Bayesian Networks

Prof. Dr. Rudolf Kruse,
Pascal Held

Computational Intelligence Group
Department of Knowledge Processing and Language Engineering
Faculty of Computer Science
`kruse@iws.cs.uni-magdeburg.de`

About me: Rudolf Kruse

In 1979 diploma in mathematics (minor computer science) at TU Braunschweig

There dissertation in 1980, habilitation in 1984

2 years full-time employee at Fraunhofer Institute

In 1986 offer of professorship for computer science at TU Braunschweig

Since 1996 professor at the University of Magdeburg

Research: data mining, explorative data analysis, fuzzy systems, neuronal networks, evolutionary algorithms, Bayesian networks

`mailto:kruse@iws.cs.uni-magdeburg.de`

Office: G29-008, telephone: 0391 67-58706

Consultation: Wednesdays, 10 a.m. – 11 noon

About the working group Computational Intelligence

Teaching:

Intelligent Systems	Bachelor (2 V + 2 Ü, 5 CP)
Neuronal Networks	Bachelor (2 V + 2 Ü, 5 CP)
Fuzzy Systems	Master (2 V + 2 Ü, 6 CP)
Bayesian Network	Master (2 V + 2 Ü, 6 CP)
Intelligent Data Analysis	Master (2 V + 2 Ü, 6 CP)

(pro-)seminars: Classification Algorithms, Clustering Algorithms

Research examples:

Validation of Density-based Clustering (C. Braune)
EEG Analysis with Deep Neural Networks (C. Doell)
Analysis of Social Networks (P. Held)

About the lecture

Lecture dates: Wednesday, 11:15 –12:45, G29-307

Information about the course:
`http://fuzzy.cs.ovgu.de/wiki/pmwiki.php?n=Lehre.BN1617`

- Weekly lecture slides as PDF
- Also assignment sheets for the exercise
- Online registration for exercises
- Important announcements and date!

Content of the lecture

About the exercise

Active participation and explanations of your solutions

Assistant will call attention to mistakes and answer questions

Pure 'calculations' of sample solution is not the purpose

Assistant:

- Pascal Held `pheld@ovgu.de`
- William Beluch `william.beluch@ovgu.de`

First assignment due October 22, 2015

- Monday, 9:15 – 10:45 (G29-K059), Beluch (english)
- Friday, 9:15 – 10:45 (G29-E037), Held (german)

Conditions for Certificate (“Schein”) and Exam

Exam or Certificate will get who...

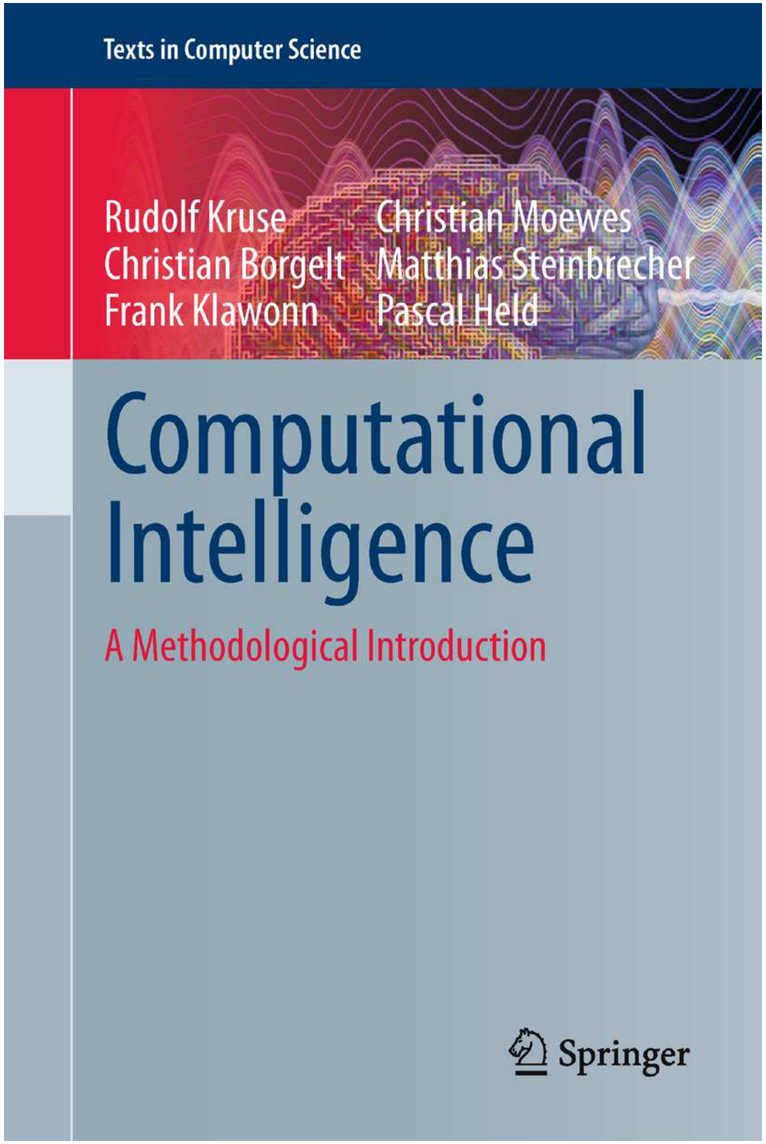
Contribute well in exercises every week,

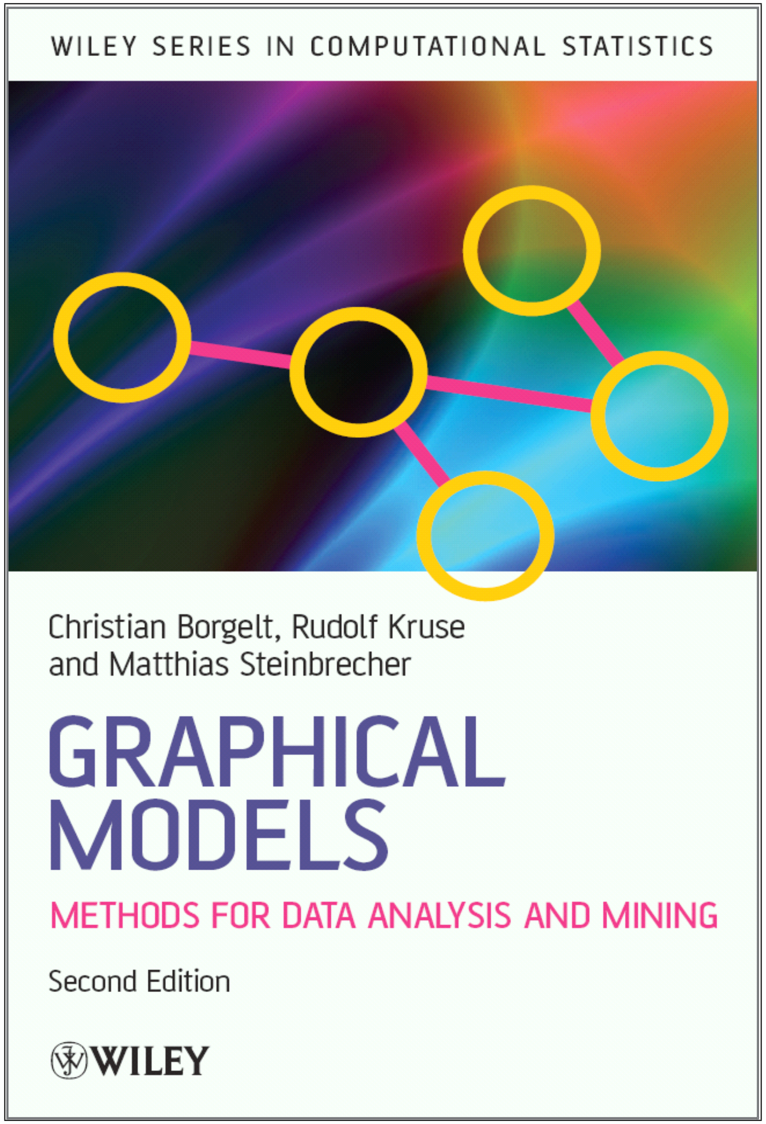
Present ≥ 2 solutions to written assignment during exercises.

Tick off $\geq 66\%$ of all written assignments,

Pass written exam (120 min)

Books about the course





http://www.computational-intelligence.eu/

Human Expert

A human *expert* is a specialist for a specific differentiated application field who creates solutions to customer problems in this respective field and supports them by applying these solutions.

Requirements

- Formulate precise problem scenarios from customer inquiries
- Find correct and complete solution
- Understandable answers
- Explanation of solution
- Support the deployment of solution

Knowledge Based Systems (2)

“Intelligent” System

An intelligent system is a program that models the knowledge and inference methods of a human expert of a specific field of application.

Requirements for construction:

- Knowledge Representation
- Knowledge Acquisition
- Knowledge Modification

Qualities of Knowledge

In most cases our knowledge about the present world is

incomplete/missing (knowledge is not comprehensive)

- e. g. "I don't know the bus departure times for public holidays because I only take the bus on working days."

vague/fuzzy/imprecise (knowledge is not exact)

- e. g. "The bus departs roughly every full hour."

uncertain (knowledge is unreliable)

- e. g. "The bus departs probably at 12 o'clock."

We have to decide nonetheless!

Reasoning under Vagueness

Reasoning with Probabilities

. . . and Cost/Benefit

Example

Objective: *Be at the university at 9:15 to attend a lecture.*

There are several plans to reach this goal:

- P_1: Get up at 8:00, leave at 8:55, take the bus at 9:00 ...
- P_2: Get up at 7:30, leave at 8:25, take the bus at 8:30 ...
- ...

All plans are *correct*, but

- they imply different *costs* and different *probabilities* to *actually* reach that goal.
- P_2 would be the plan of choice as the lecture is important and the success rate of P_1 is only about 80–95%.

Question: *Is a computer capable of solving these problems involving uncertainty?*

Uncertainty and Facts

Example:

We would like to support a robot's localization by fixed landmarks.
From the presence of a landmark we may infer the location.

Problem:

Sensors are imprecise!

- We cannot conclude definitely a location simply because there was a landmark detected by the sensors.
- The same holds true for undetected landmarks.
- Only probabilities are being increased or decreased.

Degrees of Belief

We (or other agents) are only believing facts or rules to some extent.

One possibility to express this *partial belief* is by using *probability theory*.

"The agent believes the sensor information to 0.9" means:
In 9 out of 10 cases the agent trusts in the correctness of the sensor output.

Probabilities gather the "uncertainty" that originates due to ignorance.

Probabilities $\neq$ Vagueness/Fuzziness!

- The predicate "large" is fuzzy whereas "This might be Peter's watch." is uncertain.

Rational Decisions under Uncertainty

Choice of several *actions* or *plans*

These may lead to different results with different *probabilities*.

The *actions* cause different (possibly subjective) *costs*.

The *results* yield different (possibly subjective) *benefits*.

It would be rational to choose that action that yields the largest total benefit.

Decision Theory = Utility Theory + Probability Theory

Decision-theoretic Agent

input perception
output action

1: $K \leftarrow$ a set of probabilistic beliefs about the state of the world
2: calculate updated probabilities for current state based on available evidence including current percept and previous action
3: calculate outcome probabilities for actions, given action descriptions and probabilities of current states
4: select action A with highest expected utility given probabilities of outcomes and utility information
5: **return** A

Decision Theory: An agent is rational if and only if it chooses the action yielding the largest utility averaged over all possible outcomes of all actions.

Rule-based Systems

Rule-based Systems

Modi of usage:

Query:	Facts are retrieved from database or user is interrogated
Explanation:	System answers questions how a decision was concluded

Example rule base:

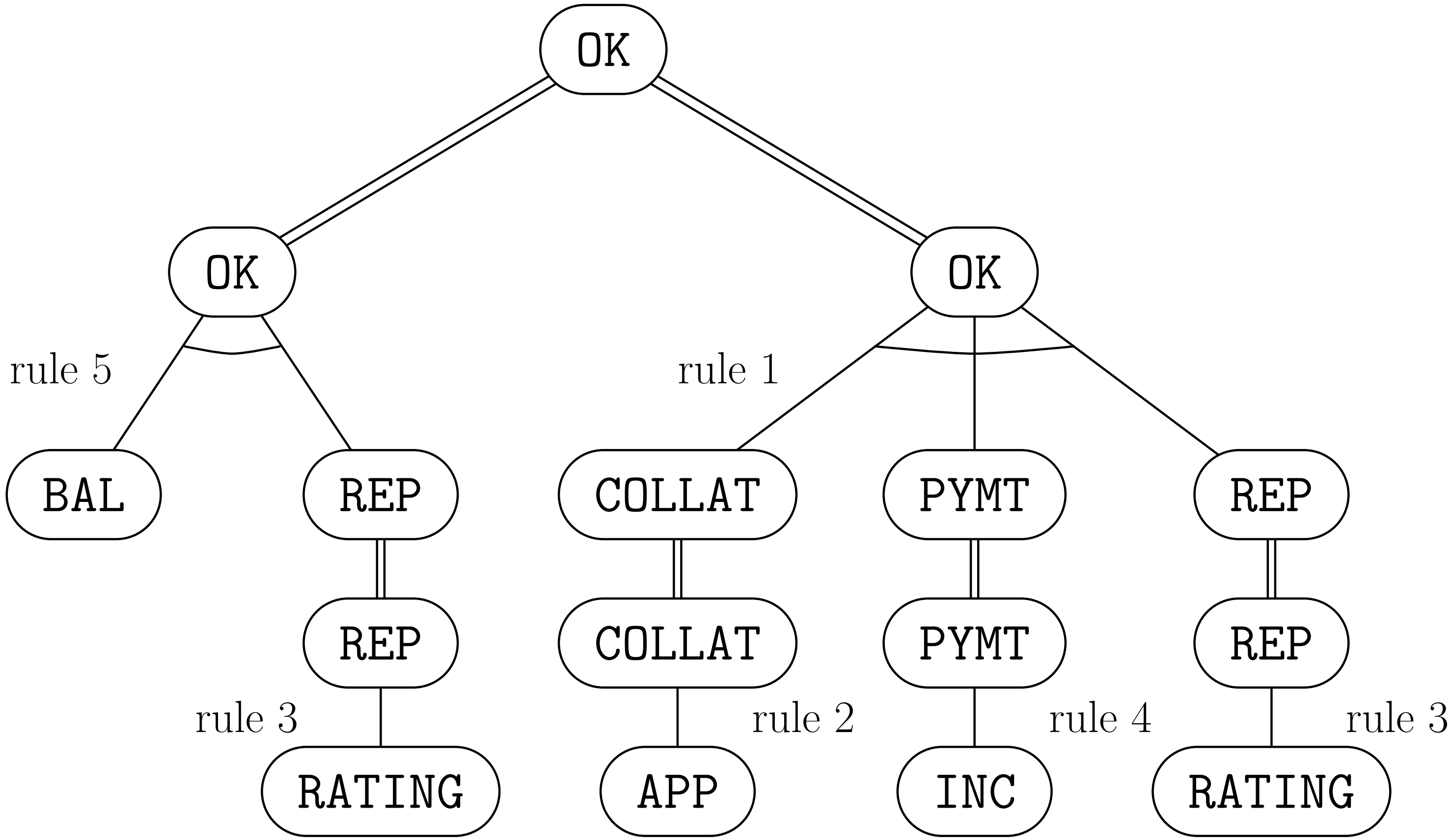
1. `COLLAT` $\wedge$ `PYMT` $\wedge$ `REP` $\Rightarrow$ `OK`
2. `APP` $\Rightarrow$ `COLLAT`
3. `RATING` $\Rightarrow$ `REP`
4. `INC` $\Rightarrow$ `PYMT`
5. `BAL` $\wedge$ `REP` $\Rightarrow$ `OK`

Description:

`COLLAT`	satisfactory collateral
`PYMT`	payments undisputed
`REP`	good reputation
`APP`	high appraisal
`RATING`	good rating
`INC`	positive income
`BAL`	excellent balance
`OK`	loan will be approved

Decision Rule Tree

Corresponding `and/or`-Tree:



Example Queries

User: Why do you believe that the loan should be approved?

System: Because the collateral for the loan is satisfactory, and the applicant is able to make the loan payments, and the applicant has a good financial reputation.

User: Why did you ask me about whether or not the applicant's income exceeds his/her expenses?

System: Because I was trying to establish whether or not the applicant is able to make the loan payments.

User: Why do you believe that the applicant has a good financial reputation?

System: Because the credit rating service said that the applicant has a good credit rating.

Rules

A rule in general is a *if-then*-construct consisting of a *condition* and an *action*.

$\boxed{\text{If } \textit{condition} \text{ then } \textit{conclusion}}$

These two parts may be interpreted differently according to the context:

- **Inference rules**: If *premise* then *conclusion*
- **Hypotheses:** If *evidence* then *hypothesis*
- **Productions:** If *condition* then *action*

Rules are often referred to as *productions* or *production rules*.

Rules

A rule in the ideal case represents a unit of knowledge.

A set of rules together with an execution/evaluation strategy comprises a program to find solutions to specific problem classes.

Prolog program: rule-based system

Rule-based systems are historically the first types of AI systems and were for a long time considered prototypical expert systems.

Nowadays, not every expert systems uses rules as its core inference mechanism.

Rising importance in the field of business process rules.

Rule Evaluation

Forward chaining

Expansion of knowledge base: as soon as new facts are inserted the system also calculates the conclusions/consequences.

Data-driven behavior

Premises-oriented reasoning: the chaining is determined by the left parts of the rules.

Backward chaining

Answering queries

Demand-driven behavior

Conclusion-oriented reasoning: the chaining is determined by the right parts of the rules.

Components of a Rules-based System

Data base

- Set of structured data objects
- Current state of modeled part of world

Rule base

- Set of rules
- Application of a rule will alter the data base

Rule interpreter

- Inference machine
- Controls the program flow of the system

Rule Interpretation

Main scheme forward chaining

- Select and apply rules from the set of rules with valid antecedences. This will lead to a modified data base and the possibility to apply further rules.

Run this cycle as long as possible.

The process terminates, if

- there is no rule left with valid antecendence
- a solution criterion is satisfied
- a stop criterion is satisfied (e. g. maximum number of steps)

Following tasks have to be solved:

- Identify those rules with a valid condition
  $\Rightarrow$ **Instantiation** or **Matching**
- Select rules to be executed
  $\Rightarrow$ need for **conflict resolution**
  (e. g. via partial or total orderings on the rules)

Certainty Factors

Mycin (1970)

Objective: Development of a system that supports physicians in diagnosing bacterial infections and suggesting antibiotics.

Features: Uncertain knowledge was represented and processed via *uncertainty factors*.

Knowledge: 500 (uncertain) decision rules as static knowledge base.

Case-specific knowledge:

- static: patients' data
- dynamic: intermediate results (facts)

Strengths:

- diagnosis-oriented interrogation
- hypotheses generation
- finding notification
- therapy recommendation
- explanation of inference path

Uncertainty Factors

Uncertainty factor $\quad \mathrm{CF} \in [-1, 1] \quad \approx \quad$ degree of belief.

