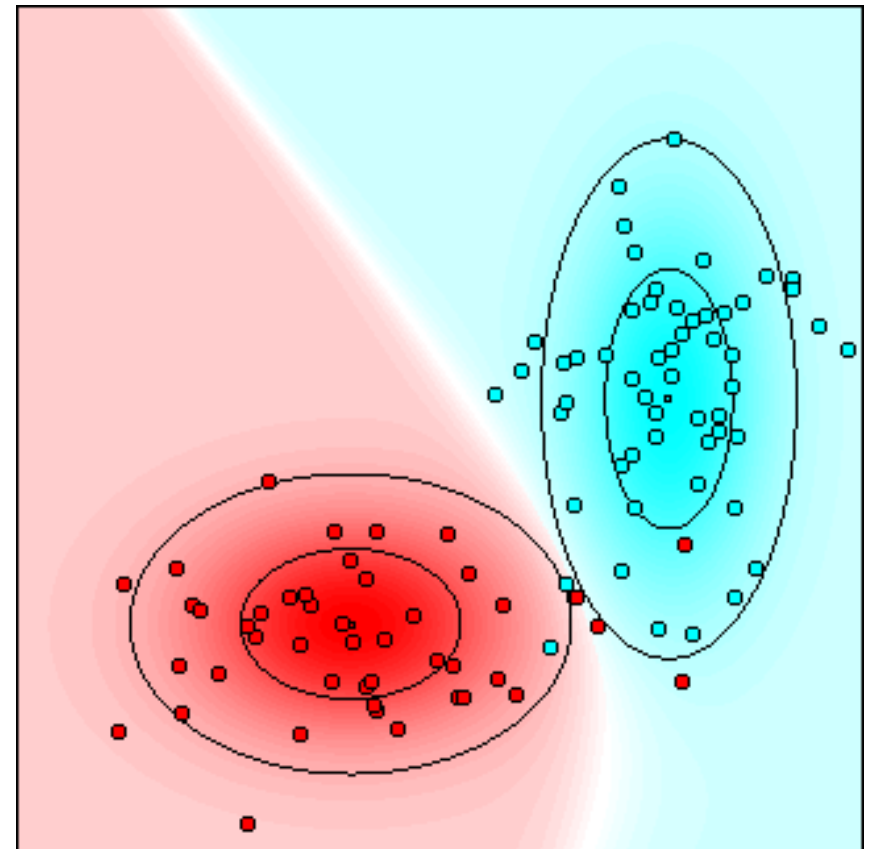
$$\begin{aligned} &P(\text{Drug B} \mid \text{male}, 61, \text{normal}) \\ &= c_1 \cdot P(\text{Drug B}) \cdot P(\text{male} \mid \text{Drug B}) \cdot P(61 \mid \text{Drug B}) \cdot P(\text{normal} \mid \text{Drug B}) \\ &\approx c_1 \cdot 0.5 \cdot 0.5 \cdot 0.017120 \cdot 0.5 = c_1 \cdot 2.140 \cdot 10^{-3} = 0.781 \end{aligned}$$

$$\begin{aligned} &P(\text{Drug A} \mid \text{female}, 30, \text{normal}) \\ &= c_2 \cdot P(\text{Drug A}) \cdot P(\text{female} \mid \text{Drug A}) \cdot P(30 \mid \text{Drug A}) \cdot P(\text{normal} \mid \text{Drug A}) \\ &\approx c_2 \cdot 0.5 \cdot 0.5 \cdot 0.027703 \cdot 0.5 = c_2 \cdot 3.471 \cdot 10^{-3} = 0.671 \end{aligned}$$

$$\begin{aligned} &P(\text{Drug B} \mid \text{female}, 30, \text{normal}) \\ &= c_2 \cdot P(\text{Drug B}) \cdot P(\text{female} \mid \text{Drug B}) \cdot P(30 \mid \text{Drug B}) \cdot P(\text{normal} \mid \text{Drug B}) \\ &\approx c_2 \cdot 0.5 \cdot 0.5 \cdot 0.013567 \cdot 0.5 = c_2 \cdot 1.696 \cdot 10^{-3} = 0.329 \end{aligned}$$

# Naive Bayes Classifiers: Simple Example 2

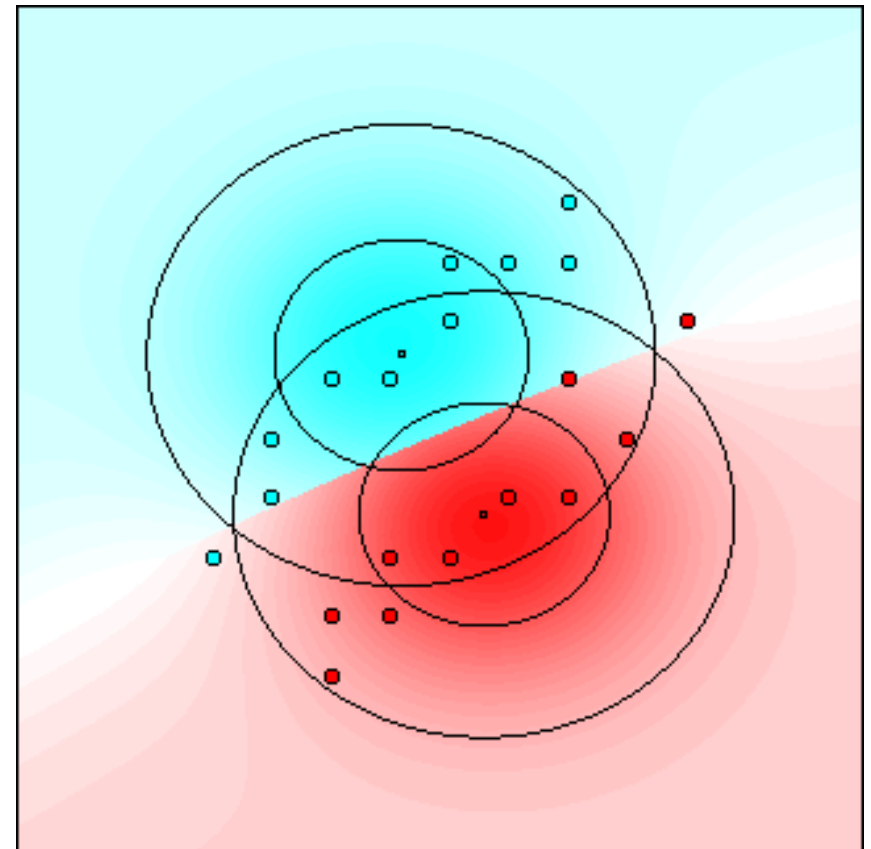
- 100 data points, 2 classes
- Small squares: mean values
- Inner ellipses: one standard deviation
- Outer ellipses: two standard deviations
- Classes overlap: classification is not perfect



Naive Bayes Classifier

# Naive Bayes Classifiers: Simple Example 3

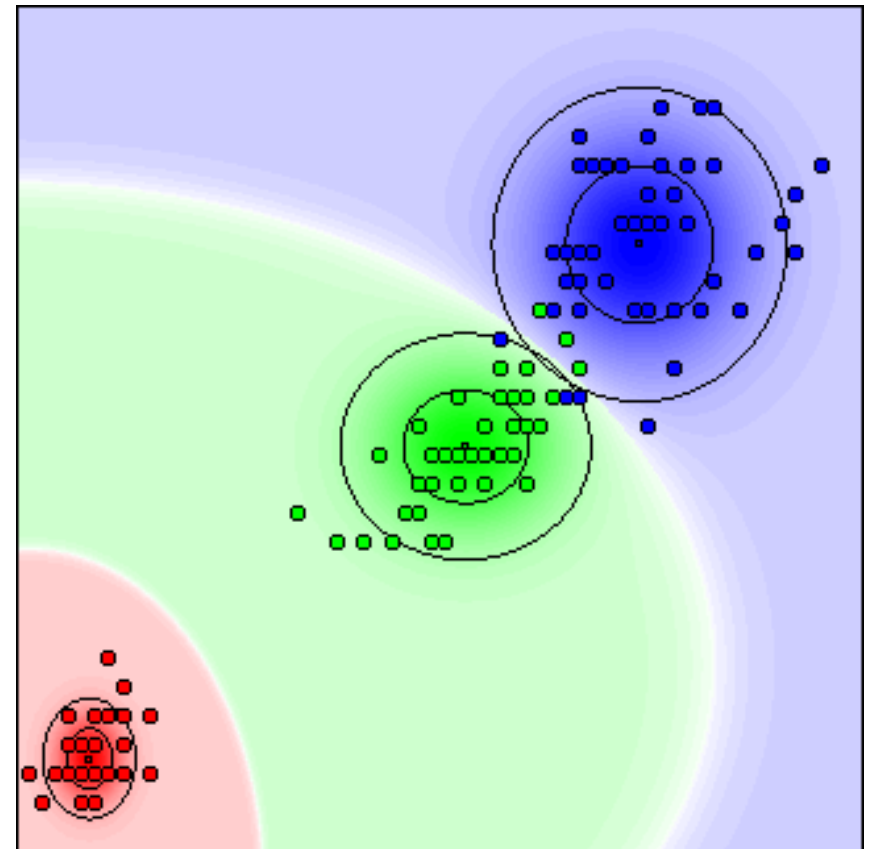
- 20 data points, 2 classes
- Small squares: mean values
- Inner ellipses: one standard deviation
- Outer ellipses: two standard deviations
- Attributes are not conditionally independent given the class.



**Naive Bayes Classifier**

# Naive Bayes Classifiers: Iris Data

- 150 data points, 3 classes
  - Iris setosa (red)
  - Iris versicolor (green)
  - Iris virginica (blue)
- Shown: 2 out of 4 attributes
  - sepal length
  - sepal width
  - petal length (horizontal)
  - petal width (vertical)
- 6 misclassifications on the training data (with all 4 attributes)



Naive Bayes Classifier

# Learning Graphical Models from Data: Learning the Structure

# Learning the Structure of Graphical Models from Data

- **Test whether a distribution is decomposable w.r.t. a given graph.**  
This is the most direct approach. It is not bound to a graphical representation, but can also be carried out w.r.t. other representations of the set of subspaces to be used to compute the (candidate) decomposition of the given distribution.
- **Find a suitable graph by measuring the strength of dependences.**  
This is a heuristic, but often highly successful approach, which is based on the frequently valid assumption that in a conditional independence graph an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to them.
- **Find an independence map by conditional independence tests.**  
This approach exploits the theorems that connect conditional independence graphs and graphs that represent decompositions. It has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs. However, wrong test results can thus have severe consequences.



# Evaluation Measures and Search Methods

- All learning algorithms for graphical models consist of an **evaluation measure** or **scoring function**, e.g.
  - Hartley information gain (relational networks)
  - Shannon information gain,  $\chi^2$ , K2 metric (probabilistic networks)and a (heuristic) **search method**, e.g.
  - conditional independence search
  - greedy search (spanning tree or K2 algorithm)
  - guided random search (simulated annealing, genetic algorithms)
- An exhaustive search over all graphs is too expensive:
  - $2^{\binom{n}{2}}$  possible undirected graphs for  $n$  attributes.
  - $f(n) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i)$  possible directed acyclic graphs.

# Learning the Structure of a Graphical Model: Testing for Decomposability

# Comparing Relations

- In order to evaluate a graph structure, we need a measure that compares the actual relation to the relation represented by the graph.
- For arbitrary  $R$ ,  $E_1$ , and  $E_2$  it is

$$R(E_1 \cap E_2) \leq \min\{R(E_1), R(E_2)\}.$$

- This relation entails that it is always

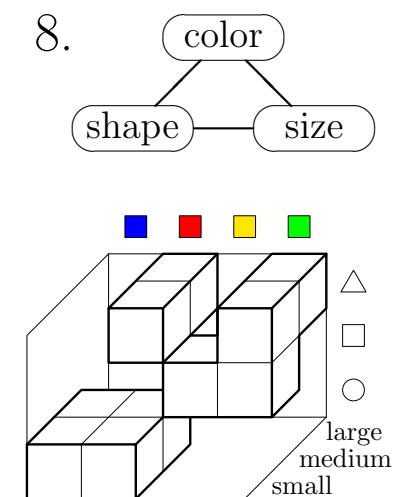
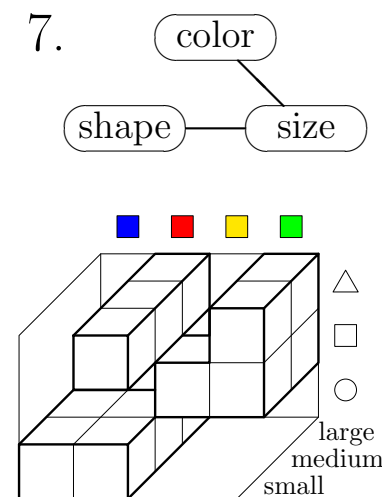
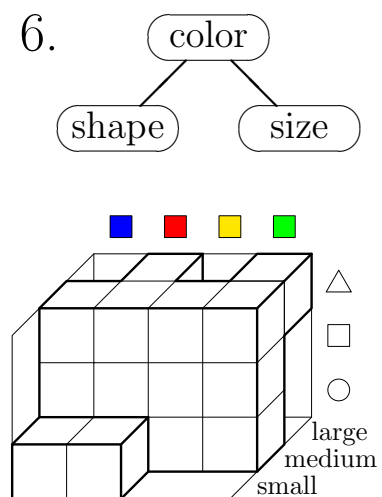
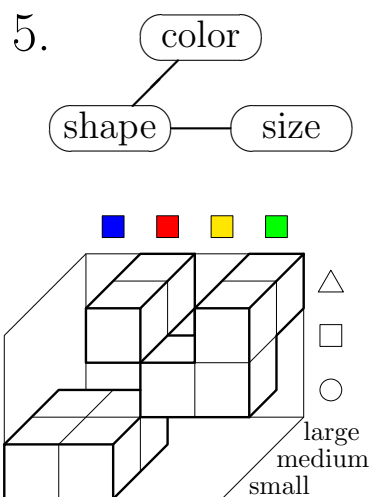
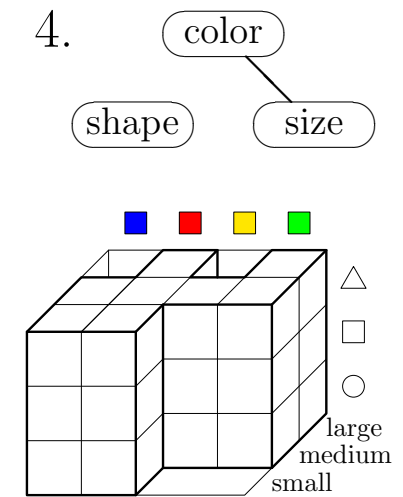
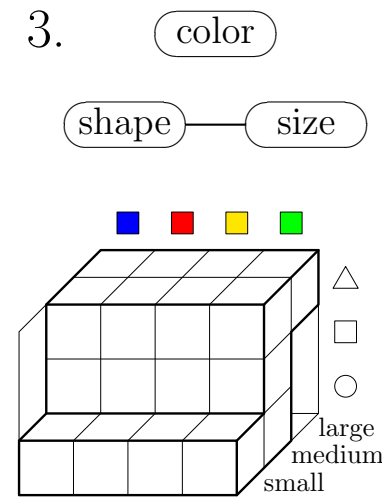
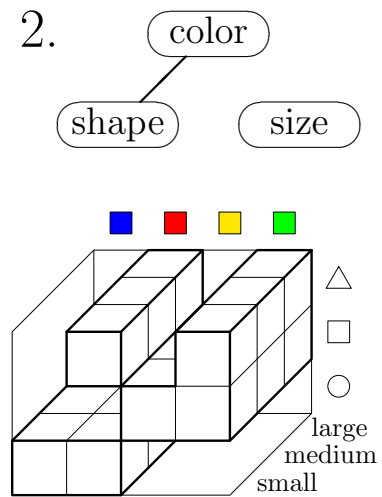
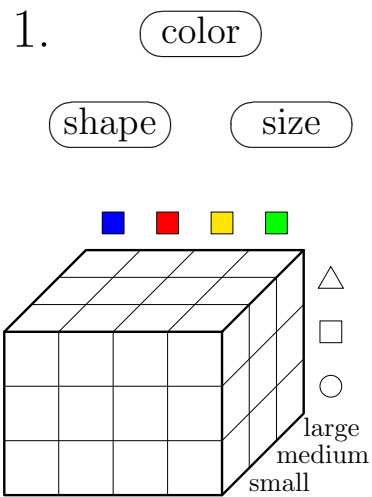
$$\forall a_1 \in \text{dom}(A_1) : \dots \forall a_n \in \text{dom}(A_n) : \\ r_U\left(\bigwedge_{A_i \in U} A_i = a_i\right) \leq \min_{M \in \mathcal{M}} \left\{ r_M\left(\bigwedge_{A_i \in M} A_i = a_i\right) \right\}.$$

- Therefore: Measure the quality of a family  $\mathcal{M}$  of subset of  $U$  as:

$$\sum_{a_1 \in \text{dom}(A_1)} \dots \sum_{a_n \in \text{dom}(A_n)} \left( \min_{M \in \mathcal{M}} \left\{ r_M\left(\bigwedge_{A_i \in M} A_i = a_i\right) \right\} - r_U\left(\bigwedge_{A_i \in U} A_i = a_i\right) \right)$$

Intuitively: Count the number of additional tuples.