Rules:

$$\mathrm{CF}(A \to B) \begin{cases} = 1 & B \text{ is certainly true given } A \\ > 0 & A \text{ supports } B \\ = 0 & A \text{ has no influence on } B \\ < 0 & A \text{ provides evidence against } B \\ = -1 & B \text{ is certainly false given } A \end{cases}$$

A Mycin Rule

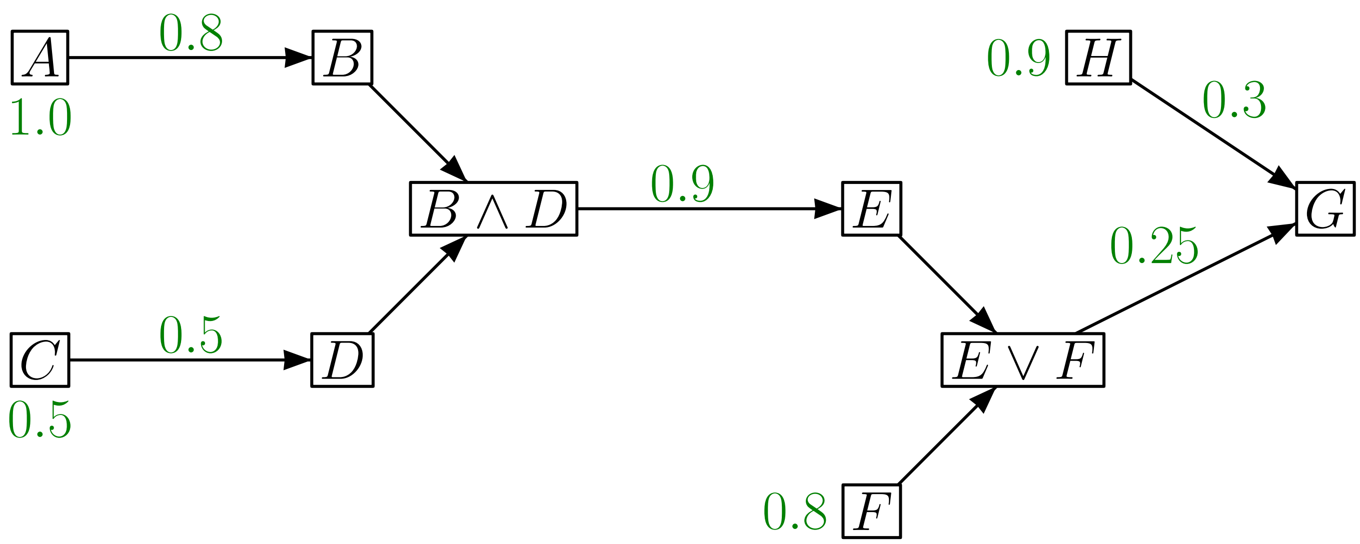
```
RULE035

PREMISE:   ($AND    (SAME CNTXT GRAM GRAMNEG)
                    (SAME CNTXT MORPH ROD)
                    (SAME CNTXT AIR ANAEROBIC))
ACTION:    (CONCL.CNTXT IDENTITY BACTEROIDES TALLY .6)
```

If	1) the *gram stain* of the organism is *gramneg*, and
	2) the *morphology* of the organism is *rod*, and
	3) the *aerobicity* of the organism is *anaerobic*
then	there is suggestive evidence (0.6) that the *identity* of the organism is *bacteroides*

Example

$$\begin{aligned} A &\rightarrow B\ [0.80] \\ C &\rightarrow D\ [0.50] \\ B \wedge D &\rightarrow E\ [0.90] \\ E \vee F &\rightarrow G\ [0.25] \\ H &\rightarrow G\ [0.30] \end{aligned}$$

$$\begin{aligned} A\ [1.00] \\ C\ [0.50] \\ F\ [0.80] \\ H\ [0.90] \end{aligned}$$



Propagation Rules

Conjunction: $\mathrm{CF}(A \wedge B) = \min\{\mathrm{CF}(A), \mathrm{CF}(B)\}$

Disjunction: $\mathrm{CF}(A \vee B) = \max\{\mathrm{CF}(A), \mathrm{CF}(B)\}$

Serial Combination: $\mathrm{CF}(B, \{A\}) = \mathrm{CF}(A \rightarrow B) \cdot \max\{0, \mathrm{CF}(A)\}$

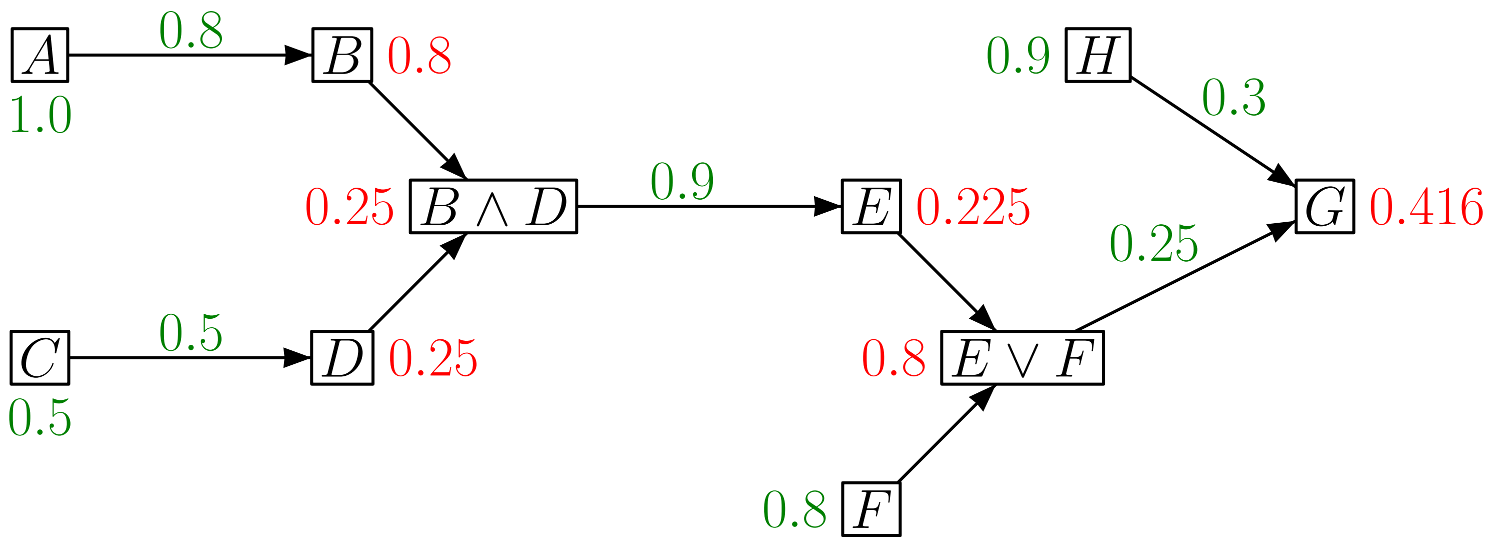
Parallel Combination: for $n > 1$:

$$\mathrm{CF}(B, \{A_1, \dots, A_n\}) = f(\mathrm{CF}(B, \{A_1, \dots, A_{n-1}\}), \mathrm{CF}(B, \{A_n\}))$$

with

$$f(x, y) = \begin{cases} x + y - xy & \text{if} \quad x, y > 0 \\ x + y + xy & \text{if} \quad x, y < 0 \\ \dfrac{x + y}{1 - \min\{|x|, |y|\}} & \text{otherwise} \end{cases}$$

Example (cont.)



$$f(0.3 \cdot 0.9, 0.25 \cdot 0.8) \;=\; 0.27 + 0.2 - 0.27 \cdot 0.2 = 0.416$$

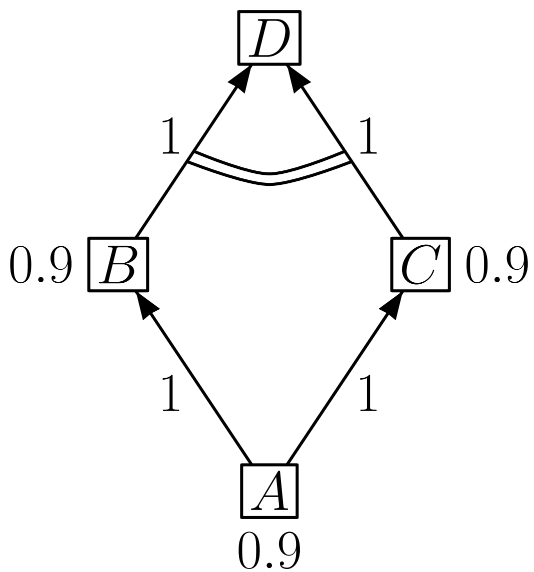
Was Mycin a failure?

It worked in the Mycin case because the rules had tree-like structure.

It can be shown that the rule combination scheme is inconsistent in general.

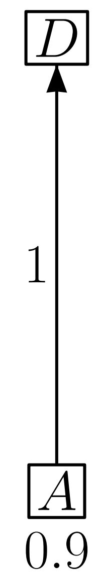
Example: $\mathrm{CF}(A) = 0.9$, $\mathrm{CF}(D) = ?$

$\mathrm{CF}(D) = 0.9 + 0.9 - 0.9 \cdot 0.9 = 0.99$



vs.

$\mathrm{CF}(D) = 0.9$



Certainty factor is increased just because (the same) evidence is transferred over different (parallel) paths!

Was Mycin a failure?

Mycin was never used for its intended purpose, because

physicians were distrustful and not willing to accept Mycin's recommendations.

Mycin was too good.

However,

Mycin was a milestone for the development of expert systems.

it gave rise to impulses for expert system development in general.

Probability Foundations

Reminder: Probability Theory

Goal: Make statements and/or predictions about results of physical processes.

Even processes that seem to be simple at first sight may reveal considerable difficulties when trying to predict.

Describing real-world physical processes always calls for a simplifying mathematical model.

Although everybody will have some intuitive notion about probability, we have to formally define the underlying mathematical structure.

Randomness or chance enters as the incapability of precisely modelling a process or the inability of measuring the initial conditions.

- *Example*: Predicting the trajectory of a billard ball over more than 9 banks requires more detailed measurement of the initial conditions (ball location, applied momentum etc.) than physically possible according to Heisenberg's uncertainty principle.

Formal Approach on the Model Side

We conduct an experiment that has a set Ω of possible outcomes.
E. g.:

- Rolling a die ($\Omega = \{1, 2, 3, 4, 5, 6\}$)
- Arrivals of phone calls ($\Omega = \mathbb{N}_0$)
- Bread roll weights ($\Omega = \mathbb{R}_+$)

Such an outcome is called an **elementary event**.

All possible elementary events are called the **frame of discernment** Ω (or sometimes **universe of discourse**).

The set representation stresses the following facts:

- All possible outcomes are covered by the elements of Ω. (**collectively exhaustive**).
- Every possible outcome is represented by exactly one element of Ω. (**mutual disjoint**).

Events

Often, we are interested in *higher-level* events
(e. g. casting an odd number, arrival of at least 5 phone calls or purchasing a bread roll heavier than 80 grams)

Any subset $A \subseteq \Omega$ is called an **event** which **occurs**, if the outcome $\omega_0 \in \Omega$ of the random experiment lies in A:

$$\text{Event } A \subseteq \Omega \text{ occurs} \quad \Leftrightarrow \quad \bigvee_{\omega \in A} (\omega = \omega_0) = \mathsf{true} \quad \Leftrightarrow \quad \omega_0 \in A$$

Since events are sets, we can define for two events A and B:

- $A \cup B$ occurs if A or B occurs; $A \cap B$ occurs if A and B occurs.
- $\overline{A}$ occurs if A does not occur (i. e., if $\Omega \backslash A$ occurs).
- A and B are *mutually exclusive*, iff $A \cap B = \emptyset$.

Event Algebra

A family of sets $\mathcal{E} = \{E_1, \ldots, E_n\}$ is called an **event algebra**, if the following conditions hold:

- The **certain event** Ω lies in $\mathcal{E}$.
- If $E \in \mathcal{E}$, then $\overline{E} = \Omega \backslash E \in \mathcal{E}$.
- If E_1 and E_2 lie in $\mathcal{E}$, then $E_1 \cup E_2 \in \mathcal{E}$ and $E_1 \cap E_2 \in \mathcal{E}$.

If Ω is uncountable, we require the additional property:

For a series of events $E_i \in \mathcal{E}, i \in \mathbb{N},$ the events $\bigcup_{i=1}^{\infty} E_i$ and $\bigcap_{i=1}^{\infty} E_i$ are also in $\mathcal{E}$.

$\mathcal{E}$ is then called a **σ-algebra**.

Side remarks:

Smallest event algebra: $\mathcal{E} = \{\emptyset, \Omega\}$

Largest event algebra (for finite or countable Ω): $\mathcal{E} = 2^{\Omega} = \{A \subseteq \Omega \mid \mathsf{true}\}$

Probability Function

Given an event algebra $\mathcal{E}$, we would like to assign every event $E \in \mathcal{E}$ its probability with a **probability function** $P: \mathcal{E} \to [0, 1]$.

We require P to satisfy the so-called **Kolmogorov Axioms**:

- $\forall E \in \mathcal{E}: \; 0 \leq P(E) \leq 1$
- $P(\Omega) = 1$
- For pairwise disjoint events $E_1, E_2, \ldots \in \mathcal{E}$ holds:

$$P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i)$$

From these axioms one can conclude the following (incomplete) list of properties:

- $\forall E \in \mathcal{E}: \; P(\overline{E}) = 1 - P(E)$
- $P(\emptyset) = 0$
- If $E_1, E_2 \in \mathcal{E}$ are mutually exclusive, then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$.

Elementary Probabilities and Densities

Question 1: How to calculate P?

Question 2: Are there "default" event algebras?

Idea for question 1: We have to find a way of distributing (thus the notion *distribution*) the unit mass of probability over all elements $\omega \in \Omega$.

- If Ω is finite or countable a **probability mass function** p is used:

$$p: \ \Omega \to [0,1] \quad \text{and} \quad \sum_{\omega \in \Omega} p(\omega) = 1$$

- If Ω is uncountable (i. e., continuous) a **probability density function** f is used:

$$f: \ \Omega \to \mathbb{R} \quad \text{and} \quad \int_{\Omega} f(\omega) \, \mathrm{d}\omega = 1$$

"Default" Event Algebras

Idea for question 2 ("default" event algebras) we have to distinguish again between the cardinalities of Ω:

- Ω finite or countable: $\mathcal{E} = 2^\Omega$
- Ω uncountable, e. g. $\Omega = \mathbb{R}$: $\mathcal{E} = \mathcal{B}(\mathbb{R})$

$\mathcal{B}(\mathbb{R})$ is the **Borel Algebra**, i. e., the smallest σ-algebra that contains all closed intervals $[a, b] \subset \mathbb{R}$ with $a < b$.

$\mathcal{B}(\mathbb{R})$ also contains all open intervals and single-item sets.

It is sufficient to note here, that all intervals are contained

$$\{[a, b]\,,]a, b]\,,]a, b[\,, [a, b[\, \subset \mathbb{R} \mid a < b\} \subset \mathcal{B}(\mathbb{R})$$

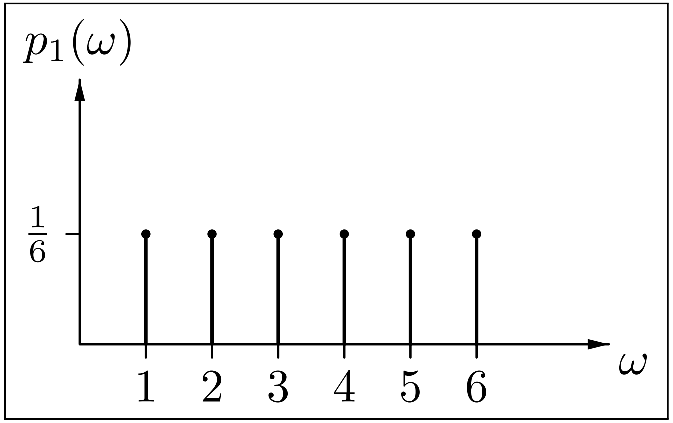
because the event of a bread roll having a weight between 80 g and 90 g is represented by the interval $[80, 90]$.

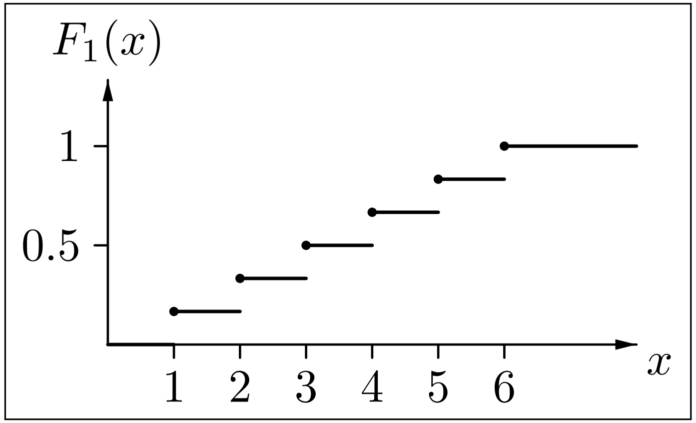
Example: Rolling a Die

$\Omega = \{1, 2, 3, 4, 5, 6\} \quad X = \text{id}$

$$p_1(\omega) = \tfrac{1}{6}$$

$$F_1(x) = P(X \leq x)$$





$$\sum_{\omega \in \Omega} p_1(\omega) = \sum_{i=1}^{6} p_1(\omega_i) = \sum_{i=1}^{6} \frac{1}{6} = 1$$

$$P(X \leq x) = \sum_{x' \leq x} P(X = x')$$

$$P(a < X \leq b) = F_1(b) - F_1(a)$$

$$P(X = x) = P(\{X = x\}) = P(X^{-1}(x)) = P(\{\omega \in \Omega \mid X(\omega) = x\})$$

Basics of Applied Probability Theory

Why (Kolmogorov) Axioms?

If P models an *objectively* observable probability, these axioms are obviously reasonable.

However, why should an agent obey formal axioms when modeling degrees of (subjective) belief?

Objective vs. subjective probabilities

Axioms constrain the set of beliefs an agent can abide.

Finetti (1931) gave one of the most plausible arguments why subjective beliefs should respect axioms:

> "When using contradictory beliefs, the agent will eventually fail."

Unconditional Probabilities

$P(A)$ designates the *unconditioned* or *a priori* probability that $A \subseteq \Omega$ occurs if *no* other additional information is present.
For example:

$$P(\mathsf{cavity}) \;=\; 0.1$$

Note: Here, cavity is a proposition.

A formally different way to state the same would be via a binary random variable Cavity:

$$P(\mathsf{Cavity} = \mathsf{true}) \;=\; 0.1$$

A priori probabilities are derived from statistical surveys or general rules.

Unconditional Probabilities

In general a random variable can assume more than two values:

$$\begin{aligned}
P(\ \mathsf{Weather} = \mathsf{sunny}\ ) &= 0.7 \\
P(\ \mathsf{Weather} = \mathsf{rainy}\ ) &= 0.2 \\
P(\ \mathsf{Weather} = \mathsf{cloudy}) &= 0.02 \\
P(\ \mathsf{Weather} = \mathsf{snowy}) &= 0.08 \\
P(\mathsf{Headache} = \mathsf{true}\ ) &= 0.1
\end{aligned}$$

$P(X)$ designates the vector of probabilities for the (ordered) domain of the random variable X:

$$\begin{aligned}
P(\mathsf{Weather}) &= \langle 0.7, 0.2, 0.02, 0.08 \rangle \\
P(\mathsf{Headache}) &= \langle 0.1, 0.9 \rangle
\end{aligned}$$

Both vectors define the respective probability distributions of the two random variables.

Conditional Probabilities

New evidence can alter the probability of an event.

Example: The probability for cavity increases if information about a toothache arises.

With additional information present, the a priori knowledge must not be used!

$P(A \mid B)$ designates the *conditional* or *a posteriori* probability of A *given* the sole observation (*evidence*) B.

$$P(\mathsf{cavity} \mid \mathsf{toothache}) = 0.8$$

For random variables X and Y $P(X \mid Y)$ represents the set of conditional distributions for each possible value of Y.

Conditional Probabilities

$P(\text{Weather} \mid \text{Headache})$ consists of the following table:

	$\text{h} \mathrel{\widehat{=}} \text{Headache} = \text{true}$	$\neg\text{h} \mathrel{\widehat{=}} \text{Headache} = \text{false}$
$\text{Weather} = \text{sunny}$	$P(\text{W} = \text{sunny} \mid \text{h})$	$P(\text{W} = \text{sunny} \mid \neg\text{h})$
$\text{Weather} = \text{rainy}$	$P(\text{W} = \text{rainy} \mid \text{h})$	$P(\text{W} = \text{rainy} \mid \neg\text{h})$
$\text{Weather} = \text{cloudy}$	$P(\text{W} = \text{cloudy} \mid \text{h})$	$P(\text{W} = \text{cloudy} \mid \neg\text{h})$
$\text{Weather} = \text{snowy}$	$P(\text{W} = \text{snowy} \mid \text{h})$	$P(\text{W} = \text{snowy} \mid \neg\text{h})$

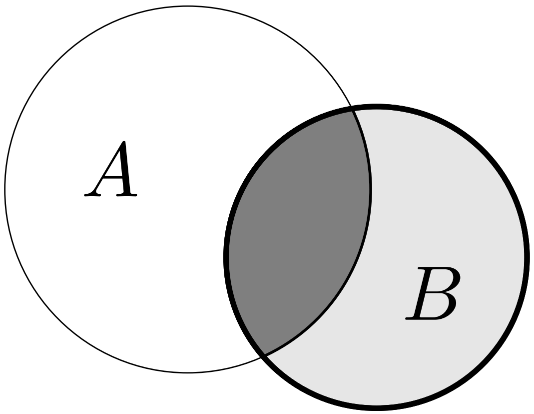
Note that we are dealing with *two* distributions now!
Therefore each column sums up to unity!

Formal definition:

$$P(A \mid B) = \frac{P(A \land B)}{P(B)} \qquad \text{if} \quad P(B) > 0$$

Conditional Probabilities

$$\boxed{P(A \mid B) \;=\; \frac{P(A \wedge B)}{P(B)}}$$



Product Rule: $P(A \wedge B) \;=\; P(A \mid B) \cdot P(B)$

Also: $P(A \wedge B) \;=\; P(B \mid A) \cdot P(A)$

A and B are *independent* iff

$$P(A \mid B) \;=\; P(A) \qquad \text{and} \qquad P(B \mid A) \;=\; P(B)$$

Equivalently, iff the following equation holds true:

$$P(A \wedge B) \;=\; P(A) \cdot P(B)$$

Interpretation of Conditional Probabilities

Caution! Common misinterpretation:

"$P(A \mid B) = 0.8$ means, that $P(A) = 0.8$, given B holds."

This statement is wrong due to (at least) two facts:

$P(A)$ is *always* the a-priori probability,
never the probability of A given that B holds!

$P(A \mid B) = 0.8$ is only applicable as long as no other evidence except B is present.
If C becomes known, $P(A \mid B \wedge C)$ has to be determined.

In general we have:

$$P(A \mid B \wedge C) \neq P(A \mid B)$$

E. g. $C \rightarrow A$ might apply.

Joint Probabilities

Let $X_1, \ldots, X_n$ be random variables over the same framce of descernment Ω and event algebra $\mathcal{E}$. Then $\vec{X} = (X_1, \ldots, X_n)$ is called a *random vector* with

$$\vec{X}(\omega) \;=\; (X_1(\omega), \ldots, X_n(\omega))$$

Shorthand notation:

$$P(\vec{X} = (x_1, \ldots, x_n)) \;=\; P(X_1 = x_1, \ldots, X_n = x_n) \;=\; P(x_1, \ldots, x_n)$$

Definition:

$$\begin{aligned} P(X_1 = x_1, \ldots, X_n = x_n) \;&=\; P\Big(\Big\{\, \omega \in \Omega \;\Big|\; \bigwedge_{i=1}^{n} X_i(\omega) = x_i \,\Big\}\Big) \\ &=\; P\Big(\bigcap_{i=1}^{n} \{X_i = x_i\} \Big) \end{aligned}$$

Joint Probabilities

Example: $P(\mathsf{Headache}, \mathsf{Weather})$ is the *joint probability distribution* of both random variables and consists of the following table:

	$\mathsf{h} \mathrel{\widehat{=}} \mathsf{Headache} = \mathsf{true}$	$\neg\mathsf{h} \mathrel{\widehat{=}} \mathsf{Headache} = \mathsf{false}$
$\mathsf{Weather} = \mathsf{sunny}$	$P(\mathsf{W} = \mathsf{sunny} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{sunny} \wedge \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{rainy}$	$P(\mathsf{W} = \mathsf{rainy} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{rainy} \wedge \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{cloudy}$	$P(\mathsf{W} = \mathsf{cloudy} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{cloudy} \wedge \neg\mathsf{h})$
$\mathsf{Weather} = \mathsf{snowy}$	$P(\mathsf{W} = \mathsf{snowy} \wedge \mathsf{h})$	$P(\mathsf{W} = \mathsf{snowy} \wedge \neg\mathsf{h})$

All table cells sum up to unity.