# Direct Test for Decomposability: Relational



# Comparing Probability Distributions

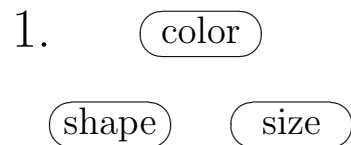
**Definition:** Let  $p_1$  and  $p_2$  be two strictly positive probability distributions on the same set  $\mathcal{E}$  of events. Then

$$I_{\text{KLdiv}}(p_1, p_2) = \sum_{E \in \mathcal{E}} p_1(E) \log_2 \frac{p_1(E)}{p_2(E)}$$

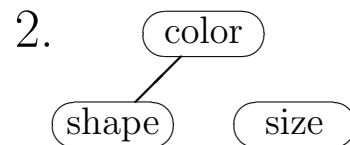
is called the **Kullback-Leibler information divergence** of  $p_1$  and  $p_2$ .

- The Kullback-Leibler information divergence is non-negative.
- It is zero if and only if  $p_1 \equiv p_2$ .
- Therefore it is plausible that this measure can be used to assess the quality of the approximation of a given multi-dimensional distribution  $p_1$  by the distribution  $p_2$  that is represented by a given graph:  
The smaller the value of this measure, the better the approximation.

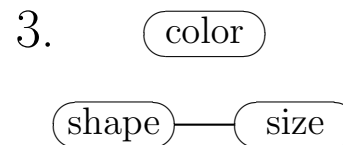
# Direct Test for Decomposability: Probabilistic



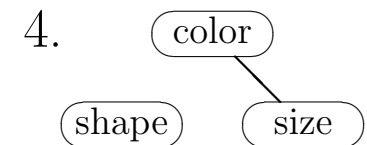
0.640  
-5041



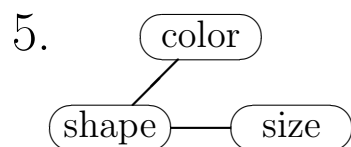
0.211  
-4612



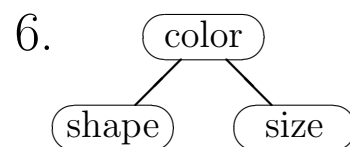
0.429  
-4830



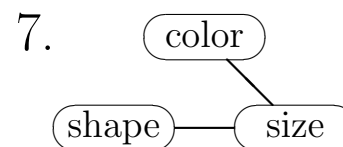
0.590  
-4991



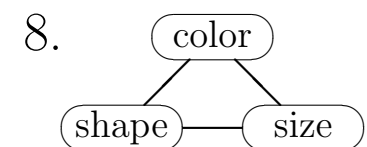
0  
-4401



0.161  
-4563



0.379  
-4780



0  
-4401

Upper numbers: The Kullback-Leibler information divergence of the original distribution and its approximation.

Lower numbers: The binary logarithms of the probability of an example database (log-likelihood of data).

# Learning the Structure of a Graphical Model: Strength of Marginal Dependences

# Strength of Marginal Dependences: Relational

- Learning a relational network consists in finding those subspace, for which the intersection of the cylindrical extensions of the projections to these subspaces approximates best the set of possible world states, i.e. contains as few additional tuples as possible.
- Since computing explicitly the intersection of the cylindrical extensions of the projections and comparing it to the original relation is too expensive, local evaluation functions are used, for instance:

subspace	color $\times$ shape	shape $\times$ size	size $\times$ color
possible combinations	12	9	12
occurring combinations	6	5	8
relative number	50%	56%	67%

- The relational network can be obtained by interpreting the relative numbers as edge weights and constructing the minimum weight spanning tree.



# Strength of Marginal Dependences: Relational

	■		■
		■	■
■	■		

Hartley information needed to determine

coordinates:  $\log_2 4 + \log_2 3 = \log_2 12 \approx 3.58$

coordinate pair:  $\log_2 6 \approx 2.58$

---

gain:  $\log_2 12 - \log_2 6 = \log_2 2 = 1$

**Definition:** Let  $A$  and  $B$  be two attributes and  $R$  a discrete possibility measure with  $\exists a \in \text{dom}(A) : \exists b \in \text{dom}(B) : R(A = a, B = b) = 1$ . Then

$$\begin{aligned} I_{\text{gain}}^{(\text{Hartley})}(A, B) &= \log_2 \left( \sum_{a \in \text{dom}(A)} R(A = a) \right) + \log_2 \left( \sum_{b \in \text{dom}(B)} R(B = b) \right) \\ &\quad - \log_2 \left( \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} R(A = a, B = b) \right) \\ &= \log_2 \frac{\left( \sum_{a \in \text{dom}(A)} R(A = a) \right) \cdot \left( \sum_{b \in \text{dom}(B)} R(B = b) \right)}{\sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} R(A = a, B = b)}, \end{aligned}$$

is called the **Hartley information gain** of  $A$  and  $B$  w.r.t.  $R$ .

# Strength of Marginal Dependences: Simple Example

- **Intuitive interpretation of Hartley information gain:**

The binary logarithm measures the number of questions to find the obtaining value with a scheme like a binary search. Thus Hartley information gain measures the reduction in the number of necessary questions.

- Results for the simple example:

$$I_{\text{gain}}^{(\text{Hartley})}(\text{color, shape}) = 1.00 \text{ bit}$$

$$I_{\text{gain}}^{(\text{Hartley})}(\text{shape, size}) \approx 0.86 \text{ bit}$$

$$I_{\text{gain}}^{(\text{Hartley})}(\text{color, size}) \approx 0.58 \text{ bit}$$

- Applying the Kruskal algorithm yields as a learning result:



As we know, this graph describes indeed a decomposition of the relation.

# Strength of Marginal Dependences: Probabilistic

## Mutual Information / Cross Entropy / Information Gain

Based on Shannon Entropy  $H = - \sum_{i=1}^n p_i \log_2 p_i$  (Shannon 1948)

$$\begin{aligned} I_{\text{gain}}(A, B) &= \overbrace{H(A)} - \overbrace{H(A | B)} \\ &= - \sum_{i=1}^{n_A} p_i \log_2 p_i - \sum_{j=1}^{n_B} p_{.j} \left( - \sum_{i=1}^{n_A} p_{i|j} \log_2 p_{i|j} \right) \end{aligned}$$

$H(A)$

Entropy of the distribution on attribute  $A$

$H(A|B)$

*Expected entropy* of the distribution on attribute  $A$   
if the value of attribute  $B$  becomes known

$H(A) - H(A|B)$

Expected reduction in entropy or *information gain*

# Interpretation of Shannon Entropy

- Let  $S = \{s_1, \dots, s_n\}$  be a finite set of alternatives having positive probabilities  $P(s_i)$ ,  $i = 1, \dots, n$ , satisfying  $\sum_{i=1}^n P(s_i) = 1$ .

- **Shannon Entropy:**

$$H(S) = - \sum_{i=1}^n P(s_i) \log_2 P(s_i)$$

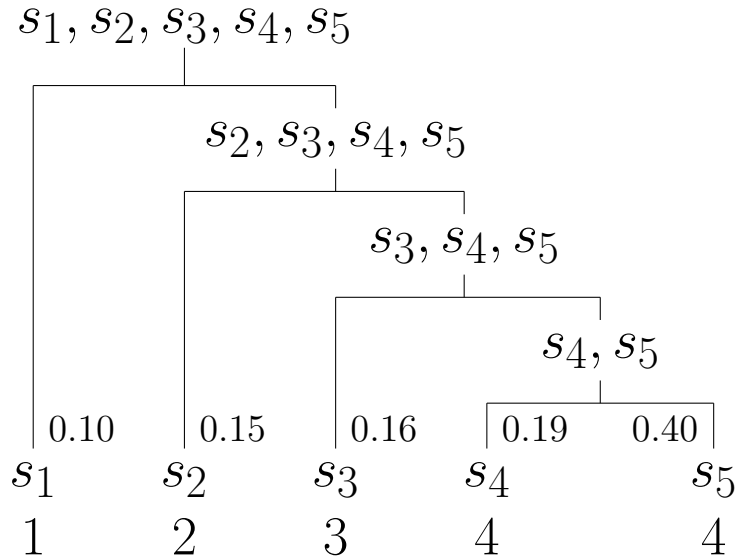
- Intuitively: **Expected number of yes/no questions that have to be asked in order to determine the obtaining alternative.**
  - Suppose there is an oracle, which knows the obtaining alternative, but responds only if the question can be answered with “yes” or “no”.
  - A better question scheme than asking for one alternative after the other can easily be found: Divide the set into two subsets of about equal size.
  - Ask for containment in an arbitrarily chosen subset.
  - Apply this scheme recursively  $\rightarrow$  number of questions bounded by  $\lceil \log_2 n \rceil$ .