Calculating with Joint Probabilities

All desired probabilities can be computed from a joint probability distribution.

	toothache	¬toothache
cavity	0.04	0.06
¬cavity	0.01	0.89

Example:
$$\begin{aligned} P(\textsf{cavity} \vee \textsf{toothache}) &= P(\textsf{cavity} \wedge \textsf{toothache}) \\ &+ P(\neg\textsf{cavity} \wedge \textsf{toothache}) \\ &+ P(\textsf{cavity} \wedge \neg\textsf{toothache}) = 0.11 \end{aligned}$$

Marginalizations:
$$\begin{aligned} \mathrm{P}(\textsf{cavity}) &= P(\textsf{cavity} \wedge \textsf{toothache}) \\ &+ P(\textsf{cavity} \wedge \neg\textsf{toothache}) = 0.10 \end{aligned}$$

Conditioning:
$$P(\textsf{cavity} \mid \textsf{toothache}) = \frac{P(\textsf{cavity} \wedge \textsf{toothache})}{P(\textsf{toothache})} = \frac{0.04}{0.04 + 0.01} = 0.80$$

Problems

Easiness of computing all desired probabilities comes at an unaffordable price:

Given n random variables with k possible values each, the joint probability distribution contains k^n entries which is infeasible in practical applications.

Hard to handle.

Hard to estimate.

Therefore:

1. Is there a more *dense* representation of joint probability distributions?

2. Is there a more *efficient* way of processing this representation?

The answer is *no* for the general case, however, certain dependencies and independencies can be exploited to reduce the number of parameters to a practical size.

Stochastic Independence

Two events A and B are *stochastically independent* iff

$$P(A \wedge B) \;=\; P(A) \cdot P(B)$$
$$\Leftrightarrow$$
$$P(A \mid B) \;=\; P(A) \;=\; P(A \mid \overline{B})$$

Two random variables X and Y are *stochastically independent* iff

$$\forall x \in \mathrm{dom}(X) : \forall y \in \mathrm{dom}(Y) : \qquad P(X = x, Y = y) \;=\; P(X = x) \cdot P(Y = y)$$
$$\Leftrightarrow$$
$$\forall x \in \mathrm{dom}(X) : \forall y \in \mathrm{dom}(Y) : \qquad P(X = x \mid Y = y) \;=\; P(X = x)$$

Shorthand notation: $P(X, Y) = P(X) \cdot P(Y)$.
Note the formal difference between $P(A) \in [0, 1]$ and $P(X) \in [0, 1]^{|\mathrm{dom}(X)|}$.

Conditional Independence

Let X, Y and Z be three random variables. We call *X and Y conditionally independent given Z*, iff the following condition holds:

$$\forall x \in \mathrm{dom}(X) : \forall y \in \mathrm{dom}(Y) : \forall z \in \mathrm{dom}(Z) :$$
$$P(X = x, Y = y \mid Z = z) \;=\; P(X = x \mid Z = z) \cdot P(Y = y \mid Z = z)$$

Shorthand notation: $X \perp\!\!\!\perp_P Y \mid Z$

Let $\boldsymbol{X} = \{A_1, \ldots, A_k\}$, $\boldsymbol{Y} = \{B_1, \ldots, B_l\}$ and $\boldsymbol{Z} = \{C_1, \ldots, C_m\}$ be three disjoint sets of random variables. We call *$\boldsymbol{X}$ and $\boldsymbol{Y}$ conditionally independent given $\boldsymbol{Z}$*, iff

$$P(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{Z}) \;=\; P(\boldsymbol{X} \mid \boldsymbol{Z}) \cdot P(\boldsymbol{Y} \mid \boldsymbol{Z}) \Leftrightarrow P(\boldsymbol{X} \mid \boldsymbol{Y}, \boldsymbol{Z}) \;=\; P(\boldsymbol{X} \mid \boldsymbol{Z})$$

Shorthand notation: $\boldsymbol{X} \perp\!\!\!\perp_P \boldsymbol{Y} \mid \boldsymbol{Z}$

Conditional Independence

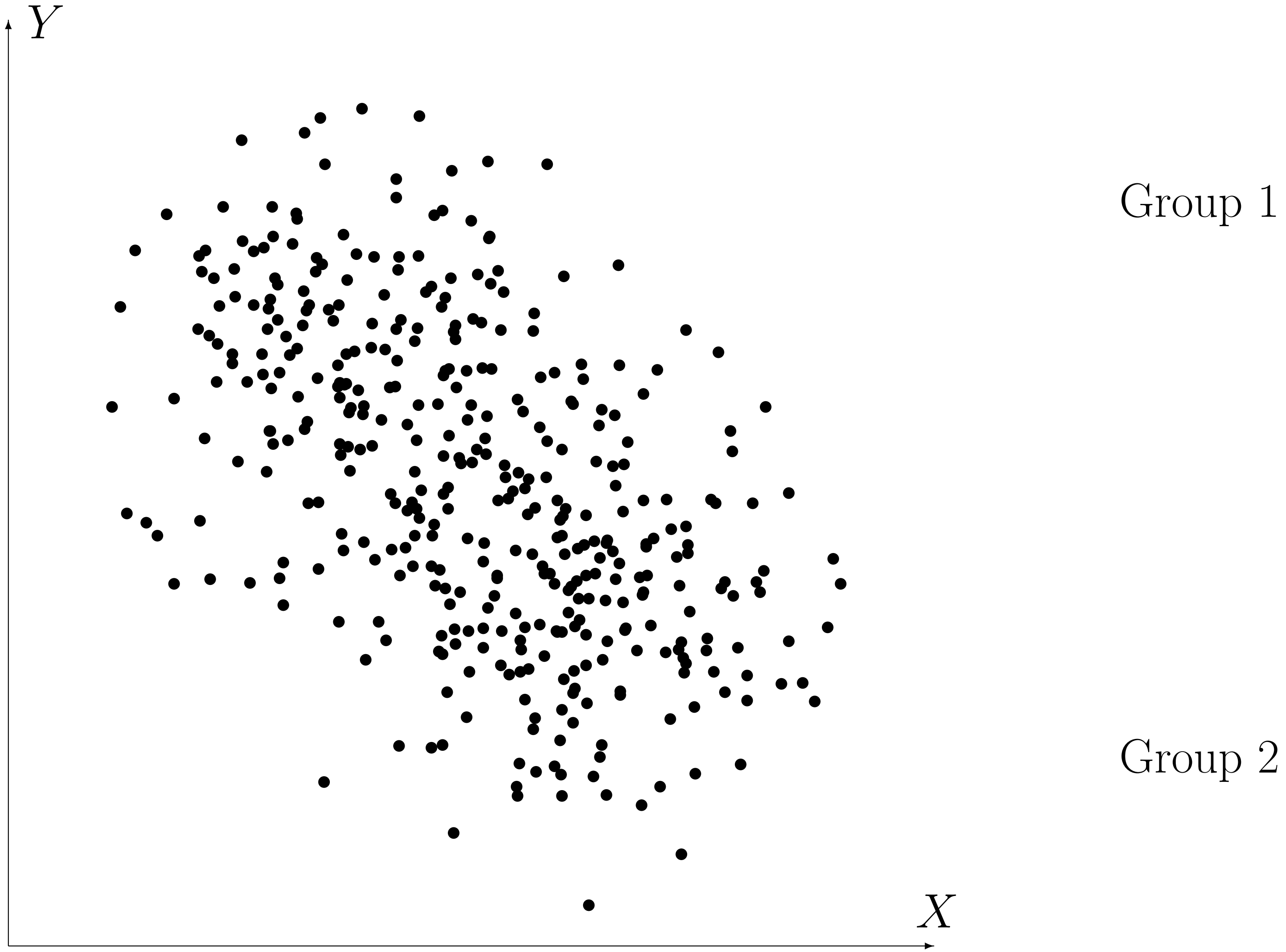
The complete condition for $\boldsymbol{X} \perp\!\!\!\perp_P \boldsymbol{Y} \mid \boldsymbol{Z}$ would read as follows:

$$
\begin{aligned}
&\forall a_1 \in \mathrm{dom}(A_1) : \cdots \forall a_k \in \mathrm{dom}(A_k) : \\
&\quad \forall b_1 \in \mathrm{dom}(B_1) : \cdots \forall b_l \in \mathrm{dom}(B_l) : \\
&\qquad \forall c_1 \in \mathrm{dom}(C_1) : \cdots \forall c_m \in \mathrm{dom}(C_m) : \\
&\qquad\quad P(A_1 = a_1, \ldots, A_k = a_k, B_1 = b_1, \ldots, B_l = b_l \mid C_1 = c_1, \ldots, C_m = c_m) \\
&\qquad\qquad = P(A_1 = a_1, \ldots, A_k = a_k \mid C_1 = c_1, \ldots, C_m = c_m) \\
&\qquad\qquad\ \cdot P(B_1 = b_1, \ldots, B_l = b_l \mid C_1 = c_1, \ldots, C_m = c_m)
\end{aligned}
$$

Remarks:

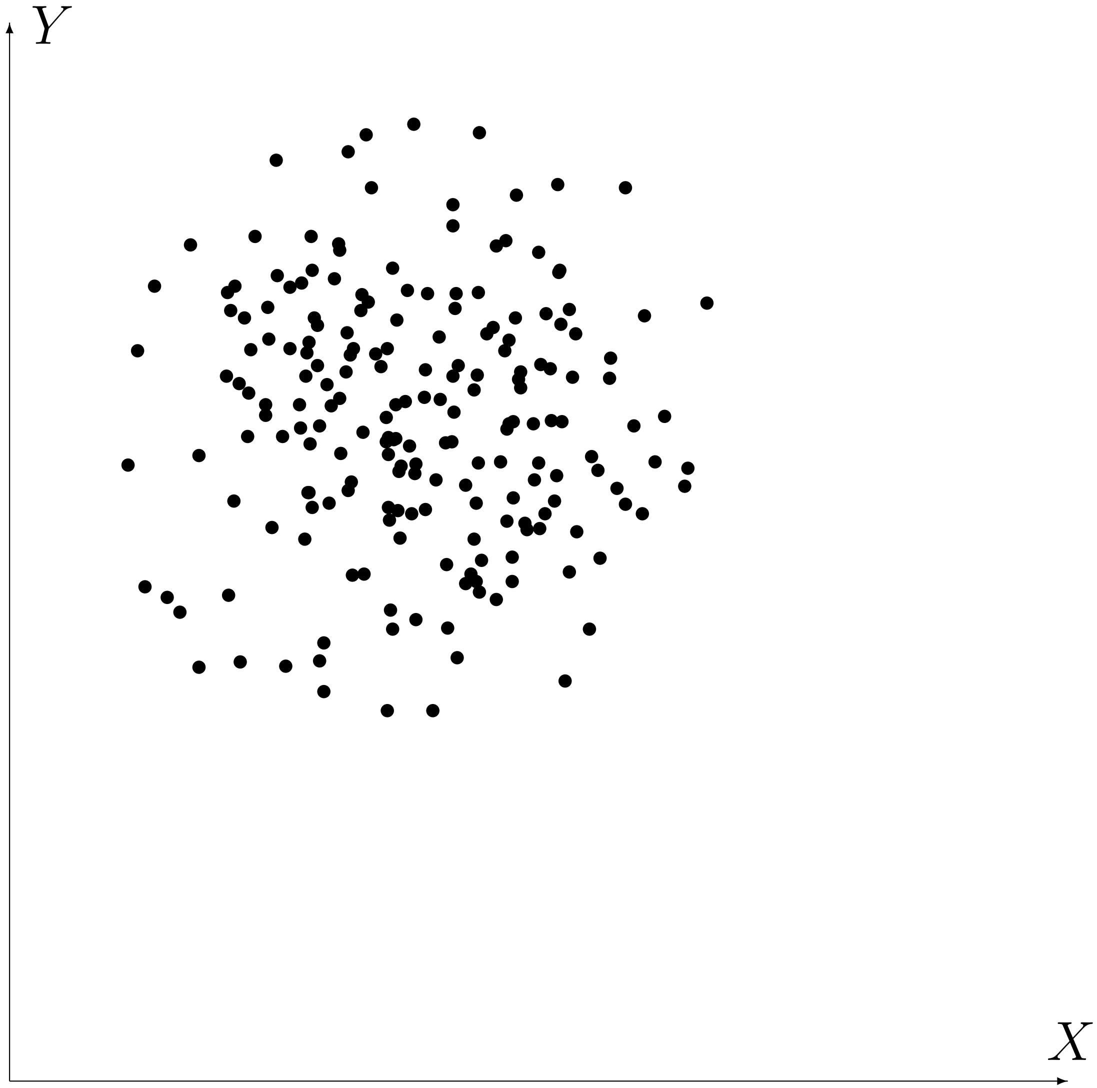
1. If $\boldsymbol{Z} = \emptyset$ we get (unconditional) independence.
2. We do not use curly braces ($\{\}$) for the sets if the context is clear. Likewise, we use X instead of $\boldsymbol{X}$ to denote sets.

Conditional Independence — Example 1



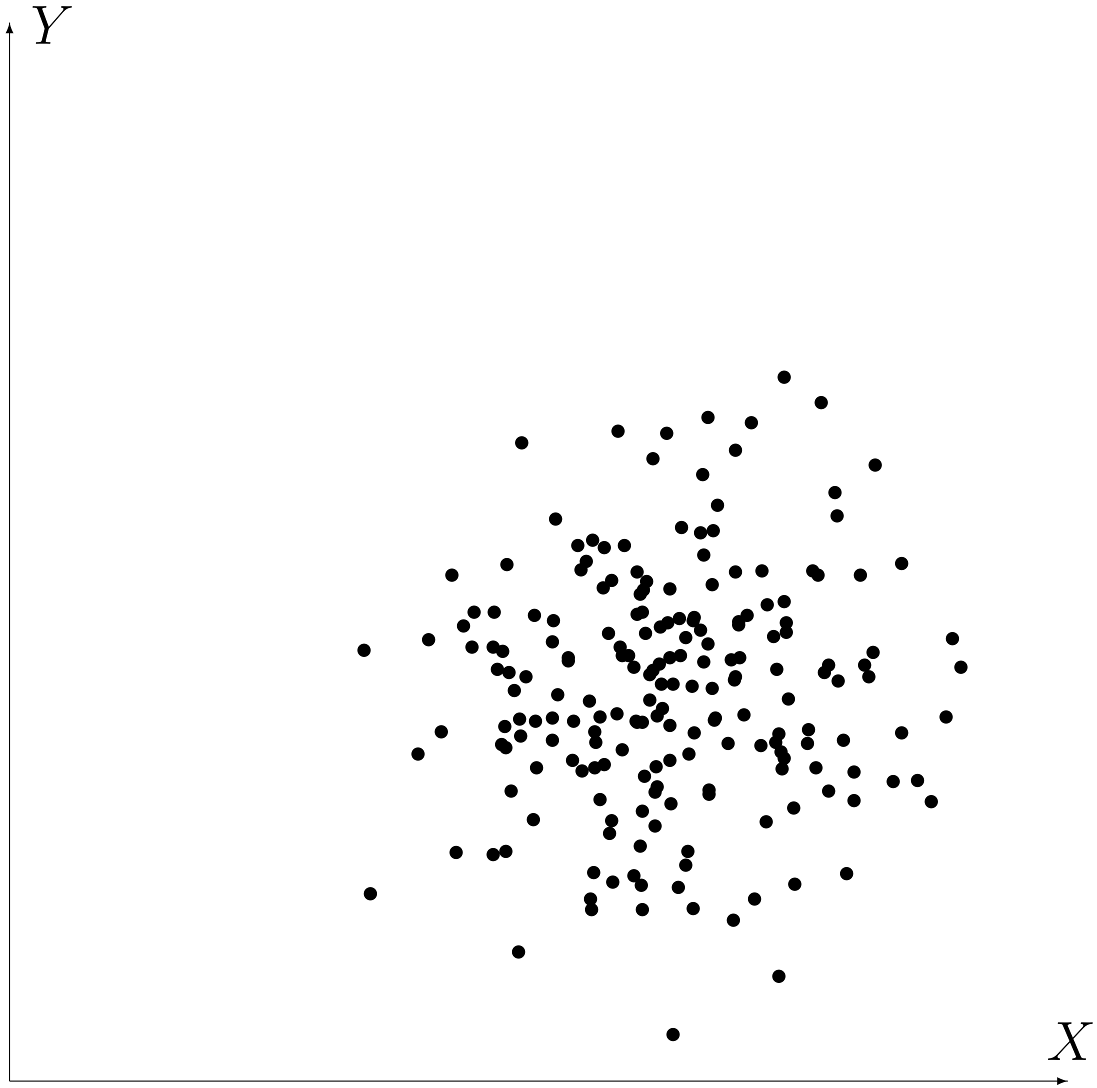
(Weak) Dependence in the entire dataset: X and Y dependent.

Conditional Independence — Example 1



No Dependence in Group 1: X and Y conditionally independent given Group 1.

Conditional Independence — Example 1



Group 2

No Dependence in Group 2: X and Y conditionally independent given Group 2.

Conditional Independence — Example 2

- $\mathrm{dom}(G) = \{\mathsf{mal}, \mathsf{fem}\}$ Geschlecht (gender)
- $\mathrm{dom}(S) = \{\mathsf{sm}, \overline{\mathsf{sm}}\}$ Raucher (smoker)
- $\mathrm{dom}(M) = \{\mathsf{mar}, \overline{\mathsf{mar}}\}$ Verheiratet (married)
- $\mathrm{dom}(P) = \{\mathsf{preg}, \overline{\mathsf{preg}}\}$ Schwanger (pregnant)

p_{GSMP}		G = mal		G = fem	
		S = sm	S = $\overline{\mathsf{sm}}$	S = sm	S = $\overline{\mathsf{sm}}$
M = mar	P = preg	0	0	0.01	0.05
	P = $\overline{\mathsf{preg}}$	0.04	0.16	0.02	0.12
M = $\overline{\mathsf{mar}}$	P = preg	0	0	0.01	0.01
	P = $\overline{\mathsf{preg}}$	0.10	0.20	0.07	0.21

Conditional Independence — Example 2

$$P(\mathsf{G}=\mathsf{fem}) = P(\mathsf{G}=\mathsf{mal}) = 0.5 \qquad P(\mathsf{P}=\mathsf{preg}) = 0.08$$
$$P(\mathsf{S}=\mathsf{sm}) = 0.25 \qquad P(\mathsf{M}=\mathsf{mar}) = 0.4$$

Gender and Smoker are not independent:

$$P(\mathsf{G}=\mathsf{fem} \mid \mathsf{S}=\mathsf{sm}) = 0.44 \neq 0.5 = P(\mathsf{G}=\mathsf{fem})$$

Gender and Marriage are marginally independent but conditionally dependent given Pregnancy:

$$P(\mathsf{fem}, \mathsf{mar} \mid \overline{\mathsf{preg}}) \approx 0.152 \quad \neq \quad 0.169 \approx P(\mathsf{fem} \mid \overline{\mathsf{preg}}) \cdot P(\mathsf{mar} \mid \overline{\mathsf{preg}})$$

Bayes Theorem

Product Rule (for events A and B):

$$P(A \cap B) = P(A \mid B)P(B) \quad \text{and} \quad P(A \cap B) = P(B \mid A)P(A)$$

Equating the right-hand sides:

$$P(A \mid B) \; = \; \frac{P(B \mid A)P(A)}{P(B)}$$

For random variables X and Y:

$$\forall x \forall y : \quad P(Y\!=\!y \mid X\!=\!x) \; = \; \frac{P(X\!=\!x \mid Y\!=\!y)P(Y\!=\!y)}{P(X\!=\!x)}$$

Generalization concerning background knowledge/evidence E:

$$P(Y \mid X, E) \; = \; \frac{P(X \mid Y, E)P(Y \mid E)}{P(X \mid E)}$$

Bayes Theorem — Application

$$P(\mathsf{toothache} \mid \mathsf{cavity}) = 0.4$$
$$P(\mathsf{cavity}) = 0.1 \qquad P(\mathsf{cavity} \mid \mathsf{toothache}) = \frac{0.4 \cdot 0.1}{0.05} = 0.8$$
$$P(\mathsf{toothache}) = 0.05$$

Why not estimate $P(\mathsf{cavity} \mid \mathsf{toothache})$ right from the start?

Causal knowledge like $P(\mathsf{toothache} \mid \mathsf{cavity})$ is more robust than diagnostic knowledge $P(\mathsf{cavity} \mid \mathsf{toothache})$.

The causality $P(\mathsf{toothache} \mid \mathsf{cavity})$ is independent of the a priori probabilities $P(\mathsf{toothache})$ and $P(\mathsf{cavity})$.

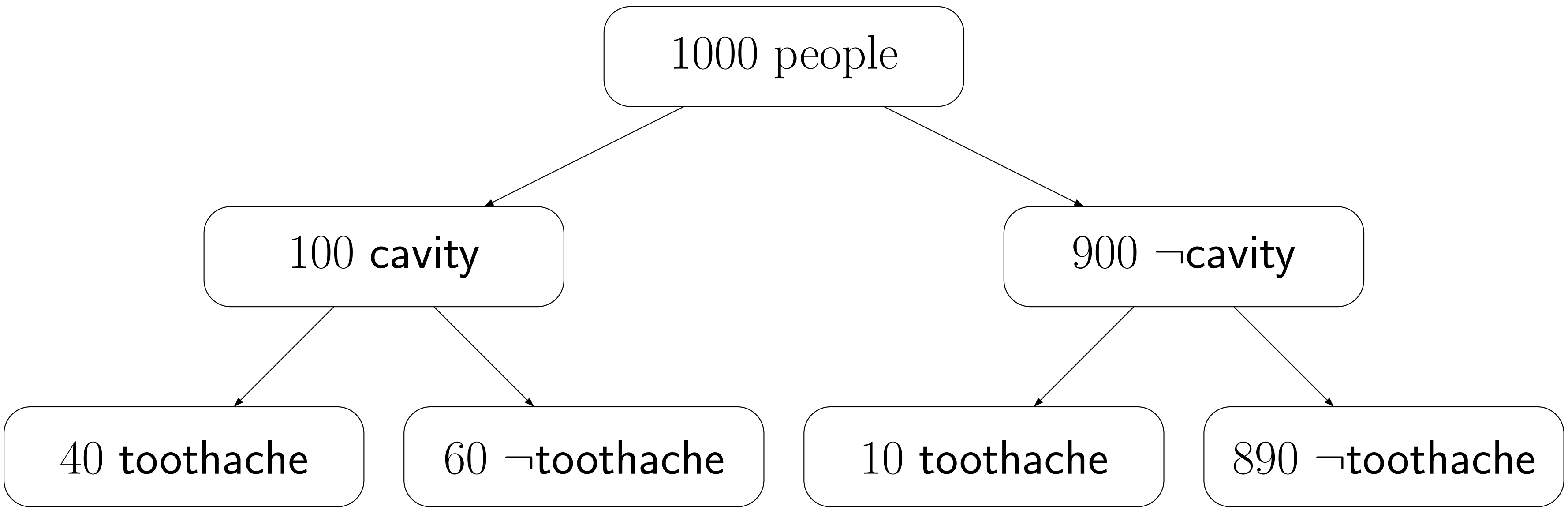
If $P(\mathsf{cavity})$ rose in a caries epidemic, the causality $P(\mathsf{toothache} \mid \mathsf{cavity})$ would remain constant whereas both $P(\mathsf{cavity} \mid \mathsf{toothache})$ and $P(\mathsf{toothache})$ would increase according to $P(\mathsf{cavity})$.

A physician, after having estimated $P(\mathsf{cavity} \mid \mathsf{toothache})$, would not know a rule for updating.

Bayes Theorem — Using absolute Numbers

$P(\mathsf{toothache} \mid \mathsf{cavity}) = 0.4$ $\qquad$ $P(\mathsf{cavity}) = 0.1$

$P(\mathsf{toothache} \mid \neg\mathsf{cavity}) = \frac{1}{90}$ $\qquad$ $P(\mathsf{cavity} \mid \mathsf{toothache}) = \dfrac{40}{40+10} = 0.8$



$$P(C \mid T) = \frac{P(T \mid C) \cdot P(C)}{P(T)} = \frac{P(T \mid C) \cdot P(C)}{P(T \mid C) \cdot P(C) + P(T \mid \neg C) \cdot P(\neg C)}$$

Relative Probabilities

Assumption:
We would like to consider the probability of the diagnosis $\mathsf{GumDisease}$ as well.

$$\begin{aligned} P(\mathsf{toothache} \mid \mathsf{gumdisease}) &= 0.7 \\ P(\mathsf{gumdisease}) &= 0.02 \end{aligned}$$

Which diagnosis is more probable?

If we are interested in *relative probabilities* only (which may be sufficient for some decisions), $P(\mathsf{toothache})$ needs not to be estimated:

$$\begin{aligned} \frac{P(C \mid T)}{P(G \mid T)} &= \frac{P(T \mid C)P(C)}{P(T)} \cdot \frac{P(T)}{P(T \mid G)P(G)} \\ &= \frac{P(T \mid C)P(C)}{P(T \mid G)P(G)} = \frac{0.4 \cdot 0.1}{0.7 \cdot 0.02} \\ &= 28.57 \end{aligned}$$

Normalization

If we are interested in the absolute probability of $P(C \mid T)$ but do not know $P(T)$, we may conduct a complete case analysis (according C) and exploit the fact that $P(C \mid T) \;+\; P(\neg C \mid T) \;=\; 1$.

$$
\begin{aligned}
P(C \mid T) &= \frac{P(T \mid C)P(C)}{P(T)} \\
P(\neg C \mid T) &= \frac{P(T \mid \neg C)P(\neg C)}{P(T)} \\
1 \;=\; P(C \mid T) \;+\; P(\neg C \mid T) &= \frac{P(T \mid C)P(C)}{P(T)} \;+\; \frac{P(T \mid \neg C)P(\neg C)}{P(T)} \\
P(T) &= P(T \mid C)P(C) \;+\; P(T \mid \neg C)P(\neg C)
\end{aligned}
$$

Normalization

Plugging into the equation for $P(C \mid T)$ yields:

$$P(C \mid T) = \frac{P(T \mid C)P(C)}{P(T \mid C)P(C) + P(T \mid \neg C)P(\neg C)}$$

For general random variables, the equation reads:

$$P(Y\!=\!y \mid X\!=\!x) = \frac{P(X\!=\!x \mid Y\!=\!y)P(Y\!=\!y)}{\sum\limits_{\forall y' \in \mathrm{dom}(Y)} P(X\!=\!x \mid Y\!=\!y')P(Y\!=\!y')}$$

Note the "loop variable" y'. Do not confuse with y.

Multiple Evidences

The patient complains about a toothache. From this first evidence the dentist infers:

$$P(\textsf{cavity} \mid \textsf{toothache}) = 0.8$$

The dentist palpates the tooth with a metal probe which catches into a fracture:

$$P(\textsf{cavity} \mid \textsf{fracture}) = 0.95$$

Both conclusions might be inferred via Bayes rule. But what does the combined evidence yield? Using Bayes rule further, the dentist might want to determine:

$$P(\textsf{cavity} \mid \textsf{toothache} \land \textsf{fracture}) = \frac{P(\textsf{toothache} \land \textsf{fracture} \mid \textsf{cavity}) \cdot P(\textsf{cavity})}{P(\textsf{toothache} \land \textsf{fracture})}$$

Multiple Evidences

Problem:
He needs $P(\textsf{toothache} \wedge \textsf{catch} \mid \textsf{cavity})$, i. e. diagnostics knowledge for all combinations of symptoms in general. Better incorporate evidences step-by-step:

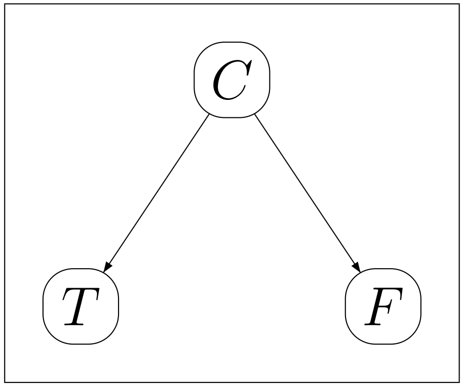
$$P(Y \mid X, E) = \frac{P(X \mid Y, E) P(Y \mid E)}{P(X \mid E)}$$

Abbreviations:

C — cavity

T — toothache

F — fracture



Objective:
Computing $P(C \mid T, F)$ with just using information about $P(\cdot \mid C)$ and under exploitation of independence relations among the variables.

Multiple Evidences

A priori: $P(C)$

Evidence toothache: $P(C \mid T) = P(C)\,\frac{P(T \mid C)}{P(T)}$

Evidence fracture: $P(C \mid T, F) = P(C \mid T)\,\frac{P(F \mid C, T)}{P(F \mid T)}$

Information about conditional independence
$P(F \mid C, T) = P(F \mid C)$

$$P(C \mid T, F) \;=\; P(C)\,\frac{P(T \mid C)}{P(T)}\,\frac{P(F \mid C)}{P(F \mid T)}$$

Seems that we still have to cope with symptom inter-dependencies?!

Multiple Evidences

Compound equation from last slide:

$$P(C \mid T, F) \;=\; P(C)\, \frac{P(T \mid C)\; P(F \mid C)}{P(T)\; P(F \mid T)}$$

$$\;=\; P(C)\, \frac{P(T \mid C)\; P(F \mid C)}{P(F, T)}$$

$P(F, T)$ is a normalizing constant and can be computed if $P(F \mid \neg C)$ and $P(T \mid \neg C)$ are known:

$$P(F, T) = \underbrace{P(F, T \mid C)}_{P(F|C)P(T|C)}\; P(C) + \underbrace{P(F, T \mid \neg C)}_{P(F|\neg C)P(T|\neg C)}\; P(\neg C)$$

Therefore, we finally arrive at the following solution...

Multiple Evidences

$$P(C \mid F, T) = \frac{\boxed{P(C)}\ \boxed{P(T \mid C)}\ \boxed{P(F \mid C)}}{\boxed{P(F \mid C)}\ \boxed{P(T \mid C)}\ \boxed{P(C)} + \boxed{P(F \mid \neg C)}\ \boxed{P(T \mid \neg C)}\ \boxed{P(\neg C)}}$$

Note that we only use causal probabilities $P(\ \cdot \mid C)$ together with the a priori (marginal) probabilities $P(C)$ and $P(\neg C)$.

Multiple Evidences — Summary

Multiple evidences can be treated by reduction on

a priori probabilities

(causal) conditional probabilities for the evidence

under assumption of conditional independence

General rule:

$$P(Z \mid X, Y) \; = \; \alpha \; P(Z) \; P(X \mid Z) \; P(Y \mid Z)$$

for X and Y conditionally independent given Z and with normalizing constant α.

Monty Hall Puzzle

Marylin Vos Savant in her riddle column in the New York Times:

You are a candidate in a game show and have to choose between three doors. Behind one of them is a Porsche, whereas behind the other two there are goats. After you chose a door, the host Monty Hall (who knows what is behind each door) opens another (not your chosen one) door with a goat. Now you have the choice between keeping your chosen door or choose the remaining one.

Which decision yields the best chance of winning the Porsche?

Monty Hall Puzzle

G	You win the Porsche.
R	You revise your decision.
A	Behind your initially chosen door is (and remains) the Porsche.

$$
\begin{aligned}
P(G \mid R) &= P(G, A \mid R) + P(G, \overline{A} \mid R) \\
&= P(G \mid A, R)P(A \mid R) + P(G \mid \overline{A}, R)P(\overline{A} \mid R) \\
&= 0 \cdot P(A \mid R) + 1 \cdot P(\overline{A} \mid R) \\
&= P(\overline{A} \mid R) = P(\overline{A}) = \frac{2}{3}
\end{aligned}
$$

$$
\begin{aligned}
P(G \mid \overline{R}) &= P(G, A \mid \overline{R}) + P(G, \overline{A} \mid \overline{R}) \\
&= P(G \mid A, \overline{R})P(A \mid \overline{R}) + P(G \mid \overline{A}, \overline{R})P(\overline{A} \mid \overline{R}) \\
&= 1 \cdot P(A \mid \overline{R}) + 0 \cdot P(\overline{A} \mid \overline{R}) \\
&= P(A \mid \overline{R}) = P(A) = \frac{1}{3}
\end{aligned}
$$

Simpson's Paradox

Example: C = Patient takes medication, E = patient recovers

	E	$\neg E$	$\sum$	Recovery rate
C	20	20	40	50%
$\neg C$	16	24	40	40%
$\sum$	36	44	80	

Men	E	$\neg E$	$\sum$	Rec.rate	Women	E	$\neg E$	$\sum$	Rec.rate
C	18	12	30	60%	C	2	8	10	20%
$\neg C$	7	3	10	70%	$\neg C$	9	21	30	30%
	25	15	40			11	29	40	

$$\begin{array}{rrcl} & P(E \mid C) & > & P(E \mid \neg C) \\ \text{but} & P(E \mid C, M) & < & P(E \mid \neg C, M) \\ & P(E \mid C, W) & < & P(E \mid \neg C, W) \end{array}$$

Excursus: Focusing vs. Revision

Philosophical topic, studied e.g. by Kant, Gärdenfors

Example for Focusing

- Prior knowledge: fair die
- New evidence: the result is an odd number
- Aposteriori knowledge via focusing: conditional probability
- Underlying probability measure did not change

Example for Revision

- Prior knowledge: fair die
- New evidence: weight near the 6
- Belief change via revision
- Underlying probability measure did change

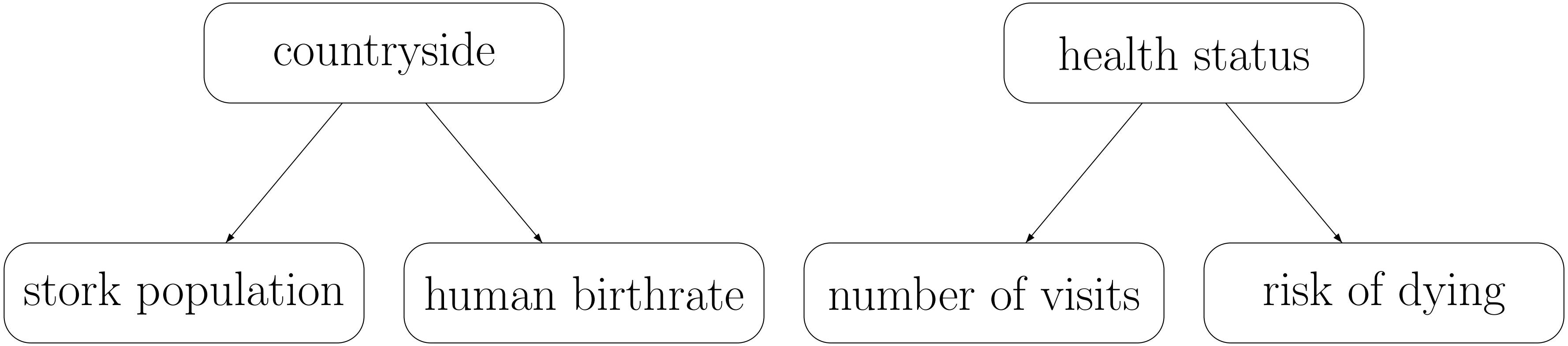
Excursus: Causality vs. Correlation

Philosophical topic, studied e.g. by Aristoteles, still under discussion

Press acceleration pedal → car is faster (causality)

Stork population high → human birthrate (correlation, but no causality)

Visit doctor often → high risk of dying (correlation, but no causality)



Probabilistic Reasoning

Probabilistic reasoning is difficult and may be problematic:

- $P(A \wedge B)$ is not determined simply by $P(A)$ and $P(B)$:
  $P(A) = P(B) = 0.5 \quad \Rightarrow \quad P(A \wedge B) \in [0, 0.5]$
- $P(C \mid A) = x, P(C \mid B) = y \quad \Rightarrow \quad P(C \mid A \wedge B) \in [0, 1]$
  Probabilistic logic is *not truth functional*!

Central problem: How does additional information affect the current knowledge?
I. e., if $P(B \mid A)$ is known, what can be said about $P(B \mid A \wedge C)$?

High complexity: n propositions $\rightarrow$ 2^n full conjunctives

Hard to specify these probabilities.

Summary

Uncertainty is inevitable in complex and dynamic scenarios that force agents to cope with ignorance.

Probabilities express the agent's inability to vote for a definitive decision. They model the degree of belief.

If an agent violates the axioms of probability, it may exhibit irrational behavior in certain circumstances.

The Bayes rule is used to derive unknown probabilities from present knowledge and new evidence.

Multiple evidences can be effectively included into computations exploiting conditional independencies.

Decomposition

Object Representation

Property family	Car body	Motor	Radio	Doors	Seat cover	Makeup mirrow	...
Property	Hatch-back	2.8 L 150 kW Otto	Type alpha	4	Leather, Type L3	yes	...

About 200 variables

Typically 4 to 8, but up to 150 possible instances per variable

More than 2^{200} possible combinations available

Example 1: Planning in car manufacturing

Available information: 10000 technical rules, 200 attributes

> “If Motor $= m_4$ and Heating $= h_1$ then Generator $\in \{g_1, g_2, g_3\}$”
>
> “Engine type e_1 can only be combined with transmission t_2 or t_5.”
>
> “Transmission t_5 requires crankshaft c_2.”
>
> “Convertibles have the same set of radio options as SUVs.”

Each peace of information corresponds to a constraint in a high dimensional subspace, possible questions/inferences:

> “Can a station wagon with engine e_4 be equipped with tire set y_6?”
>
> “Supplier S_8 failed to deliver on time. What production line has to be modified and how?”
>
> “Are there any peculiarities within the set of cars that suffered an aircondition failure?”

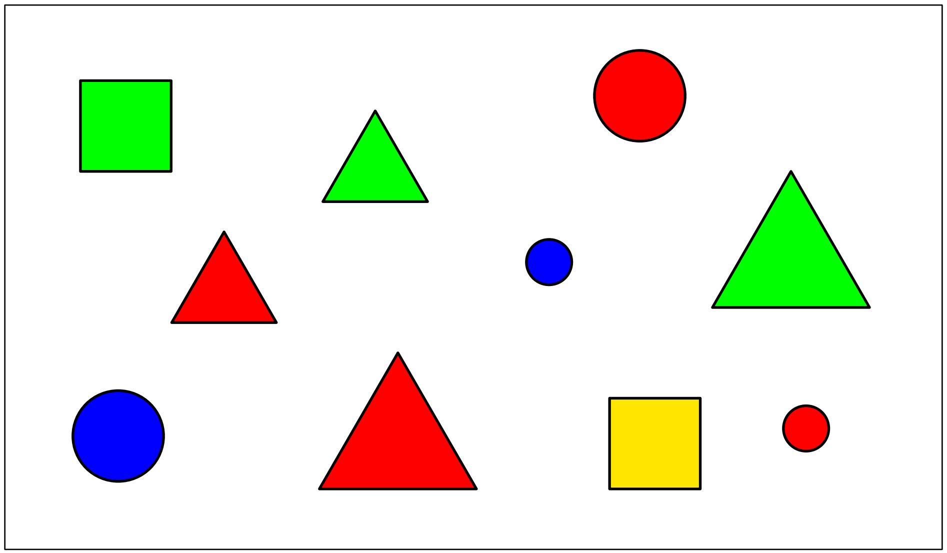
Idea to Solve the Problems

Given: A large (high-dimensional) δ representing the domain knowledge.

Desired: A set of smaller (lower-dimensional) $\{\delta_1, \ldots, \delta_s\}$ (maybe overlapping) from which the original δ *could* be reconstructed with no (or as few as possible) errors.

With such a decomposition we can draw any conclusions from $\{\delta_1, \ldots, \delta_s\}$ that could be inferred from δ — without, however, actually reconstructing it.

Example World

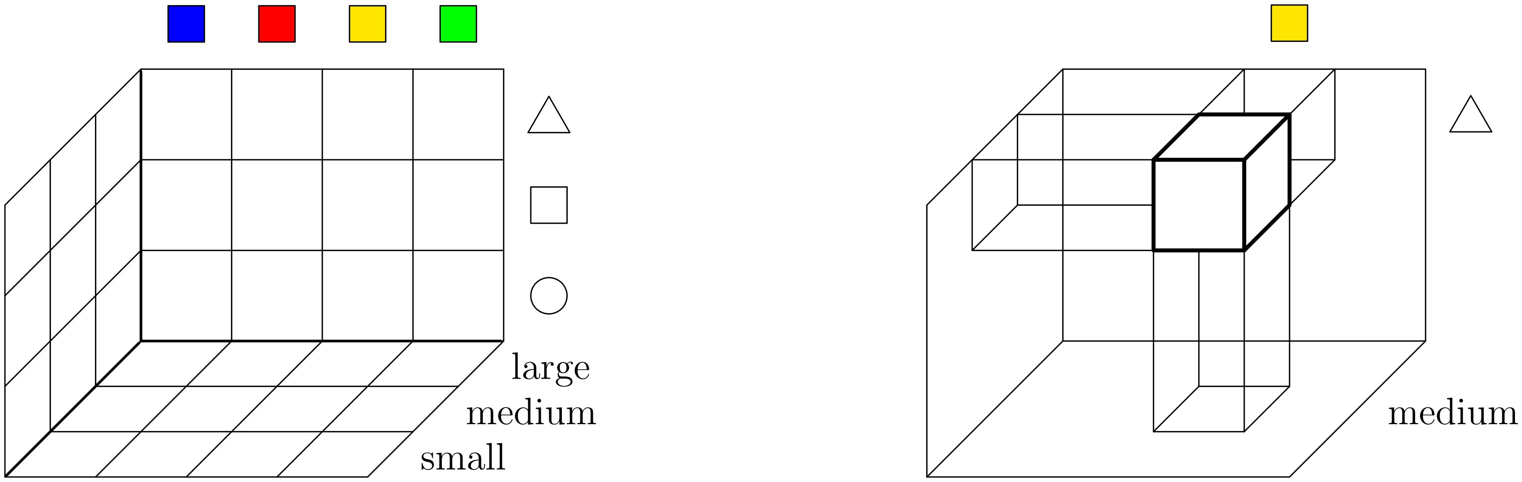


Relation

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

- 10 simple geometric 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 Ω of states.

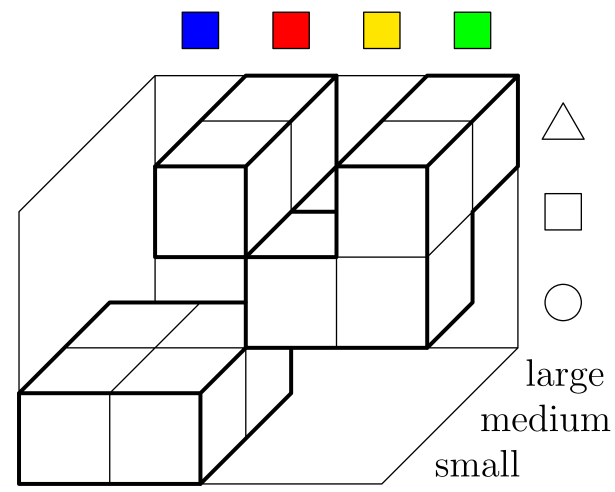
The states are described by a set of n attributes A_i, $i = 1, \ldots, n$, whose domains $\{a_1^{(i)}, \ldots, a_{n_i}^{(i)}\}$ can be seen as sets of propositions or events.

The events in a domain are mutually exclusive and exhaustive.

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.

Possibility-Based Formalization

Definition: Let Ω be a (finite) sample space.
A **discrete possibility measure** R on Ω is a function $R : 2^{\Omega} \to \{0, 1\}$ 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).

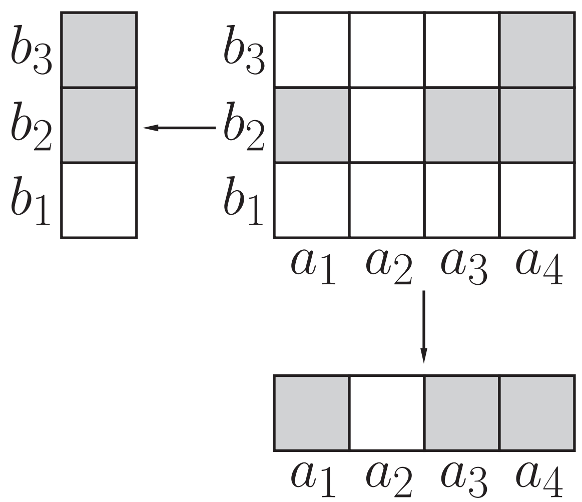
$R(A = a)$ and $R(a)$ are abbreviations of $R(\{\omega \mid A(\omega) = a\})$.