# Question/Coding Schemes

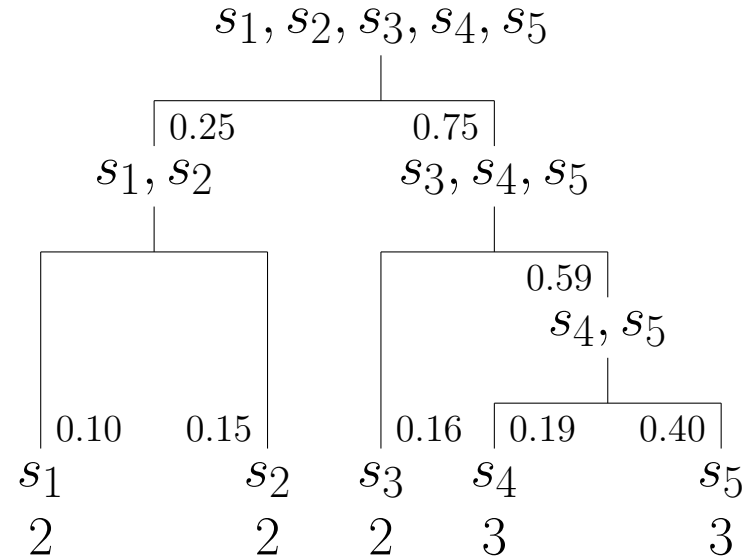
$$P(s_1) = 0.10, \quad P(s_2) = 0.15, \quad P(s_3) = 0.16, \quad P(s_4) = 0.19, \quad P(s_5) = 0.40$$

$$\text{Shannon entropy: } -\sum_i P(s_i) \log_2 P(s_i) = 2.15 \text{ bit/symbol}$$

## Linear Traversal



## Equal Size Subsets



# Question/Coding Schemes

- Splitting into subsets of about equal size can lead to a bad arrangement of the alternatives into subsets → high expected number of questions.
- Good question schemes take the probability of the alternatives into account.
- **Shannon-Fano Coding** (1948)
  - Build the question/coding scheme top-down.
  - Sort the alternatives w.r.t. their probabilities.
  - Split the set so that the subsets have about equal *probability* (splits must respect the probability order of the alternatives).
- **Huffman Coding** (1952)
  - Build the question/coding scheme bottom-up.
  - Start with one element sets.
  - Always combine those two sets that have the smallest probabilities.

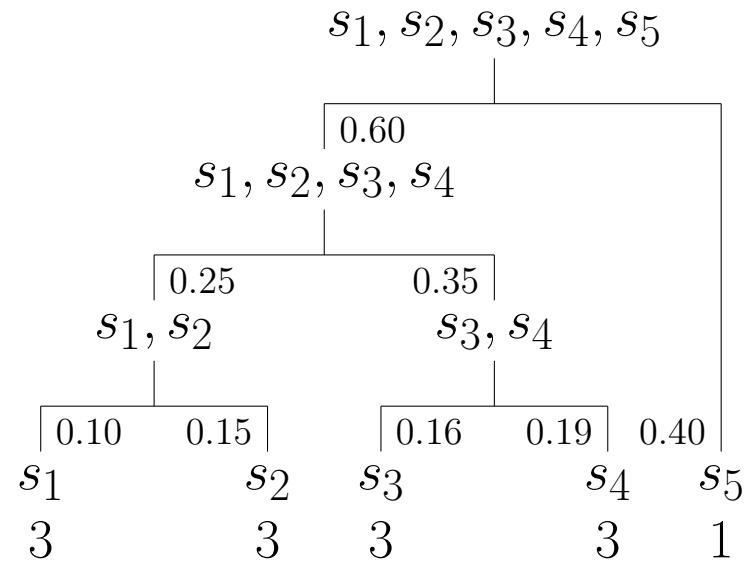
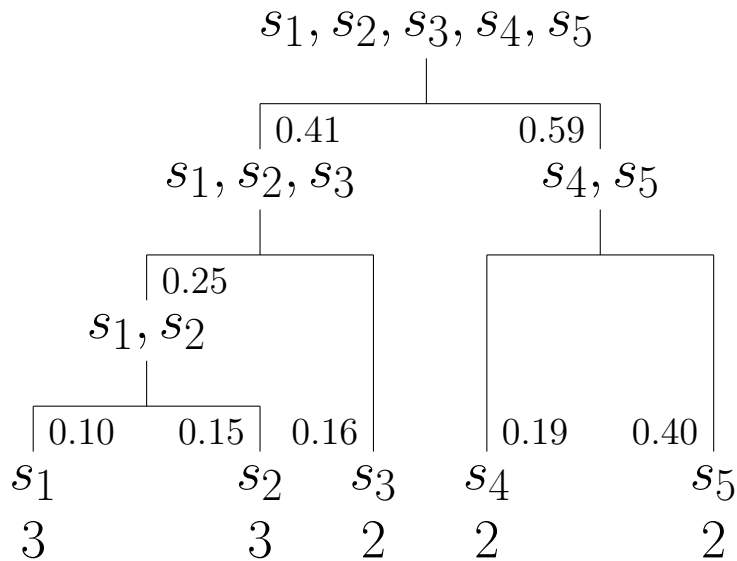
# Question/Coding Schemes

$$P(s_1) = 0.10, \quad P(s_2) = 0.15, \quad P(s_3) = 0.16, \quad P(s_4) = 0.19, \quad P(s_5) = 0.40$$

$$\text{Shannon entropy: } -\sum_i P(s_i) \log_2 P(s_i) = 2.15 \text{ bit/symbol}$$

## Shannon–Fano Coding (1948)

## Huffman Coding (1952)



Code length: 2.25 bit/symbol

Code efficiency: 0.955

Code length: 2.20 bit/symbol

Code efficiency: 0.977

# Question/Coding Schemes

- It can be shown that Huffman coding is optimal if we have to determine the obtaining alternative in a single instance.  
(No question/coding scheme has a smaller expected number of questions.)
- Only if the obtaining alternative has to be determined in a sequence of (independent) situations, this scheme can be improved upon.
- Idea: Process the sequence not instance by instance, but combine two, three or more consecutive instances and ask directly for the obtaining combination of alternatives.
- Although this enlarges the question/coding scheme, the expected number of questions per identification is reduced (because each interrogation identifies the obtaining alternative for several situations).
- However, the expected number of questions per identification cannot be made arbitrarily small. Shannon showed that there is a lower bound, namely the Shannon entropy.



# Interpretation of Shannon Entropy

$$P(s_1) = \frac{1}{2}, \quad P(s_2) = \frac{1}{4}, \quad P(s_3) = \frac{1}{8}, \quad P(s_4) = \frac{1}{16}, \quad P(s_5) = \frac{1}{16}$$

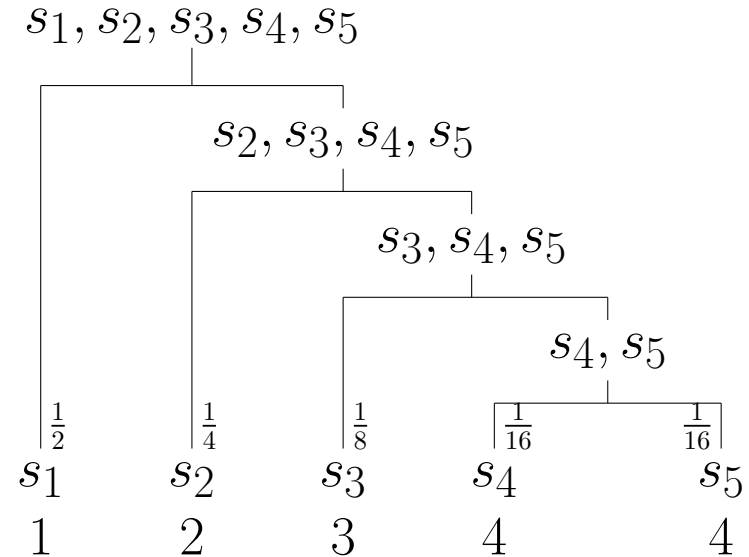
Shannon entropy:  $-\sum_i P(s_i) \log_2 P(s_i) = 1.875$  bit/symbol

If the probability distribution allows for a perfect Huffman code (code efficiency 1), the Shannon entropy can easily be interpreted as follows:

$$\begin{aligned} & -\sum_i P(s_i) \log_2 P(s_i) \\ &= \sum_i \underbrace{P(s_i)}_{\substack{\text{occurrence} \\ \text{probability}}} \cdot \underbrace{\log_2 \frac{1}{P(s_i)}}_{\substack{\text{path length} \\ \text{in tree}}} . \end{aligned}$$

In other words, it is the expected number of needed yes/no questions.








## Perfect Question Scheme










Code length: 1.875 bit/symbol  
Code efficiency: 1

# Information Gain: Simple Example

projection to  
subspace

				
	40	180	20	160
	12	6	120	102
	168	144	30	18




product of  
marginals

				
	88	132	68	112
	53	79	41	67
	79	119	61	101




information  
gain

0.429 bit

s m l

	20	180	200
	40	160	40
	180	120	60

s m l

	96	184	120
	58	110	72
	86	166	108

0.211 bit

large	50	115	35	100
medium	82	133	99	146
small	88	82	36	34

large	66	99	51	84
medium	101	152	78	129
small	53	79	41	67

0.050 bit

# Strength of Marginal Dependences: Simple Example

- Results for the simple example:

$$I_{\text{gain}}(\text{color}, \text{shape}) = 0.429 \text{ bit}$$

$$I_{\text{gain}}(\text{shape}, \text{size}) = 0.211 \text{ bit}$$

$$I_{\text{gain}}(\text{color}, \text{size}) = 0.050 \text{ bit}$$

- Applying the Kruskal algorithm yields as a learning result:



- It can be shown that this approach always yields the best possible spanning tree w.r.t. Kullback-Leibler information divergence (Chow and Liu 1968).
- In an extended form this also holds for certain classes of graphs (for example, tree-augmented naive Bayes classifiers).
- For more complex graphs, the best graph need not be found (there are counterexamples, see below).

# Strength of Marginal Dependences: General Algorithms

- **Optimum Weight Spanning Tree Construction**
  - Compute an evaluation measure on all possible edges (two-dimensional subspaces).
  - Use the Kruskal algorithm to determine an optimum weight spanning tree.
- **Greedy Parent Selection** (for directed graphs)
  - Define a topological order of the attributes (to restrict the search space).
  - Compute an evaluation measure on all single attribute hyperedges.
  - For each preceding attribute (w.r.t. the topological order):  
add it as a candidate parent to the hyperedge and compute the evaluation measure again.
  - Greedily select a parent according to the evaluation measure.
  - Repeat the previous two steps until no improvement results from them.

# Another Probabilistic Evaluation Measure: K2 Metric

- Idea: Compute the probability of a graph given the data (Bayesian approach)

$$P(\vec{G} \mid D) = \frac{1}{P(D)} \int_{\Theta} P(D \mid \vec{G}, \Theta) f(\Theta \mid \vec{G}) P(\vec{G}) d\Theta$$

$\vec{G}$  directed acyclic graph underlying the graphical model

$\Theta$  probability parameters of the graphical model

$D$  database to learn from

- In order to compare two graphs, it is sufficient to compute the **Bayes factor**

$$\frac{P(\vec{G}_1 \mid D)}{P(\vec{G}_2 \mid D)} = \frac{P(\vec{G}_1, D)}{P(\vec{G}_2, D)} = \frac{\int_{\Theta_1} P(D \mid \vec{G}_1, \Theta_1) f(\Theta_1 \mid \vec{G}_1) P(\vec{G}_1) d\Theta_1}{\int_{\Theta_2} P(D \mid \vec{G}_2, \Theta_2) f(\Theta_2 \mid \vec{G}_2) P(\vec{G}_2) d\Theta_2}.$$

In this way one can avoid computing the probability  $P(D)$ .

Assuming equal probability of all graphs simplifies further.

# Another Probabilistic Evaluation Measure: K2 Metric

- Assumptions about data and parameter independence yield:

$$P(\vec{G}, D) = \gamma \prod_{k=1}^r \prod_{j=1}^{m_k} \int \cdots \int_{\theta_{ijk}} \left( \prod_{i=1}^{n_k} \theta_{ijk}^{N_{ijk}} \right) f(\theta_{1jk}, \dots, \theta_{n_kjk}) d\theta_{1jk} \cdots d\theta_{n_kjk}$$

$r$  number of attributes describing the domain under consideration

$n_k$  number of values of the  $k$ -th attribute  $A_k$ , i.e.,  $n_k = |\text{dom}(A_k)|$

$m_k$  number of instantiations of the parents of the  $k$ -th attribute in  $\vec{G}$ , i.e.,  $m_k = \prod_{A_j \in \text{parents}(A_k)} n_j = \prod_{A_j \in \text{parents}(A_k)} |\text{dom}(A_j)|$

$\theta_{ijk}$  probability that the  $k$ -th attribute takes its  $i$ -th value and its parents in  $\vec{G}$  take their  $j$ -th instantiation

$N_{ijk}$  number of sample cases in which the  $k$ -th attribute has its  $i$ -th value and its parents in  $\vec{G}$  have their  $j$ -th instantiation

$\gamma$  normalization factor

# Another Probabilistic Evaluation Measure: K2 Metric

- Choose  $f(\theta_{1jk}, \dots, \theta_{n_kjk}) = \text{const.}$  [Cooper and Herskovits 1992]
- Then the solution can be obtained via Dirichlet's integral:

$$K_2(\vec{G}, D) = \gamma \prod_{k=1}^r \prod_{j=1}^{m_k} \frac{(n_k - 1)!}{(N_{.jk} + n_k - 1)!} \prod_{i=1}^{n_k} N_{ijk}!$$

- Since this formula is a product over the attributes, each attribute can be handled more or less separately.
- Core ideas of the K2 algorithm:
  - Fix a topological order of the attributes.  
(Reduces the search space and ensures that the graph is acyclic.)
  - Select the parents of each attribute greedily based on the K2 metric (or rather its corresponding factor).

# A Generalization of the K2 Metric

- Choose a maximum likelihood estimation of the probability parameters:

$$f(\theta_{1jk}, \dots, \theta_{n_kjk}) = \prod_{i=1}^{n_k} \delta \left( \theta_{ijk} - \frac{N_{ijk}}{N_{.jk}} \right)$$

$$\Rightarrow g_{\infty}(\vec{G}, D) = \gamma \prod_{k=1}^r \prod_{j=1}^{m_k} \prod_{i=1}^{n_k} \left( \frac{N_{ijk}}{N_{.jk}} \right)^{N_{ijk}} \quad (\text{equivalent to information gain})$$

- Choose the likelihood function scaled to maximum 1 and raised to the power  $\alpha$ :