Projection / Marginalization

Let R_{AB} be a relation over two attributes A and B. The projection (or marginalization) from schema $\{A, B\}$ to schema $\{A\}$ is defined as:

$$\forall a \in \mathrm{dom}(A): \; R_A(A = a) \; = \max_{\forall b \in \mathrm{dom}(B)} \{R_{AB}(A = a, B = b)\}$$

This principle is easily generalized to sets of attributes.

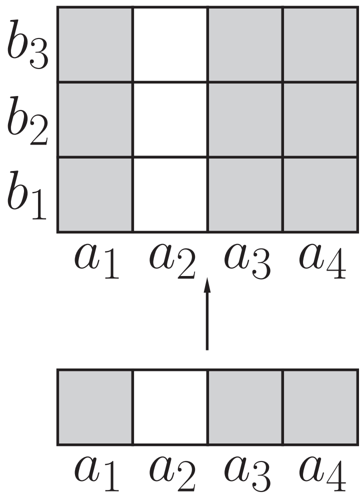


Cylindrical Extention

Let R_A be a relation over an attribute A. The cylindrical extention R_{AB} from $\{A\}$ to $\{A, B\}$ is defined as:

$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : \; R_{AB}(A = a, B = b) \; = \; R_A(A = a)$$

This principle is easily generalized to sets of attributes.

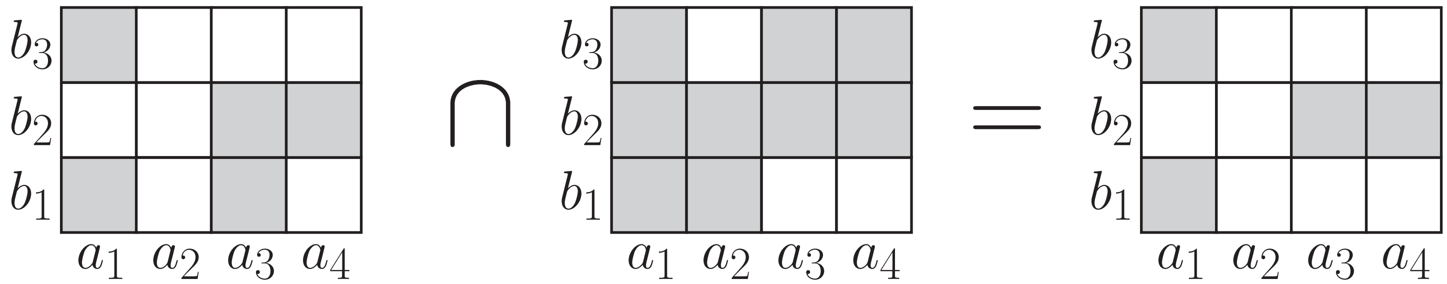


Intersection

Let $R_{AB}^{(1)}$ and $R_{AB}^{(2)}$ be two relations with attribute schema $\{A, B\}$. The intersection R_{AB} of both is defined in the natural way:

$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) :$$
$$R_{AB}(A = a, B = b) \;=\; \min\{R_{AB}^{(1)}(A = a, B = b), R_{AB}^{(2)}(A = a, B = b)\}$$

This principle is easily generalized to sets of attributes.

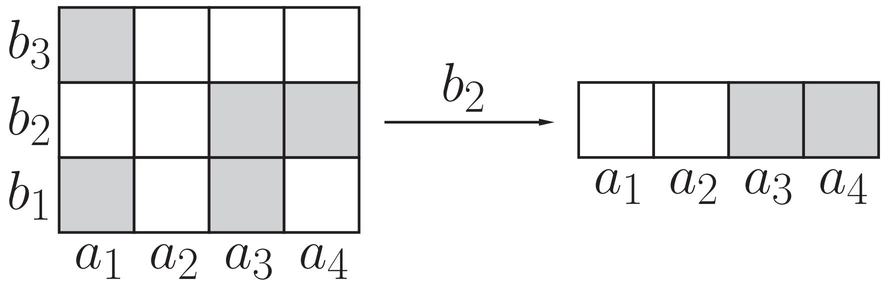


Conditional Relation

Let R_{AB} be a relation over the attribute schema $\{A, B\}$. The conditional relation of A given B is defined as follows:

$$\forall a \in \text{dom}(A) : \forall b \in \text{dom}(B) : R_A(A = a \mid B = b) \; = \; R_{AB}(A = a, B = b)$$

This principle is easily generalized to sets of attributes.

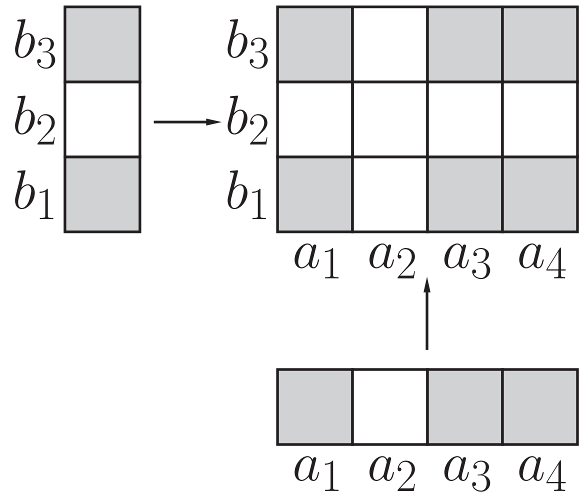


(Unconditional) Independence

Let R_{AB} be a relation over the attribute schema $\{A, B\}$. We call A and B relationally independent (w. r. t. R_{AB}) if the following condition holds:

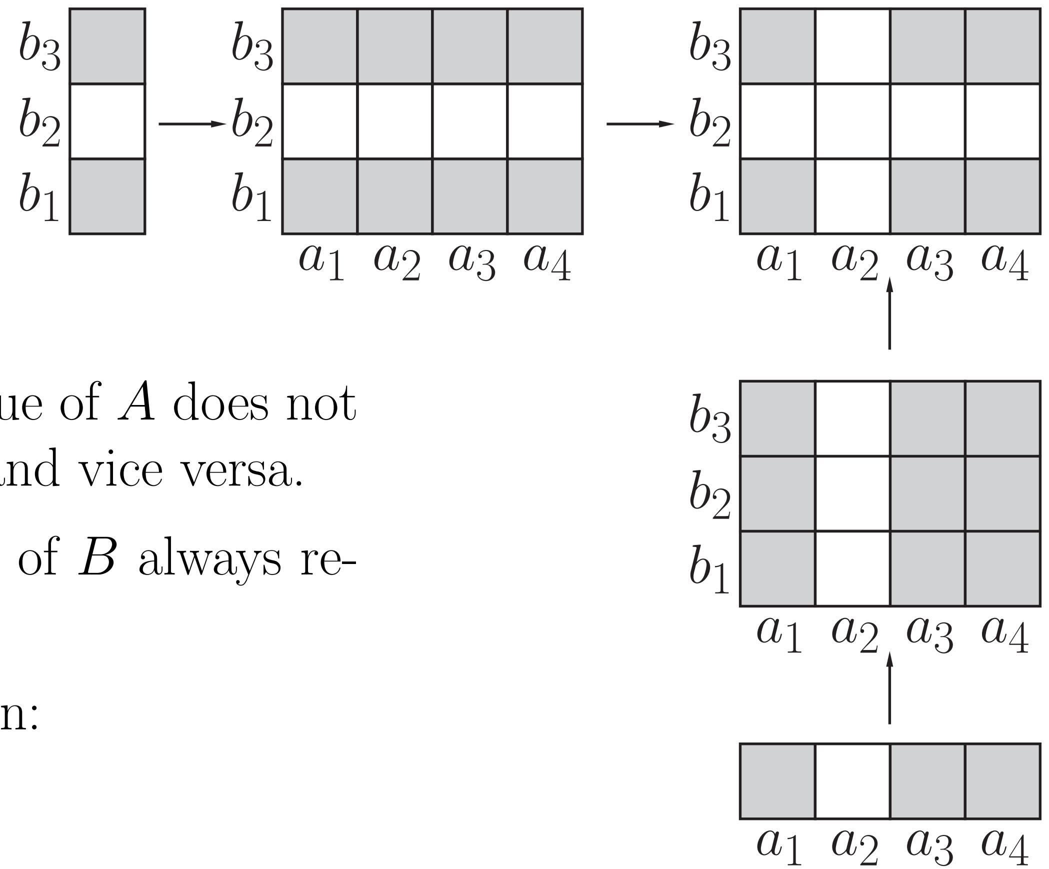
$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : R_{AB}(A = a, B = b) \;=\; \min\{R_A(A = a), R_B(B = b)\}$$

This principle is easily generalized to sets of attributes.



Object Representation

(Unconditional) Independence



Intuition: Fixing one (possible) value of A does not restrict the (possible) values of B and vice versa.

Conditioning on any possible value of B always results in the same relation R_A.

Alternative independence expression:

$$\forall b \in \mathrm{dom}(B): \;\; R_B(B = b) = 1 :$$
$$R_A(A = a \mid B = b) \;=\; R_A(A = a)$$

Decomposition

Obviously, the original two-dimensional relation can be reconstructed from the two one-dimensional ones, if we have (unconditional) independence.

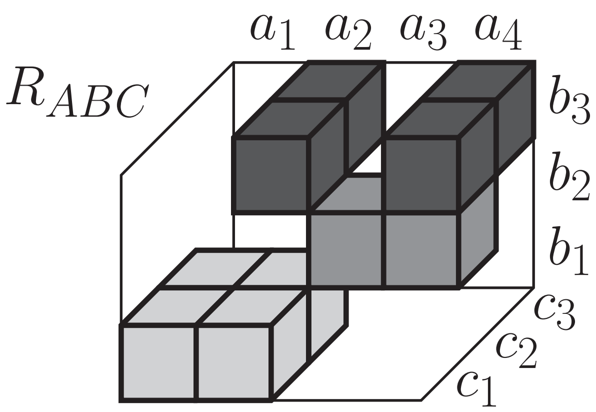
The definition for (unconditional) independence already told us how to do so:

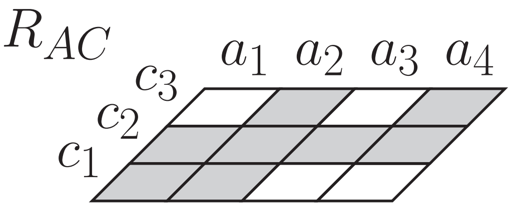
$$R_{AB}(A = a, B = b) \;=\; \min\{R_A(A = a), R_B(B = b)\}$$

Storing R_A and R_B is sufficient to represent the information of R_{AB}.

Question: The (unconditional) independence is a rather strong restriction. Are there other types of independence that allow for a decomposition as well?

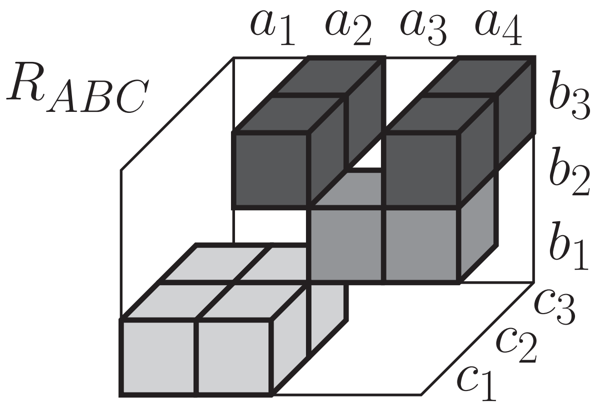
Conditional Relational Independence

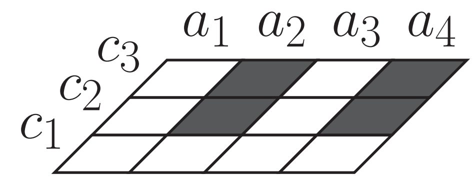




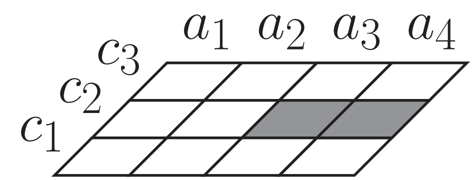
Clearly, A and C are unconditionally dependent, i. e. the relation R_{AC} cannot be reconstructed from R_A and R_C.

Conditional Relational Independence

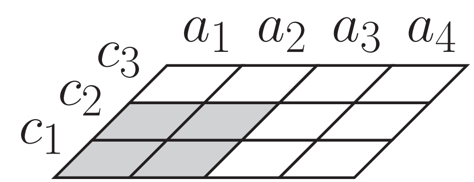




$R_{AC}(\ \cdot, \cdot \mid B = b_3)$



$R_{AC}(\ \cdot, \cdot \mid B = b_2)$



$R_{AC}(\ \cdot, \cdot \mid B = b_1)$

However, given all possible values of B, all respective conditional relations R_{AC} show the independence of A and C.

$$R_{AC}(a, c \mid b) \;=\; \min\{R_A(a \mid b), R_C(c \mid b)\}$$

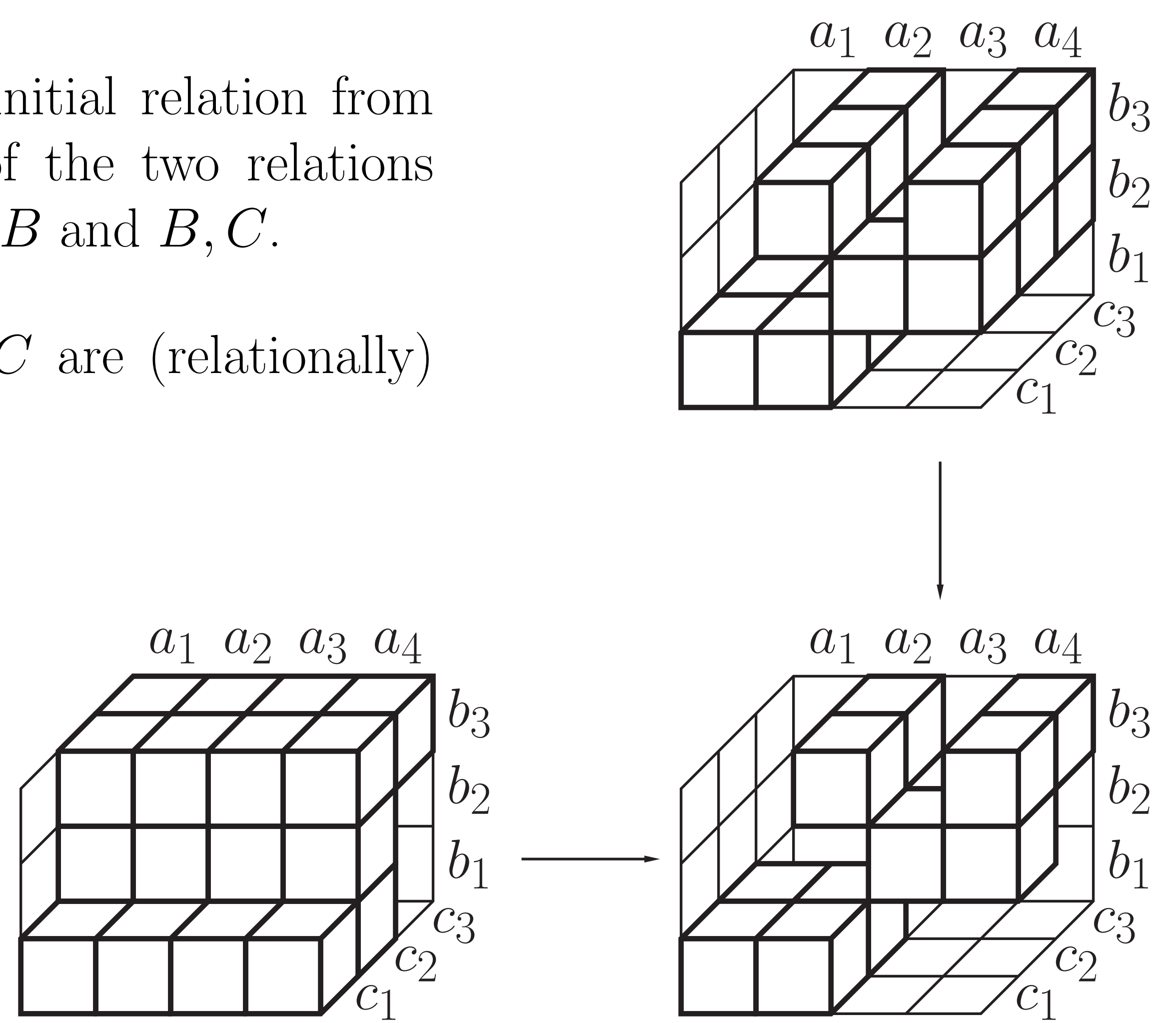
With the definition of a conditional relation, the decomposition description for R_{ABC} reads:

$$R_{ABC}(a, b, c) \;=\; \min\{R_{AB}(a, b), R_{BC}(b, c)\}$$

Conditional Relational Independence

Again, we reconstruct the initial relation from the cylindrical extentions of the two relations formed by the attributes A, B and B, C.

It is possible since A and C are (relationally) independent given B.



Possibility-Based Formalization (continued)

Definition: Let $U = \{A_1, \ldots, A_n\}$ be a set of attributes defined on a (finite) sample space Ω with respective domains $\mathrm{dom}(A_i)$, $i = 1, \ldots, n$. A **relation** r_U over U is the restriction of a discrete possibility measure R on Ω 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

$$\begin{aligned}\mathcal{E}_U &= \left\{ E \in 2^\Omega \;\middle|\; \exists a_1 \in \mathrm{dom}(A_1) : \ldots \exists a_n \in \mathrm{dom}(A_n) : E \mathrel{\widehat{=}} \bigwedge_{A_j \in U} A_j = a_j \right\} \\ &= \left\{ E \in 2^\Omega \;\middle|\; \exists a_1 \in \mathrm{dom}(A_1) : \ldots \exists a_n \in \mathrm{dom}(A_n) : \right. \\ &\qquad\qquad \left. E = \left\{ \omega \in \Omega \;\middle|\; \bigwedge_{A_j \in U} A_j(\omega) = a_j \right\} \right\}.\end{aligned}$$

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 $U = \{A_1, \ldots, A_n\}$ be a set of attributes and r_U a relation over U. Furthermore, let $\mathcal{M} = \{M_1, \ldots, M_m\} \subseteq 2^U$ 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

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

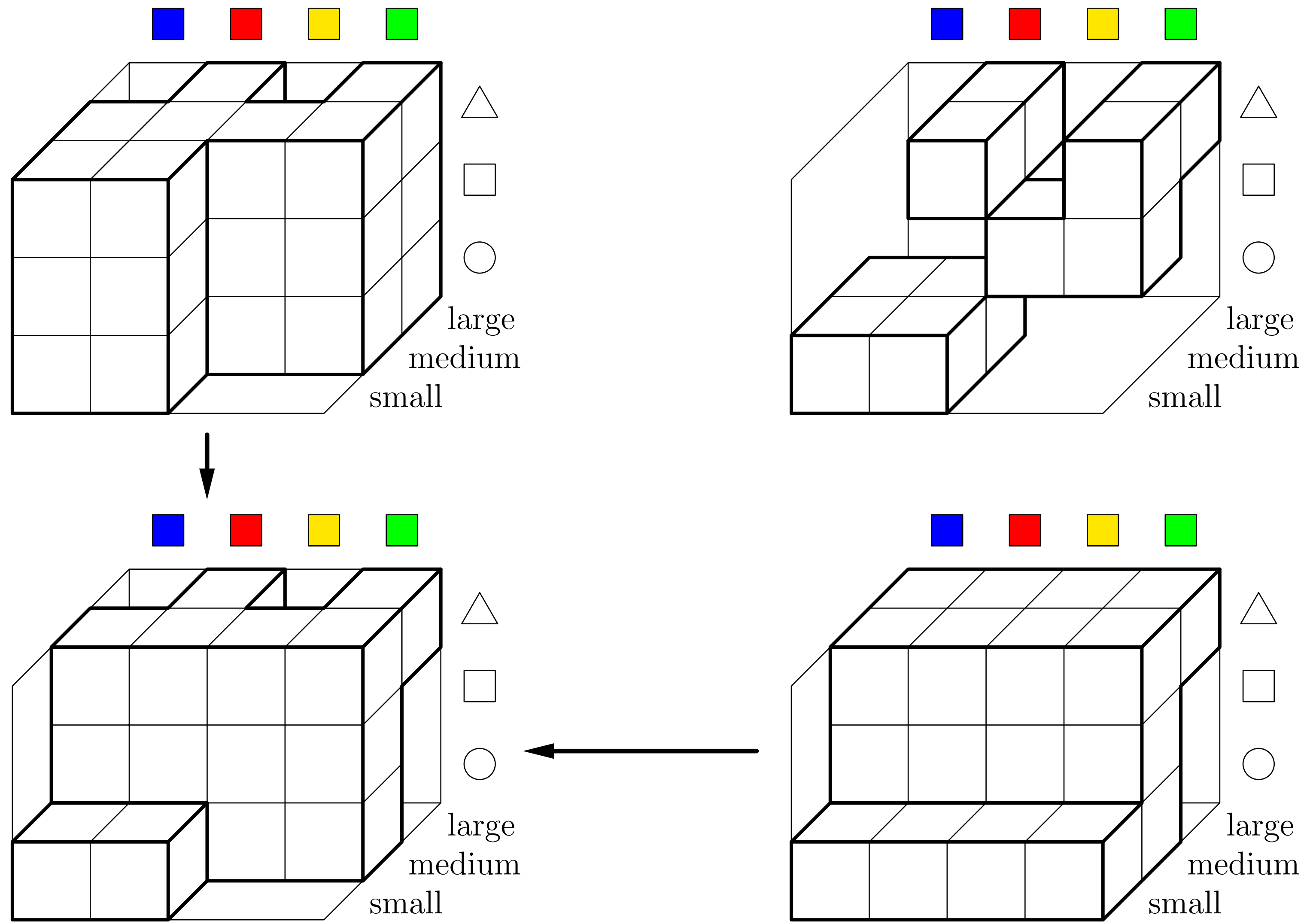
If r_U is decomposable w.r.t. $\mathcal{M}$, the set of relations

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

is called the **decomposition** of r_U.