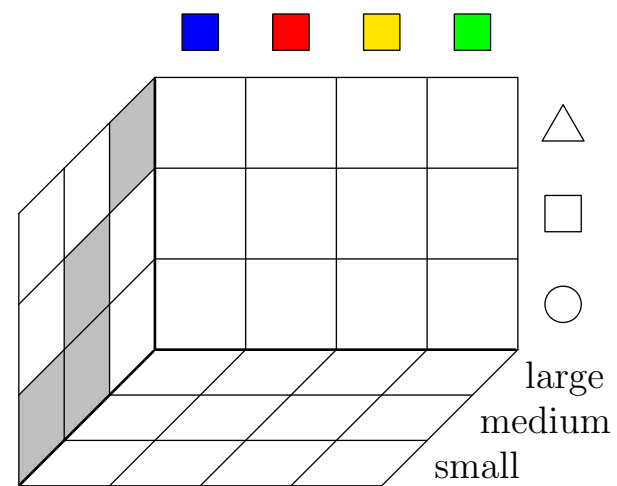
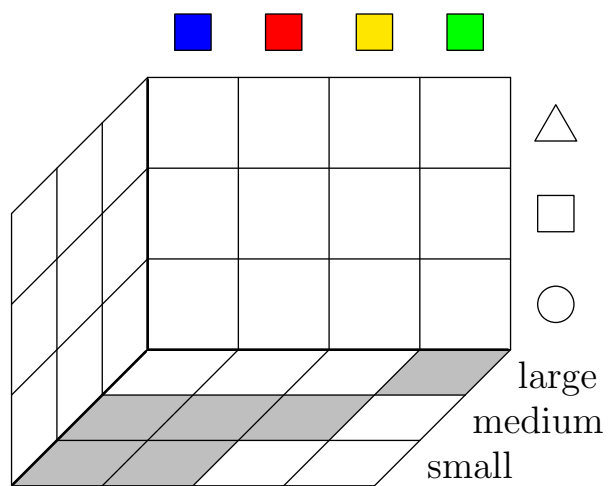
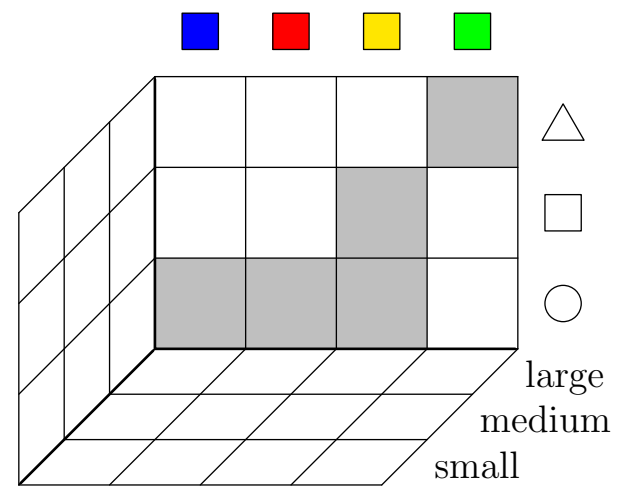
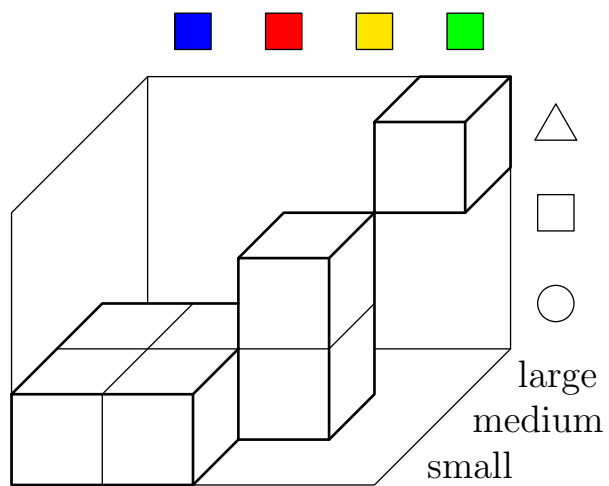
$$f_{\alpha}(\theta_{1jk}, \dots, \theta_{n_kjk}) = \beta \cdot \prod_{i=1}^{n_k} \theta_{ijk}^{\alpha N_{ijk}}$$

$$\Rightarrow g_{\alpha}(\vec{G}, D) = \gamma \prod_{k=1}^r \prod_{j=1}^{m_k} \frac{\Gamma(\alpha N_{.jk} + n_k)}{\Gamma((\alpha + 1)N_{.jk} + n_k)} \cdot \prod_{i=1}^{n_k} \frac{\Gamma((\alpha + 1)N_{ijk} + 1)}{\Gamma(\alpha N_{ijk} + 1)}$$

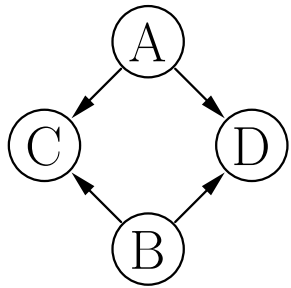
- The parameter  $\alpha$  can be interpreted as a sensitivity parameter, which determines the strength of the tendency to select parent attributes.



# Strength of Marginal Dependences: Drawbacks



# Strength of Marginal Dependences: Drawbacks



$p_A$	$a_1$	$a_2$
	0.5	0.5

$p_B$	$b_1$	$b_2$
	0.5	0.5

$p_{C AB}$	$a_1b_1$	$a_1b_2$	$a_2b_1$	$a_2b_2$
$c_1$	0.9	0.3	0.3	0.5
$c_2$	0.1	0.7	0.7	0.5

$p_{D AB}$	$a_1b_1$	$a_1b_2$	$a_2b_1$	$a_2b_2$
$d_1$	0.9	0.3	0.3	0.5
$d_2$	0.1	0.7	0.7	0.5

$p_{AD}$	$a_1$	$a_2$
$d_1$	0.3	0.2
$d_2$	0.2	0.3

$p_{BD}$	$b_1$	$b_2$
$d_1$	0.3	0.2
$d_2$	0.2	0.3

$p_{CD}$	$c_1$	$c_2$
$d_1$	0.31	0.19
$d_2$	0.19	0.31

- Greedy parent selection can lead to suboptimal results if there is more than one path connecting two attributes.
- Here: the edge  $C \rightarrow D$  is selected first.

# Learning the Structure of a Graphical Model: Conditional Independence Tests

# Structure Learning with Conditional Independence Tests

**General Idea:** Exploit the theorems that connect conditional independence graphs and graphs that represent decompositions.

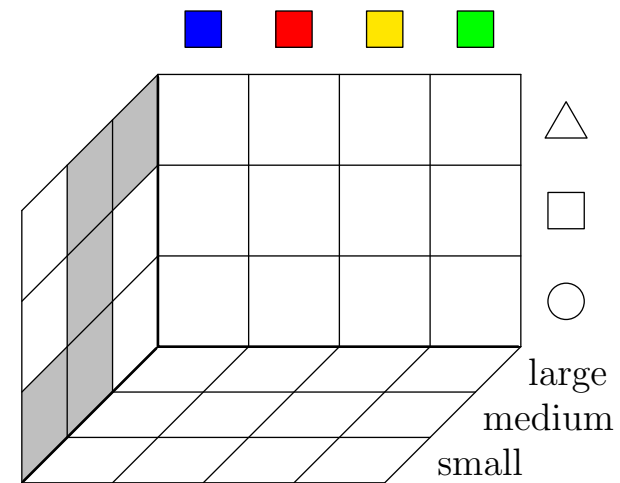
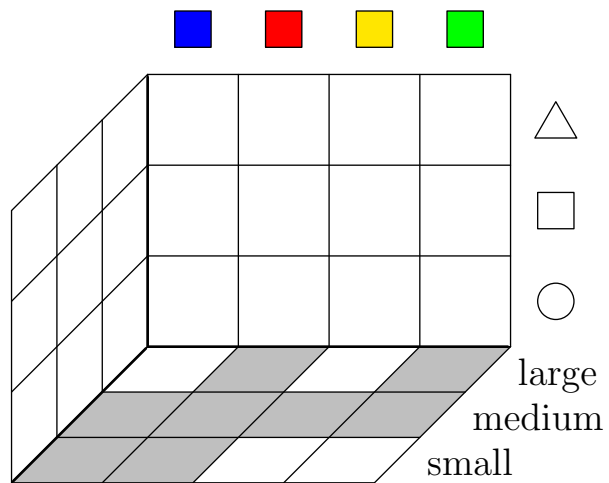
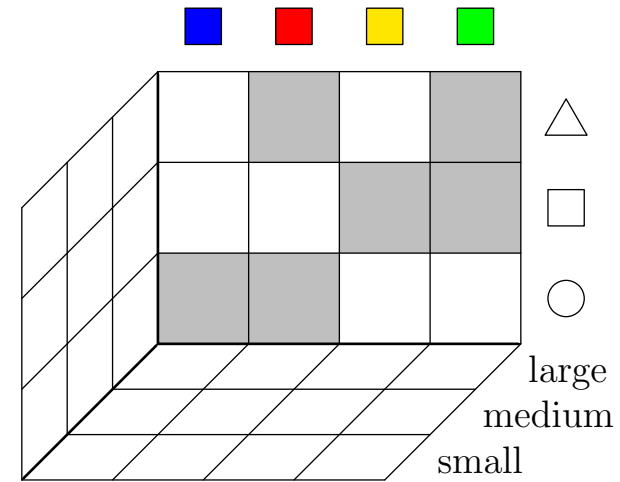
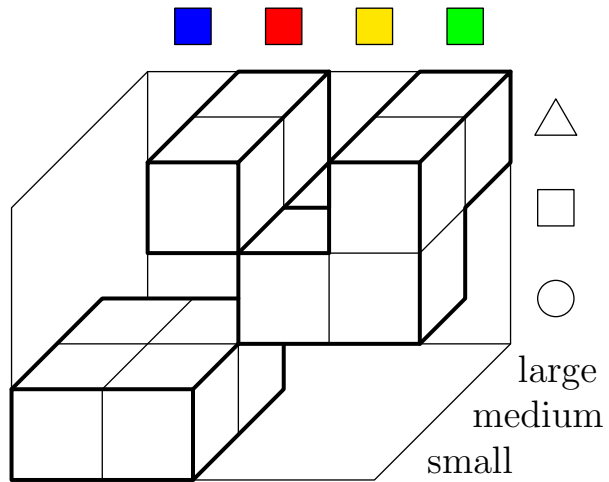
In other words: we want a graph describing a decomposition,  
but we search for a conditional independence graph.

This approach has the advantage that a single conditional independence test, if it fails, can exclude several candidate graphs.

## Assumptions:

- *Faithfulness:* The domain under consideration can be accurately described with a graphical model (more precisely: there exists a perfect map).
- *Reliability of Tests:* The result of all conditional independence tests coincides with the actual situation in the underlying distribution.
- Other assumptions that are specific to individual algorithms.

# Conditional Independence Tests: Relational



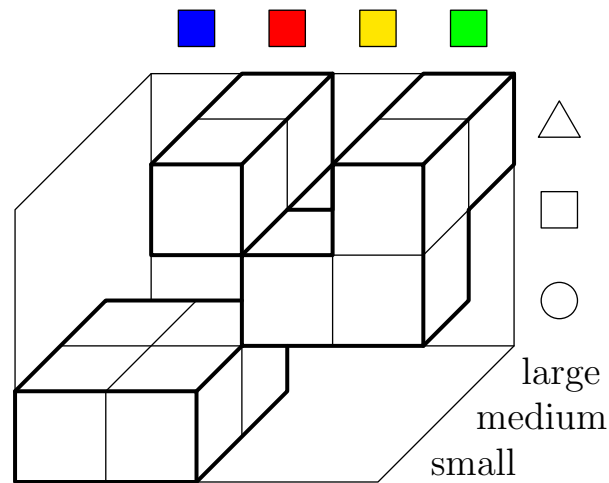
# Conditional Independence Tests: Relational

- The Hartley information gain can be used directly to test for (approximate) **marginal independence**.

attributes	relative number of possible value combinations	Hartley information gain
color, shape	$\frac{6}{3 \cdot 4} = \frac{1}{2} = 50\%$	$\log_2 3 + \log_2 4 - \log_2 6 = 1$
color, size	$\frac{8}{3 \cdot 4} = \frac{2}{3} \approx 67\%$	$\log_2 3 + \log_2 4 - \log_2 8 \approx 0.58$
shape, size	$\frac{5}{3 \cdot 3} = \frac{5}{9} \approx 56\%$	$\log_2 3 + \log_2 3 - \log_2 5 \approx 0.85$

- In order to test for (approximate) **conditional independence**:
  - Compute the Hartley information gain for each possible instantiation of the conditioning attributes.
  - Aggregate the result over all possible instantiations, for instance, by simply averaging them.

# Conditional Independence Tests: Simple Example



color	Hartley information gain
<span style="color: blue;">■</span>	$\log_2 1 + \log_2 2 - \log_2 2 = 0$
<span style="color: red;">■</span>	$\log_2 2 + \log_2 3 - \log_2 4 \approx 0.58$
<span style="color: yellow;">■</span>	$\log_2 1 + \log_2 1 - \log_2 1 = 0$
<span style="color: green;">■</span>	$\log_2 2 + \log_2 2 - \log_2 2 = 1$
	average: $\approx 0.40$

shape	Hartley information gain
△	$\log_2 2 + \log_2 2 - \log_2 4 = 0$
□	$\log_2 2 + \log_2 1 - \log_2 2 = 0$
○	$\log_2 2 + \log_2 2 - \log_2 4 = 0$
	average: $= 0$

size	Hartley information gain
large	$\log_2 2 + \log_2 1 - \log_2 2 = 0$
medium	$\log_2 4 + \log_2 3 - \log_2 6 = 1$
small	$\log_2 2 + \log_2 1 - \log_2 2 = 0$
	average: $\approx 0.33$

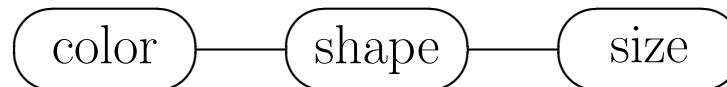
# Conditional Independence Tests: Simple Example

- The Shannon information gain can be used directly to test for (approximate) **marginal independence**.
- Conditional independence tests may be carried out by summing the information gain for all instantiations of the conditioning variables:

$$I_{\text{gain}}(A, B \mid C) = \sum_{c \in \text{dom}(C)} P(c) \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} P(a, b \mid c) \log_2 \frac{P(a, b \mid c)}{P(a \mid c) P(b \mid c)},$$

where  $P(c)$  is an abbreviation of  $P(C = c)$  etc.

- Since  $I_{\text{gain}}(\text{color}, \text{size} \mid \text{shape}) = 0$  indicates the only conditional independence, we get the following learning result:





# Conditional Independence Tests: General Algorithm

**Algorithm:** (conditional independence graph construction)

1. For each pair of attributes  $A$  and  $B$ , search for a set  $S_{AB} \subseteq U \setminus \{A, B\}$  such that  $A \perp\!\!\!\perp B \mid S_{AB}$  holds in  $\hat{P}$ , i.e.,  $A$  and  $B$  are independent in  $\hat{P}$  conditioned on  $S_{AB}$ . If there is no such  $S_{AB}$ , connect the attributes by an undirected edge.
2. For each pair of non-adjacent variables  $A$  and  $B$  with a common neighbour  $C$  (i.e.,  $C$  is adjacent to  $A$  as well as to  $B$ ), check whether  $C \in S_{AB}$ .
  - If it is, continue.
  - If it is not, add arrow heads pointing to  $C$ , i.e.,  $A \rightarrow C \leftarrow B$ .
3. Recursively direct all undirected edges according to the rules:
  - If for two adjacent variables  $A$  and  $B$  there is a strictly directed path from  $A$  to  $B$  not including  $A \rightarrow B$ , then direct the edge towards  $B$ .
  - If there are three variables  $A$ ,  $B$ , and  $C$  with  $A$  and  $B$  not adjacent,  $B - C$ , and  $A \rightarrow C$ , then direct the edge  $C \rightarrow B$ .

# Conditional Independence Tests: Simple Example

Suppose that the following conditional independence statements hold:

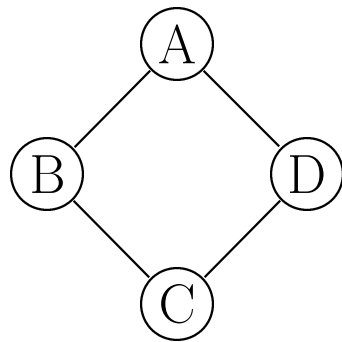
$$\begin{array}{ll} A \perp\!\!\!\perp_{\hat{P}} B \mid \emptyset & B \perp\!\!\!\perp_{\hat{P}} A \mid \emptyset \\ A \perp\!\!\!\perp_{\hat{P}} D \mid C & D \perp\!\!\!\perp_{\hat{P}} A \mid C \\ B \perp\!\!\!\perp_{\hat{P}} D \mid C & D \perp\!\!\!\perp_{\hat{P}} B \mid C \end{array}$$

All other possible conditional independence statements that can be formed with the attributes  $A$ ,  $B$ ,  $C$ , and  $D$  (with single attributes on the left) do not hold.

- **Step 1:** Since there is no set rendering  $A$  and  $C$ ,  $B$  and  $C$  and  $C$  and  $D$  independent, the edges  $A - C$ ,  $B - C$ , and  $C - D$  are inserted.
- **Step 2:** Since  $C$  is a common neighbor of  $A$  and  $B$  and we have  $A \perp\!\!\!\perp_{\hat{P}} B \mid \emptyset$ , but  $A \not\perp\!\!\!\perp_{\hat{P}} B \mid C$ , the first two edges must be directed  $A \rightarrow C \leftarrow B$ .
- **Step 3:** Since  $A$  and  $D$  are not adjacent,  $C - D$  and  $A \rightarrow C$ , the edge  $C - D$  must be directed  $C \rightarrow D$ .  
(Otherwise step 2 would have already fixed the orientation  $C \leftarrow D$ .)

# Conditional Independence Tests: Drawbacks

- The conditional independence graph construction algorithm presupposes that there is a **perfect map**. If there is no perfect map, the result may be invalid.



$p_{ABCD}$	$A = a_1$		$A = a_2$	
	$B = b_1$	$B = b_2$	$B = b_1$	$B = b_2$
$C = c_1$ $D = d_1$	$1/47$	$1/47$	$1/47$	$2/47$
$D = d_2$	$1/47$	$1/47$	$2/47$	$4/47$
$C = c_2$ $D = d_1$	$1/47$	$2/47$	$1/47$	$4/47$
$D = d_2$	$2/47$	$4/47$	$4/47$	$16/47$

- Independence tests of high order**, i.e., with a large number of conditions, may be necessary.
- There are approaches to mitigate these drawbacks.  
(For example, the order is restricted and all tests of higher order are assumed to fail, if all tests of lower order failed.)