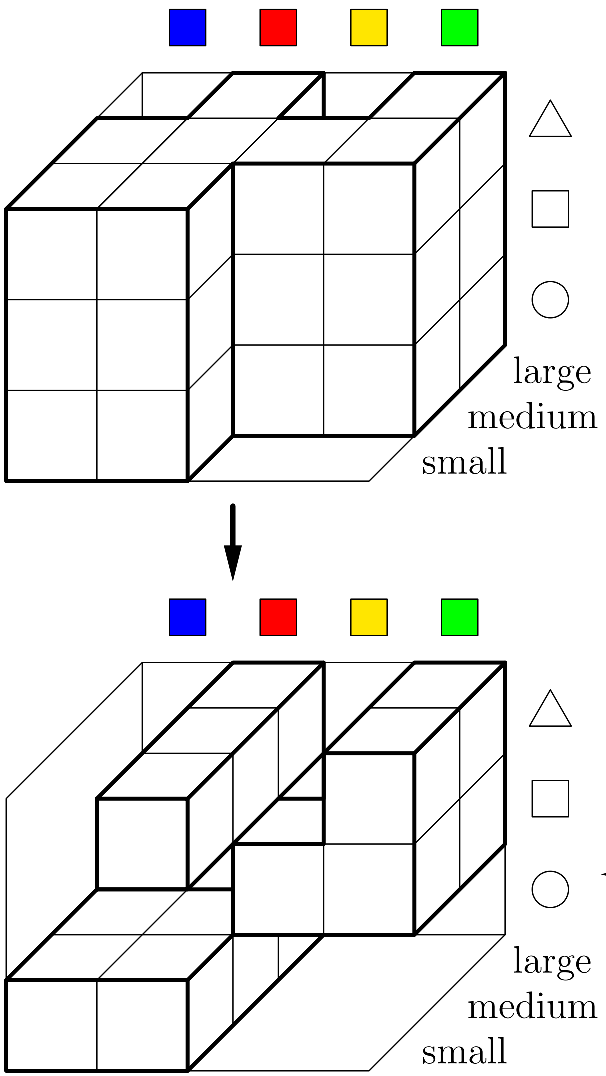
Equivalent to **join decomposability** in database theory (natural join).

Using other Projections 1



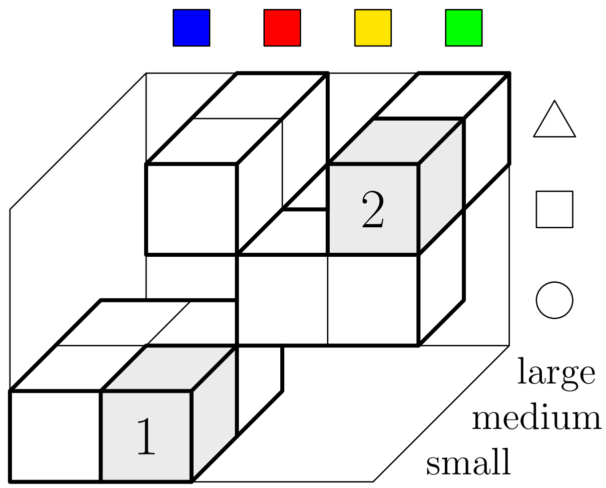
This choice of subspaces does not yield a decomposition.

Using other Projections 2



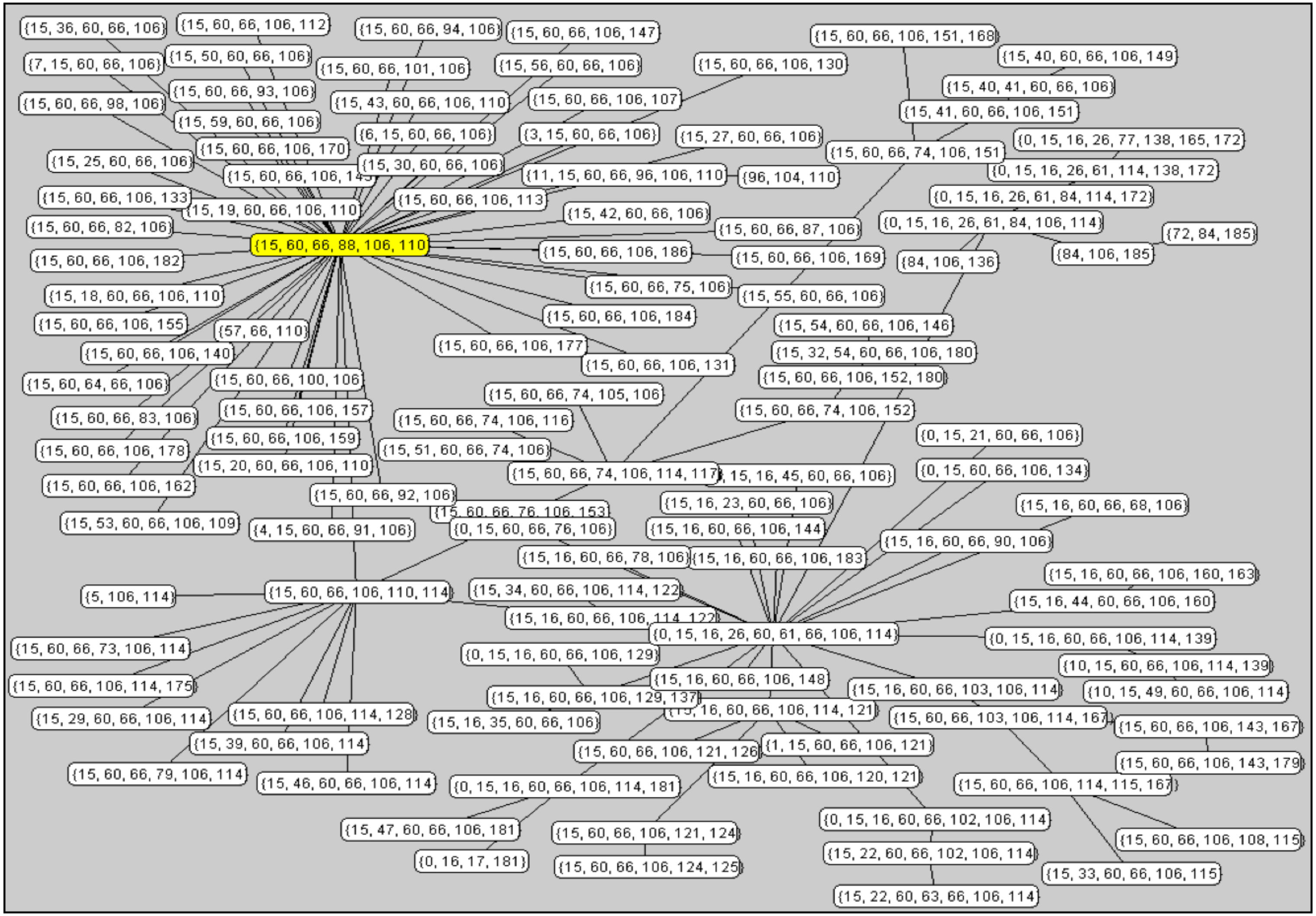
This choice of subspaces does not yield a decomposition.

Is Decomposition Always Possible?



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

Example: VW Bora

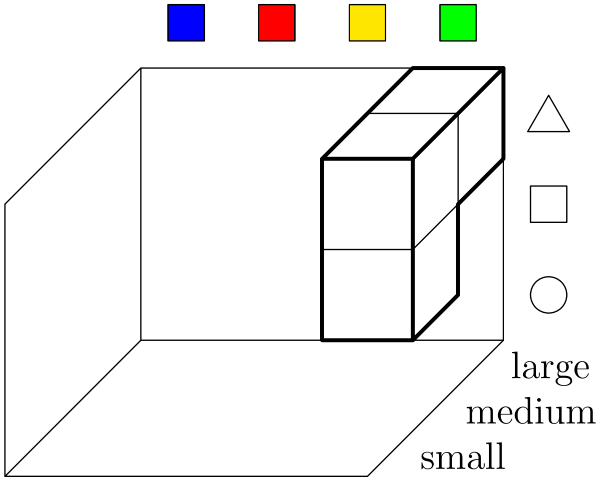


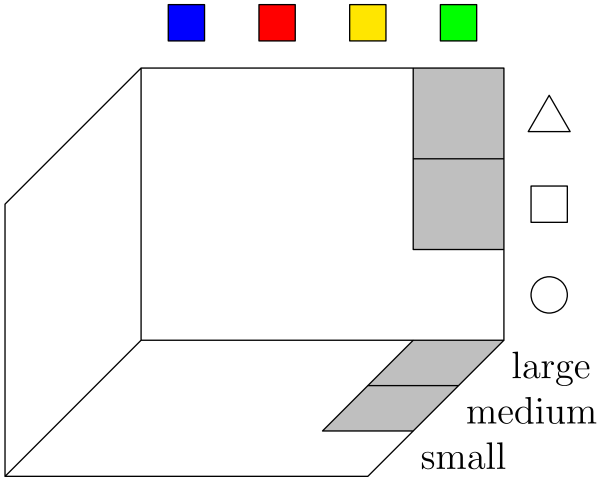
10000 rules
186 variables
174 subspaces

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

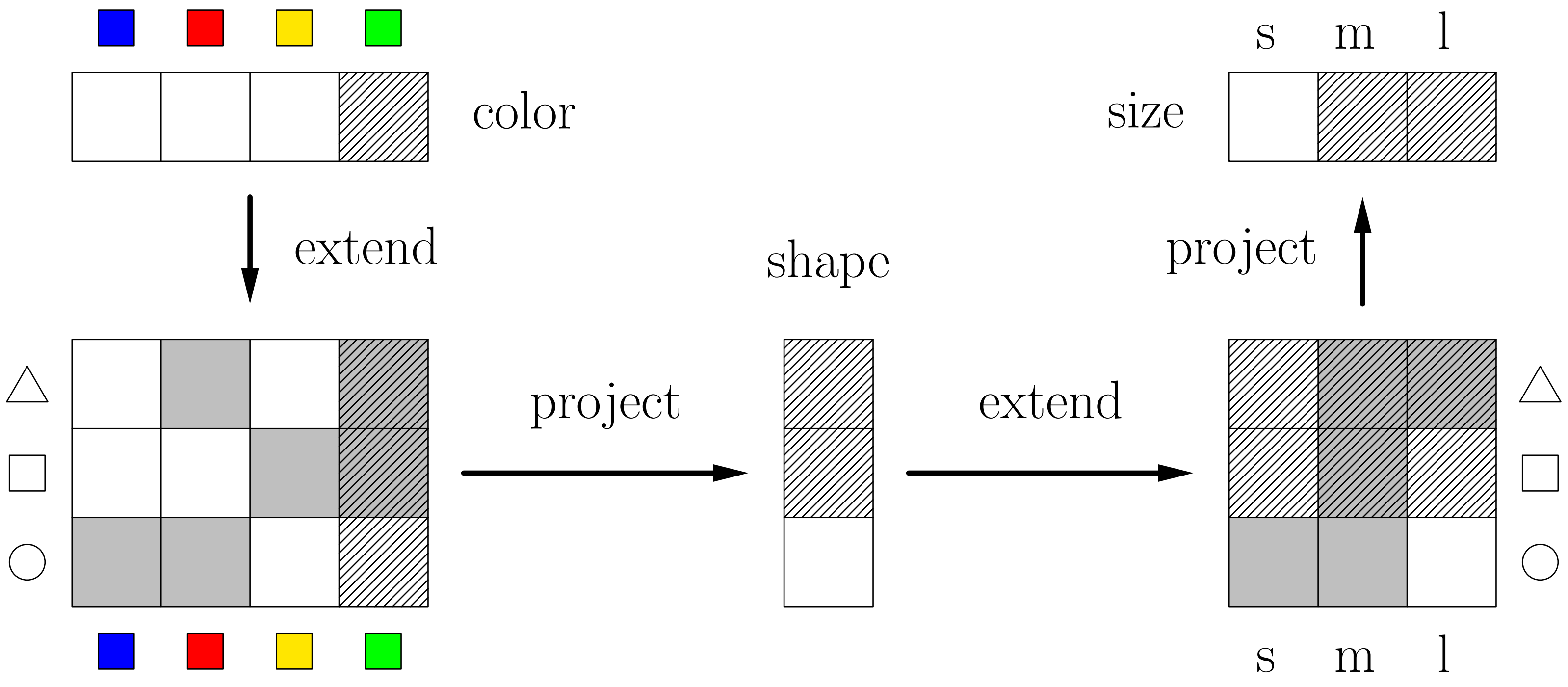
- either a triangle or a square and
- either medium or large.



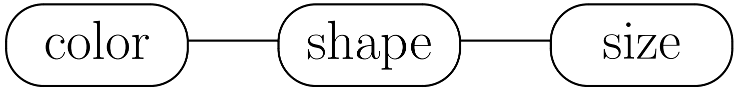


Reasoning with Projections

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

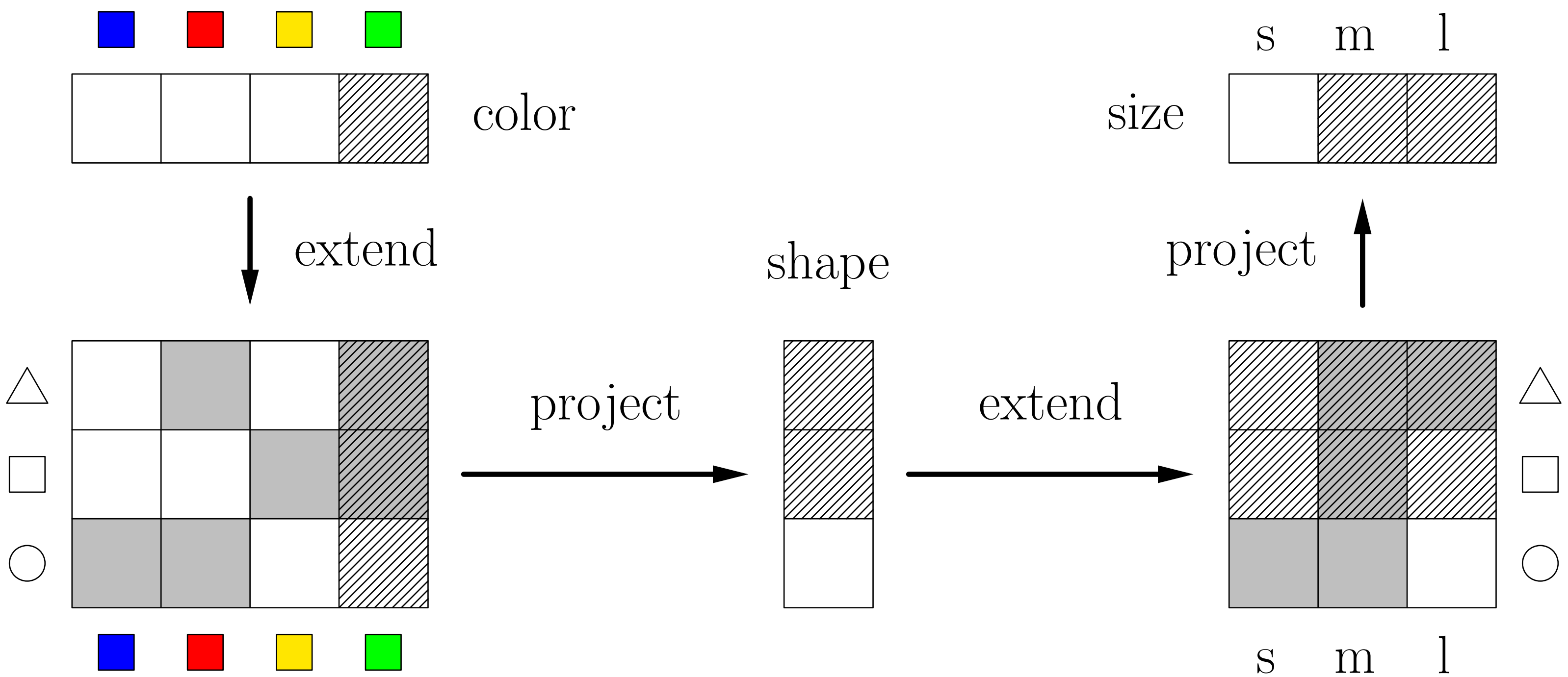


This justifies a graph representation:



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

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

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

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

and secondly

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

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

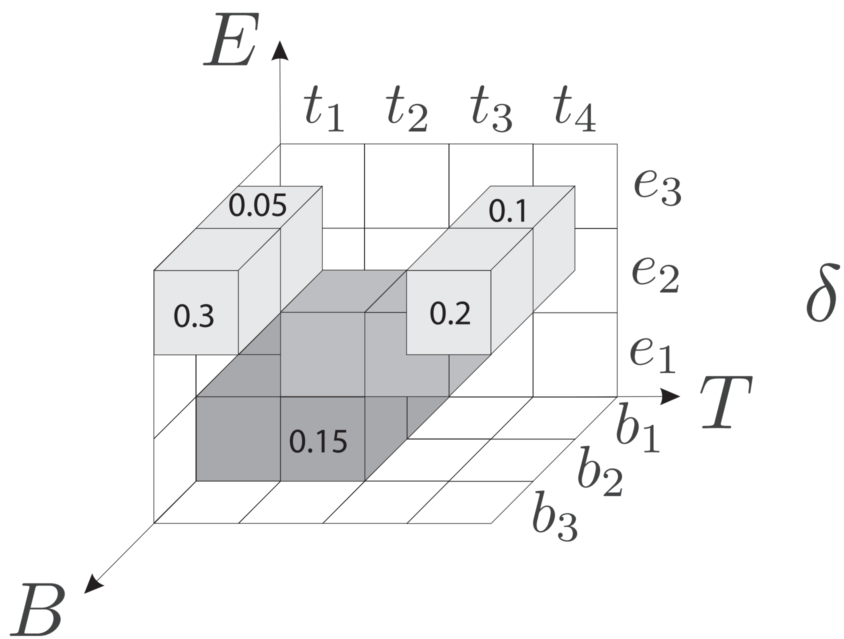
$$\begin{aligned} &R(A = a, B = b, C = c \mid A = a_{obs}) \\ &\quad = \min\{R(A = a, B = b, C = c), R(A = a \mid A = a_{\text{obs}})\}. \end{aligned}$$

Relational Evidence Propagation, Step 2

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

A:	color
B:	shape
C:	size

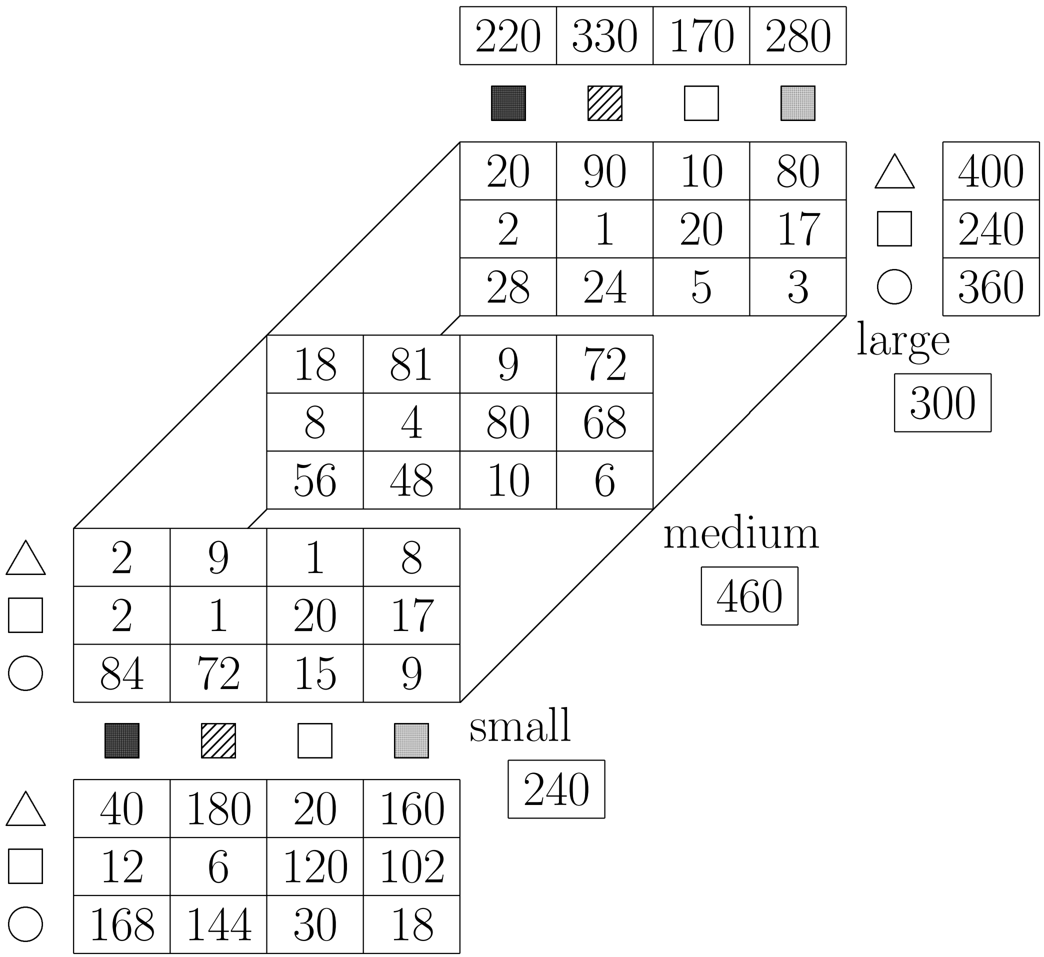
Probable car configurations



Every cube designates a value combination with its probability.

The installation rate of a value combinations is a good estimate for the probability

Extensions to Probability Distribution



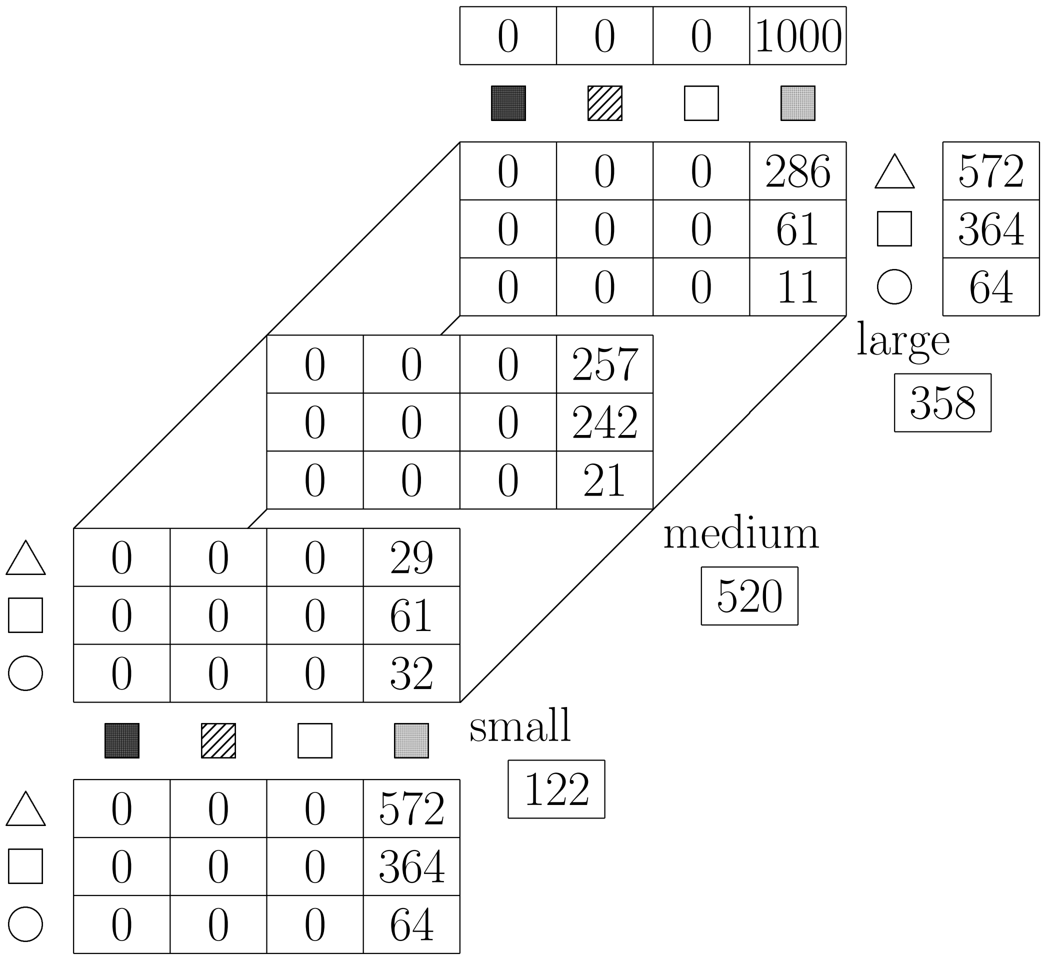
all numbers in
parts per 1000

	s	m	l
△	20	180	200
□	40	160	40
○	180	120	60

	■	▨	□	▩
large	50	115	35	100
medium	82	133	99	146
small	88	82	36	34

- The numbers state the probability of the corresponding value combination.

Reasoning: Computiong Conditional Probabilities



all numbers in parts per 1000

	s	m	l
△	29	257	286
□	61	242	61
○	32	21	11

large	0	0	0	358
medium	0	0	0	520
small	0	0	0	122

- Using the information that the given object is green.

Probabilistic Decomposition: Simple Example

- As for relational networks, the three-dimensional probability distribution can be decomposed into projections to subspaces, namely the marginal distribution on the subspace formed by color and shape and the marginal distribution on the subspace formed by shape and size.

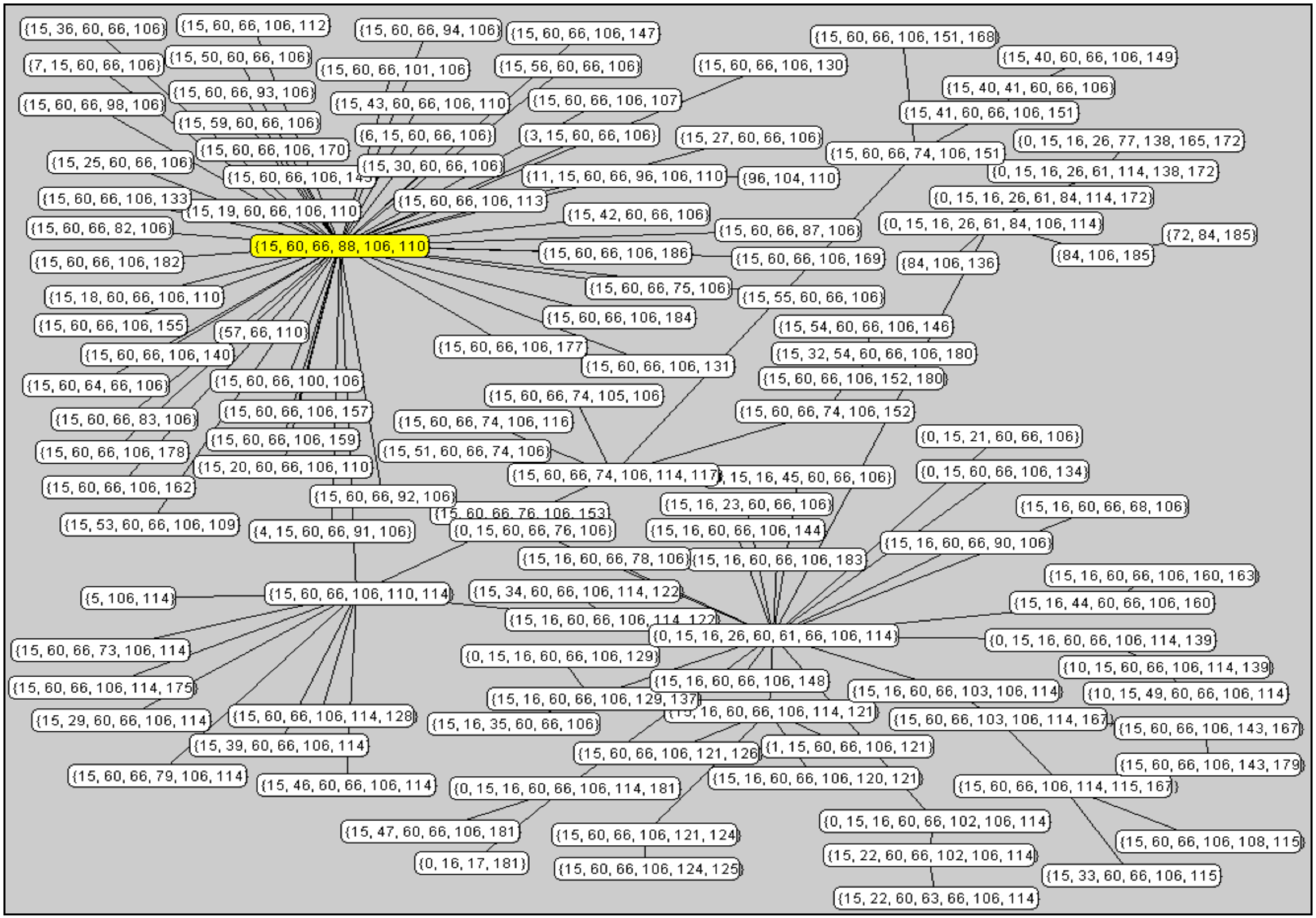
- The original probability distribution can be reconstructed from the marginal distributions using the following formulae $\forall i, j, k:$

$$\begin{aligned} P(\omega_i^{(\text{color})}, \omega_j^{(\text{shape})}, \omega_k^{(\text{size})}) &= P(\omega_i^{(\text{color})}, \omega_j^{(\text{shape})}) \cdot P(\omega_k^{(\text{size})} \mid \omega_j^{(\text{shape})}) \\ &= P(\omega_i^{(\text{color})}, \omega_j^{(\text{shape})}) \cdot \frac{P(\omega_j^{(\text{shape})}, \omega_k^{(\text{size})})}{P(\omega_j^{(\text{shape})})} \end{aligned}$$

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

$$P(\omega_k^{(\text{size})} \mid \omega_j^{(\text{shape})}) = P(\omega_k^{(\text{size})} \mid \omega_i^{(\text{color})}, \omega_j^{(\text{shape})})$$

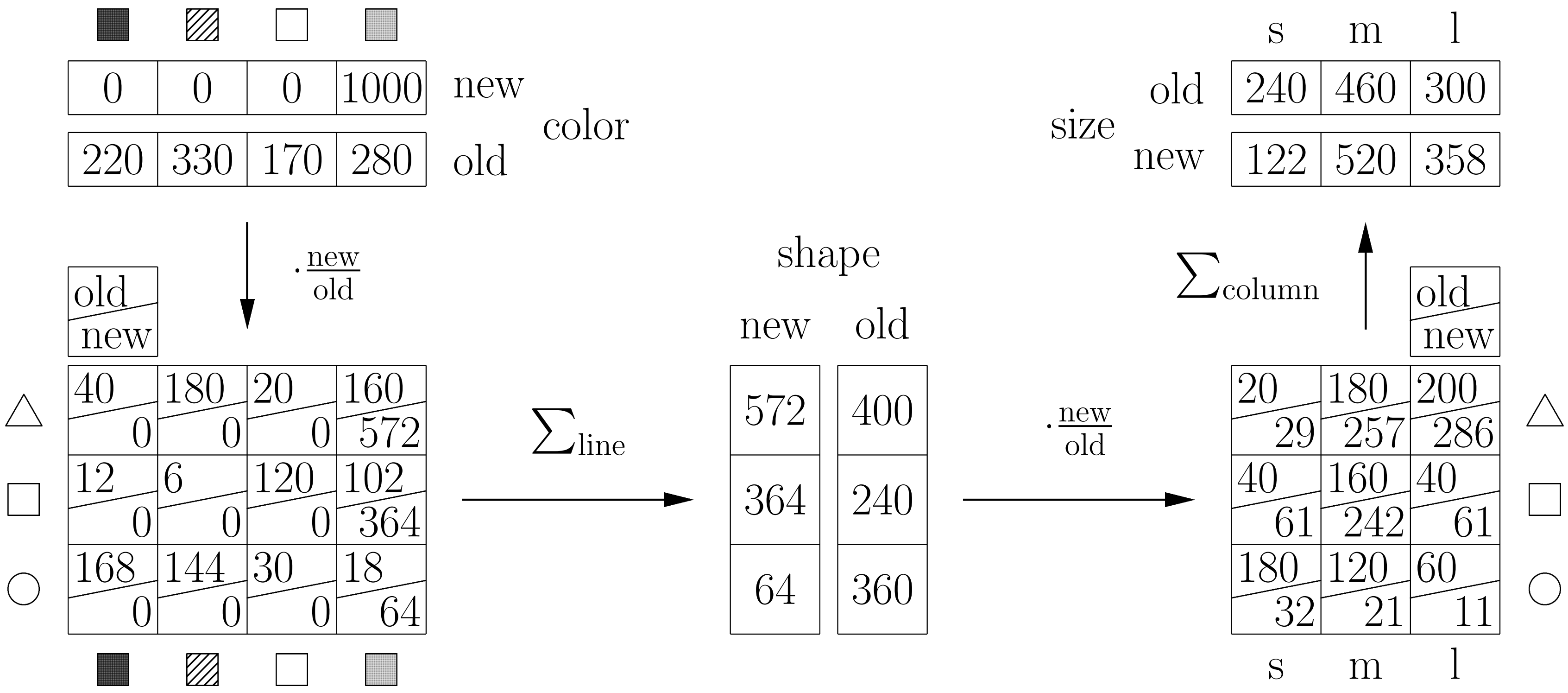
Example: VW Bora



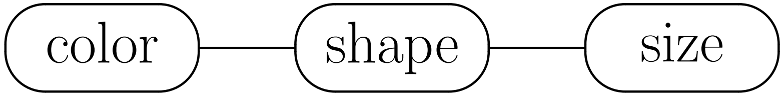
186 dim Probability space
174 Marginal Probability spaces

Reasoning with Projections

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



This justifies a network representation:



Probabilistic Decomposition

Definition: Let $U = \{A_1, \ldots, A_n\}$ be a set of attributes and p_U a probability distribution over U. Furthermore, let $\mathcal{M} = \{M_1, \ldots, M_m\} \subseteq 2^U$ 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 $\phi_M : \mathcal{E}_M \to \mathbb{R}_0^+$, $M \in \mathcal{M}$, i.e., iff

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

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

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

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 Ω be a (finite) sample space, P a probability measure on Ω, 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

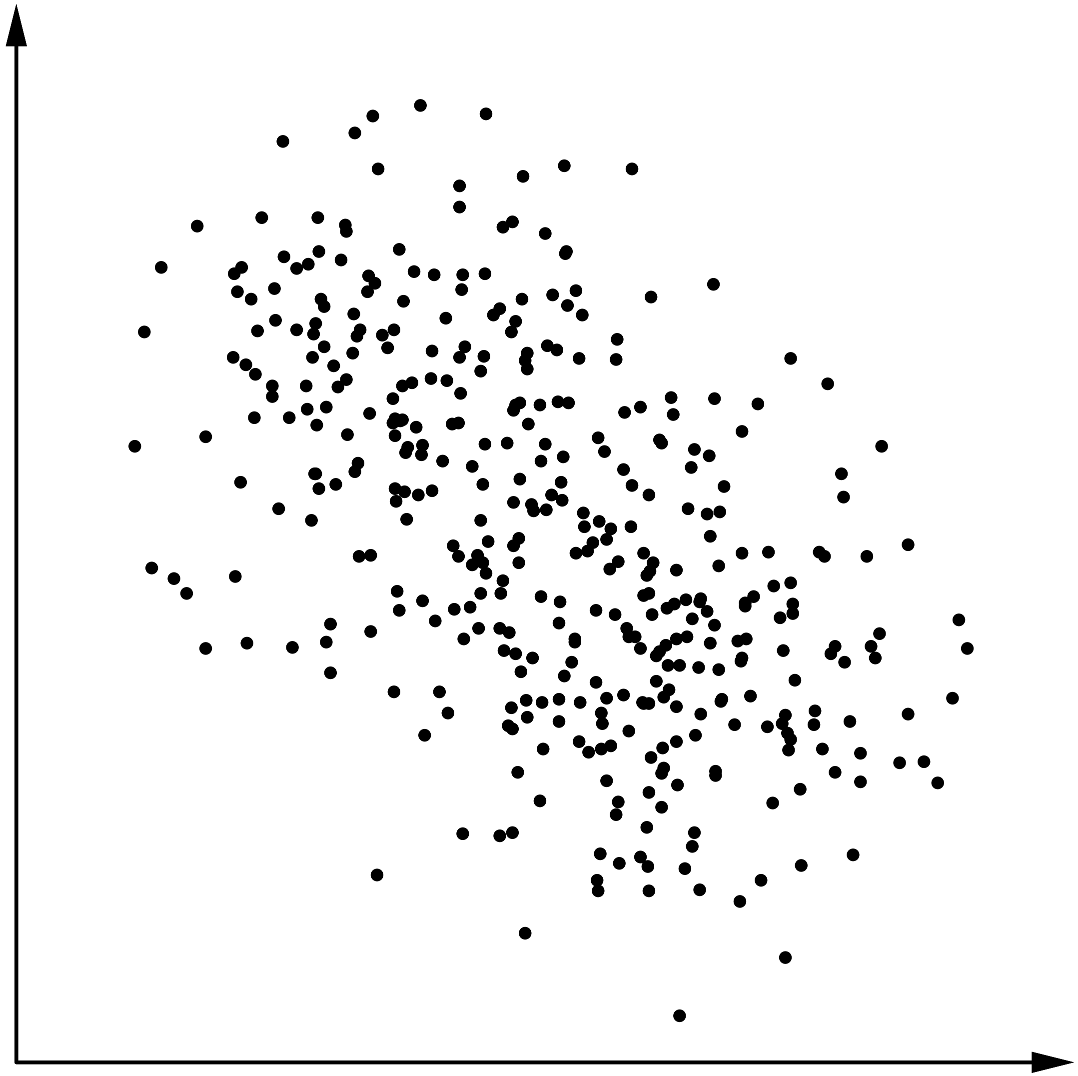
$$\forall a \in \text{dom}(A) : \forall b \in \text{dom}(B) : \forall c \in \text{dom}(C) :$$
$$P(A = a, B = b \mid C = c) = P(A = a \mid C = c) \cdot P(B = b \mid C = c)$$

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.

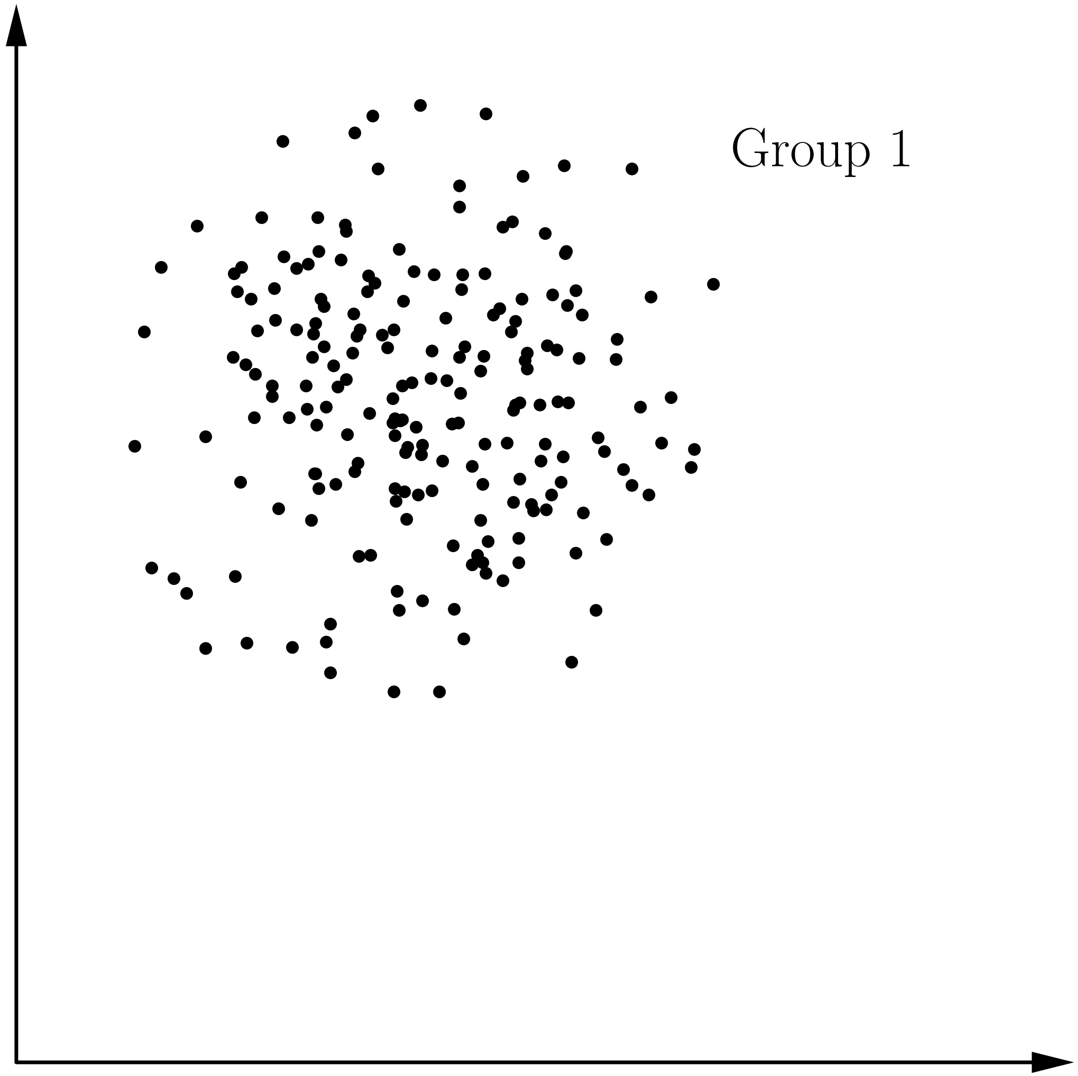
x-axis: age at death
y-axis: average number of cigarettes per day

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



Conjectured explanation:

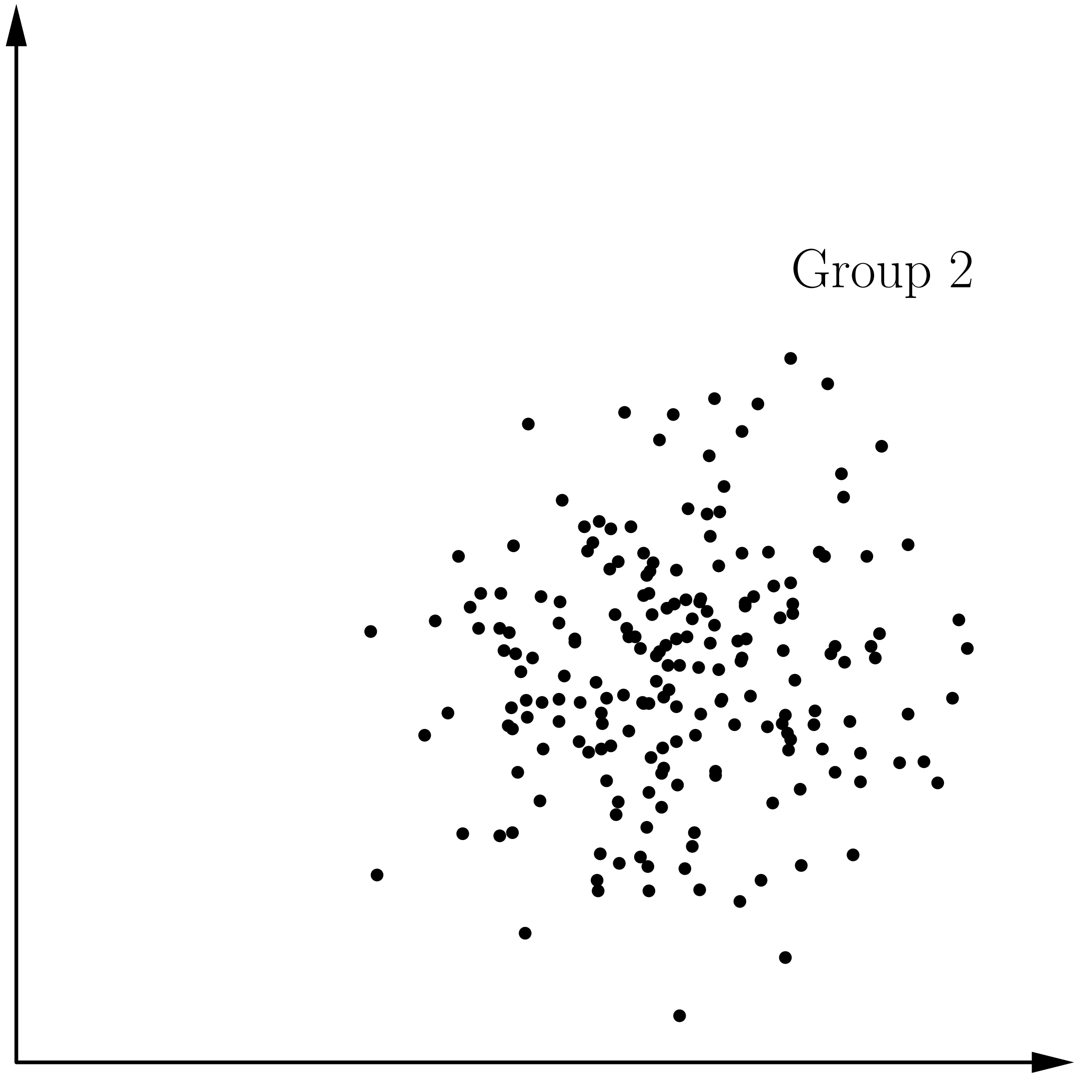
There is a common cause, namely whether the person is exposed to stress at work.