# The Cheng–Bell–Liu Algorithm

- **Drafting:** Build a so-called Chow–Liu tree as an initial graphical model.
  - Evaluate all attribute pairs (candidate edges) with information gain.
  - Discard edges with evaluation below independence threshold ( $\sim 0.1$  bits).
  - Build optimum (maximum) weight spanning tree.
- **Thickening:** Add necessary edges.
  - Traverse remaining candidate edges in the order of decreasing evaluation.
  - Test for conditional independence in order to determine whether an edge is needed in the graphical model.
  - Use local Markov property to select a condition set: an attribute is conditionally independent of all non-descendants given its parents.
  - Since the graph is undirected in this step, the set of adjacent nodes is reduced iteratively and greedily in order to remove possible children.

# The Cheng–Bell–Liu Algorithm (continued)

- **Thinning:** Remove superfluous edges.
  - In the thickening phase a conditional independence test may have failed, because the graph was still too sparse.
  - Traverse all edges that have been added to the current graphical model and test for conditional independence.
  - Remove unnecessary edges.  
(two phases/approaches: heuristic test/strict test)
- **Orienting:** Direct the edges of the graphical model.
  - Identify the  $v$ -structures (converging directed edges).  
(Markov equivalence: same skeleton and same set of  $v$ -structures.)
  - Traverse all pairs of attributes with common neighbors and check which common neighbors are in the (maximally) reduced set of conditions.
  - Direct remaining edges by extending chains and avoiding cycles.

# Learning Undirected Graphical Models Directly

- **Drafting:** Build a Chow–Liu tree as an initial graphical model
  - Evaluate all attribute pairs (candidate edges) with specificity gain.
  - Discard edges with evaluation below independence threshold ( $\sim 0.015$ ).
  - Build optimum (maximum) weight spanning tree.
- **Thickening:** Add necessary edges.
  - Traverse remaining candidate edges in the order of decreasing evaluation.
  - Test for conditional independence in order to determine whether an edge is needed in the graphical model.
  - Use local Markov property to select a condition set: an attribute is conditionally independent of any non-neighbor given its neighbors.
  - Since the graphical model to be learned is undirected, *no (iterative) reduction of the condition set is needed* (decisive difference to Cheng–Bell–Liu Algorithm).

# Learning Undirected Graphical Models Directly

- **Moralizing:** Take care of possible  $v$ -structures.
  - If one assumes a perfect undirected map, this step is unnecessary. However,  $v$ -structures are too common and cannot be represented without loss in an undirected graphical model.
  - Possible  $v$ -structures can be taken care of by connecting the parents.
  - Traverse all edges with an evaluation below the independence threshold that have a common neighbor in the graph.
  - Add edge if conditional independence given the neighbors does not hold.
- **Thinning:** Remove superfluous edges.
  - In the thickening phase a conditional independence test may have failed, because the graph was still too sparse.
  - Traverse all edges that have been added to the current graphical model and test for conditional independence.

# Learning the Structure of a Graphical Model: Experiments and Applications



# Danish Jersey Cattle Blood Type Determination

network	edges	params.	train	test
indep.	0	59	-19921.2	-20087.2
orig.	22	219	-11391.0	-11506.1

## Optimum Weight Spanning Tree Construction

measure	edges	params.	train	test
$I_{\text{gain}}$	20.0	285.9	-12122.6	-12339.6
$\chi^2$	20.0	282.9	-12122.6	-12336.2

## Greedy Parent Selection w.r.t. a Topological Order

measure	edges	add.	miss.	params.	train	test
$I_{\text{gain}}$	35.0	17.1	4.1	1342.2	-11229.3	-11817.6
$\chi^2$	35.0	17.3	4.3	1300.8	-11234.9	-11805.2
K2	23.3	1.4	0.1	229.9	-11385.4	-11511.5
$L_{\text{red}}^{(\text{rel})}$	22.5	0.6	0.1	219.9	-11389.5	-11508.2

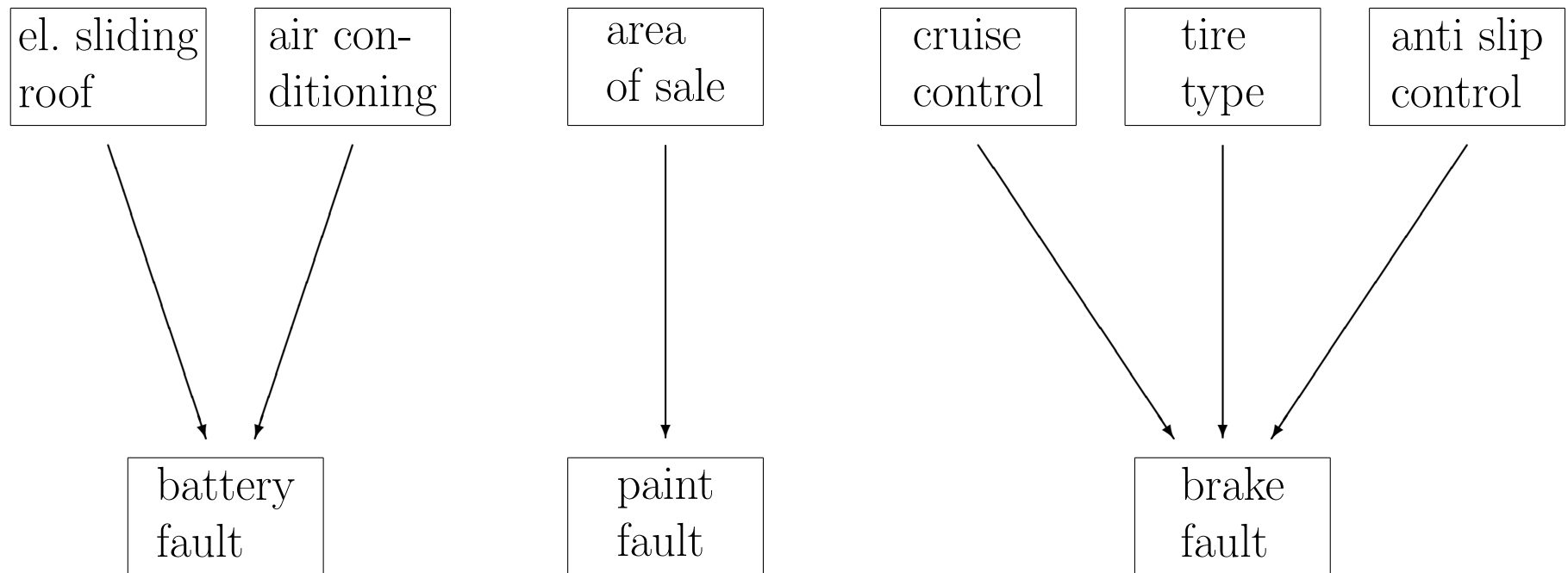
# Fields of Application (DaimlerChrysler AG)

- **Improvement of Product Quality by Finding Weaknesses**
  - Learn decision trees or inference network for vehicle properties and faults.
  - Look for unusual conditional fault frequencies.
  - Find causes for these unusual frequencies.
  - Improve construction of vehicle.
- **Improvement of Error Diagnosis in Garages**
  - Learn decision trees or inference network for vehicle properties and faults.
  - Record properties of new faulty vehicle.
  - Test for the most probable faults.

# A Simple Approach to Fault Analysis

- Check subnets consisting of an attribute and its parent attributes.
- Select subnets with highest deviation from independent distribution.

## Vehicle Properties



## Fault Data

# Example Subnet

## Influence of special equipment on battery faults:

(fictitious) frequency of battery faults	air conditioning		
	with	without	
electrical sliding roof	with	8 %	3 %
	without	3 %	2 %

- Significant deviation from independent distribution.
- Hints to possible causes and improvements.
- Here: Larger battery may be required, if an air conditioning system. *and* an electrical sliding roof are built in.

(The dependencies and frequencies of this example are fictitious, true numbers are confidential.)

# Summary

- **Decomposition:** Under certain conditions a distribution  $\delta$  (e.g. a probability distribution) on a multi-dimensional domain, which encodes *prior* or *generic knowledge* about this domain, can be decomposed into a set  $\{\delta_1, \dots, \delta_s\}$  of (overlapping) distributions on lower-dimensional subspaces.
- **Simplified Reasoning:** If such a decomposition is possible, it is sufficient to know the distributions on the subspaces to draw all inferences in the domain under consideration that can be drawn using the original distribution  $\delta$ .
- **Graphical Model:** The decomposition is represented by a graph (in the sense of graph theory). The edges of the graph indicate the paths along which evidence has to be propagated. Efficient and correct evidence propagation algorithms can be derived, which exploit the graph structure.
- **Learning from Data:** There are several highly successful approaches to learn graphical models from data, although all of them are based on heuristics. Exact learning methods are usually too costly.