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.)

Chain Rule of Probability:

$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$P\left(\bigwedge\nolimits_{i=1}^{n} A_i = a_i\right) = \prod_{i=1}^{n} P\left(A_i = a_i \,\middle|\, \bigwedge\nolimits_{j=1}^{i-1} A_j = a_j\right)$$

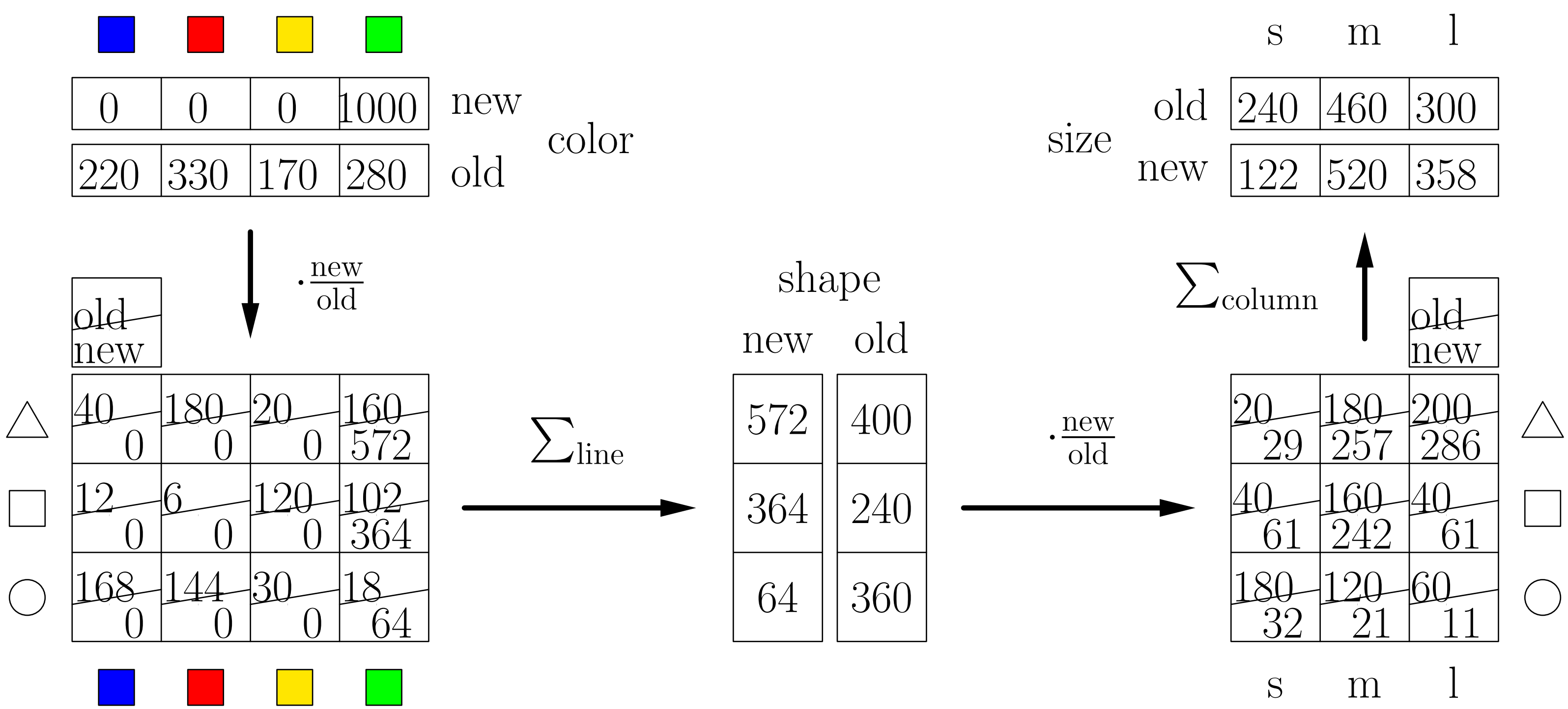
The chain rule of probability is valid in general
(or at least for strictly positive distributions).

Chain Rule Factorization:

$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$P\left(\bigwedge\nolimits_{i=1}^{n} A_i = a_i\right) = \prod_{i=1}^{n} P\left(A_i = a_i \,\middle|\, \bigwedge\nolimits_{A_j \in \mathrm{parents}(A_i)} A_j = a_j\right)$$

Conditional independence statements are used to “cancel” conditions.

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

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

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

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

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

and secondly

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

and therefore

$$P(A = a, B = b, C = c \mid A = a_{obs})$$

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

Probabilistic Evidence Propagation, Step 2

A:	color
B:	shape
C:	size

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

Objective

It is often possible to exploit local constraints (wherever they may come from — both structural and expert knowledge-based) in a way that allows for a decomposition of the large (intractable) distribution $P(X_1, \ldots, X_n)$ into several sub-structures $\{C_1, \ldots, C_m\}$ such that:

The collective size of those sub-structures is much smaller than that of the original distribution P.

The original distribution P is recomposable (with no or at least as few as possible errors) from these sub-structures in the following way:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{m} \Psi_i(c_i)$$

where c_i is an instantiation of C_i and $\Psi_i(c_i) \in \mathbb{R}^+$ a *factor potential*.

Separation Concepts

Simple Graph

A simple graph (or just: graph) is a tuple $\mathcal{G} = (V, E)$ where

$$V = \{A_1, \ldots, A_n\}$$

represents a finite set of **vertices** (or **nodes**) and

$$E \subseteq (V \times V) \setminus \{(A, A) \mid A \in V\}$$

denotes the set of **edges**.
It is called simple since there are no self-loops and no multiple edges.

Edge Types

Let $\mathcal{G} = (V, E)$ be a graph. An edge $e = (A, B)$ is called

directed if $(A, B) \in E \;\Rightarrow\; (B, A) \notin E$
Notation: $A \rightarrow B$

undirected if $(A, B) \in E \;\Rightarrow\; (B, A) \in E$
Notation: $A - B$ or $B - A$

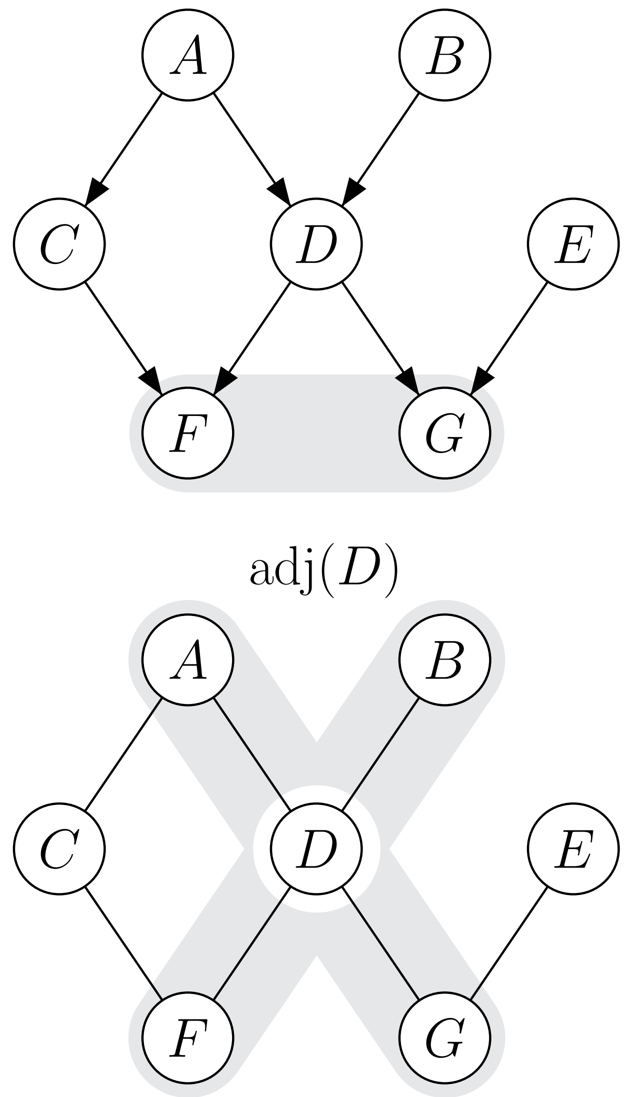
(Un)directed Graph

A graph with only (un)directed edges is called an (un)directed graph.

Adjacency Set

Let $\mathcal{G} = (V, E)$ be a graph. The set of nodes that is accessible via a given node $A \in V$ is called the **adjacency set** of A:

$$\mathrm{adj}(A) \;=\; \{B \in V \mid (A, B) \in E\}$$



Paths

Let $\mathcal{G} = (V, E)$ be a graph. A series ρ of r pairwise different nodes

$$\rho = \left\langle A_{i_1}, \ldots, A_{i_r} \right\rangle$$

is called a **path** from A_i to A_j if

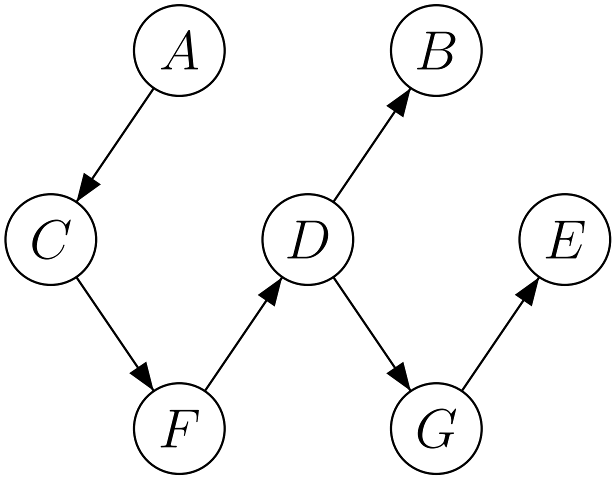
$$A_{i_1} = A_i, \quad A_{i_r} = A_j$$
$$A_{i_{k+1}} \in \operatorname{adj}(A_{i_k}), \quad 1 \leq k < r$$

A path with only undirected edges is called an **undirected path**

$$\rho = A_{i_1} - \cdots - A_{i_r}$$

whereas a path with only directed edges is referred to as a **directed path**

$$\rho = A_{i_1} \to \cdots \to A_{i_r}$$



If there is a directed path ρ from node A to node B in a directed graph $\mathcal{G}$ we write

$$A \overset{\rho}{\underset{\mathcal{G}}{\rightsquigarrow}} B.$$

If the path ρ is undirected we denote this with

$$A \overset{\rho}{\underset{\mathcal{G}}{\leftrightsquigarrow}} B.$$

Graph Types

Loop

Let $\mathcal{G} = (V, E)$ be an undirected graph. A path

$$\rho \;=\; X_1 - \cdots - X_k$$

with $X_k - X_1 \in E$ is called a loop.

Cycle

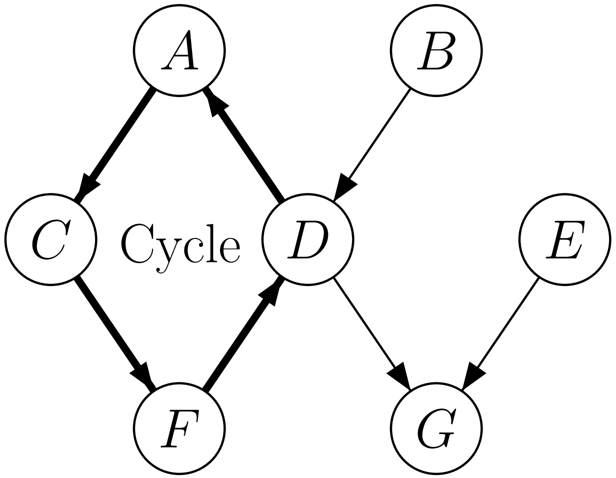
Let $\mathcal{G} = (V, E)$ be a directed graph. A path

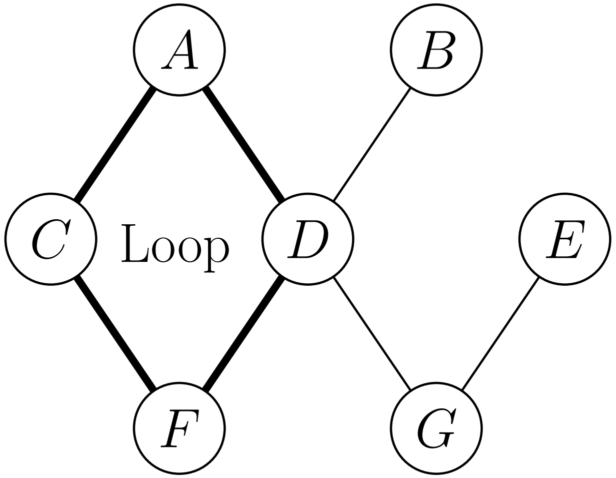
$$\rho \;=\; X_1 \to \cdots \to X_k$$

with $X_k \to X_1 \in E$ is called a cycle.

Directed Acyclic Graph (DAG)

A directed graph $\mathcal{G} = (V, E)$ is called **acyclic** if for every path $X_1 \to \cdots \to X_k$ in $\mathcal{G}$ the condition $X_k \to X_1 \notin E$ is satisfied, i. e. it contains no cycle.





Parents, Children and Families

Let $\mathcal{G} = (V, E)$ be a directed graph. For every node $A \in V$ we define the following sets:

Parents:

$$\text{parents}_{\mathcal{G}}(A) = \{B \in V \mid B \to A \in E\}$$

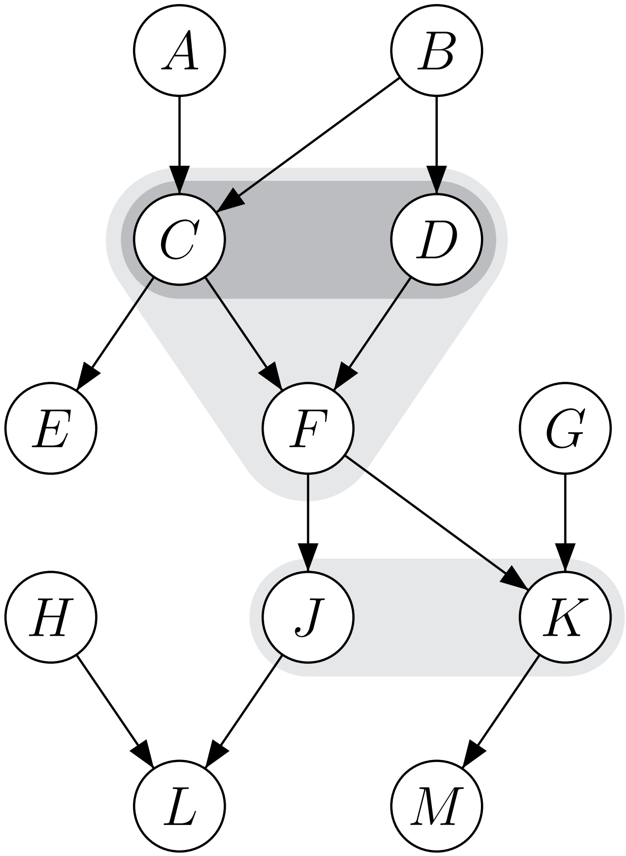
Children:

$$\text{children}_{\mathcal{G}}(A) = \{B \in V \mid A \to B \in E\}$$

Family:

$$\text{family}_{\mathcal{G}}(A) = \{A\} \cup \text{parents}_{\mathcal{G}}(A)$$

If the respective graph is clear from the context, the index $\mathcal{G}$ is omitted.



$$\begin{aligned}\text{parents}(F) &= \{C, D\}\\ \text{children}(F) &= \{J, K\}\\ \text{family}(F) &= \{C, D, F\}\end{aligned}$$

Ancestors, Descendants, Non-Descendants

Let $\mathcal{G} = (V, E)$ be a DAG. For every node $A \in V$ we define the following sets:

Ancestors:

$$\mathrm{ancs}_{\mathcal{G}}(A) = \{B \in V \mid \exists \rho : B \overset{\rho}{\underset{\mathcal{G}}{\rightsquigarrow}} A\}$$

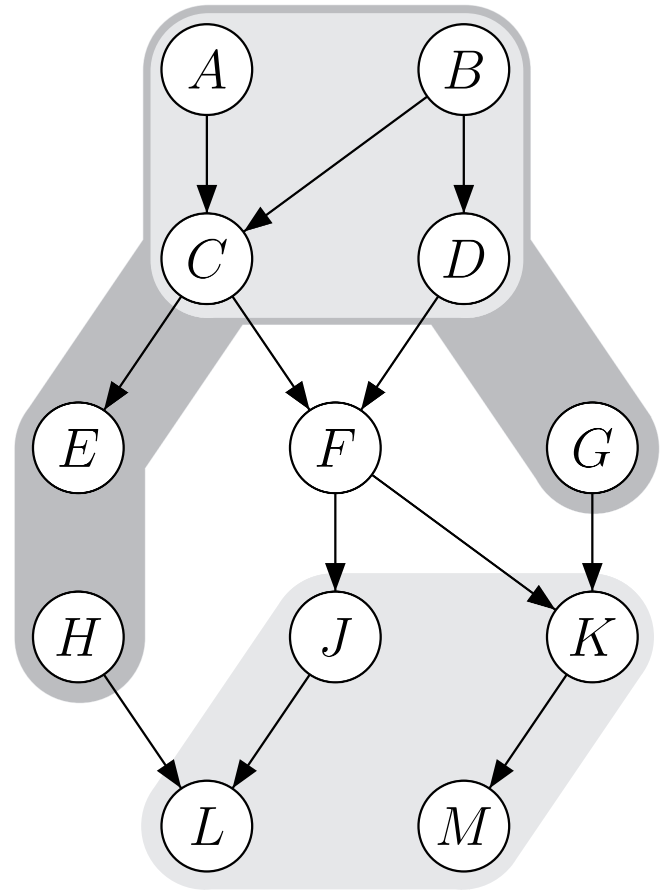
Descendants:

$$\mathrm{descs}_{\mathcal{G}}(A) = \{B \in V \mid \exists \rho : A \overset{\rho}{\underset{\mathcal{G}}{\rightsquigarrow}} B\}$$

Non-Descendants:

$$\text{non-descs}_{\mathcal{G}}(A) = V \setminus \{A\} \setminus \mathrm{descs}_{\mathcal{G}}(A)$$

If the respective graph is clear from the context, the index $\mathcal{G}$ is omitted.



$$\mathrm{ancs}(F) = \{A, B, C, D\}$$
$$\mathrm{descs}(F) = \{J, K, L, M\}$$
$$\text{non-descs}(F) = \{A, B, C, D, E, G, H\}$$

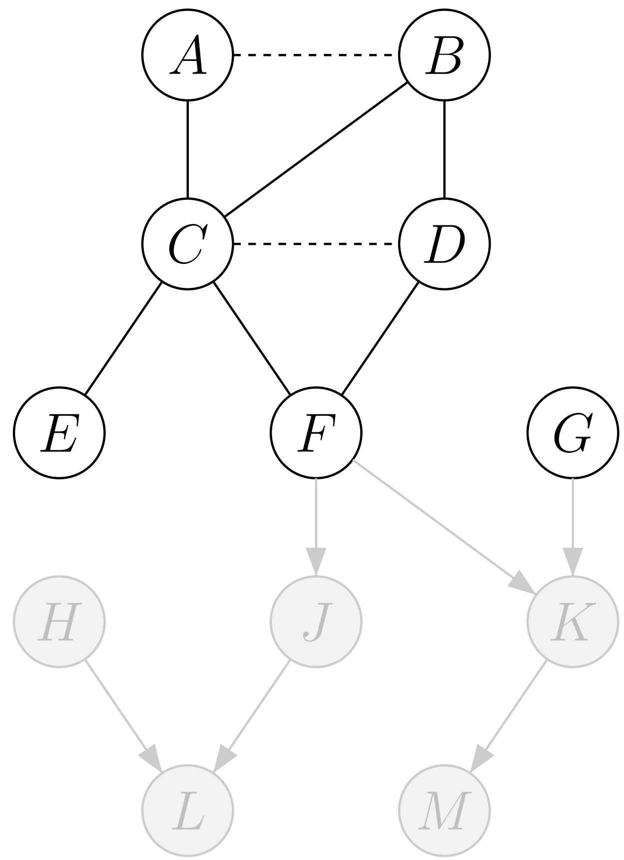
Operations on Graphs

Let $\mathcal{G} = (V, E)$ be a DAG.

The **Minimal Ancestral Subgraph** of $\mathcal{G}$ given a set $M \subseteq V$ of nodes is the smallest subgraph that contains all ancestors of all nodes in M.

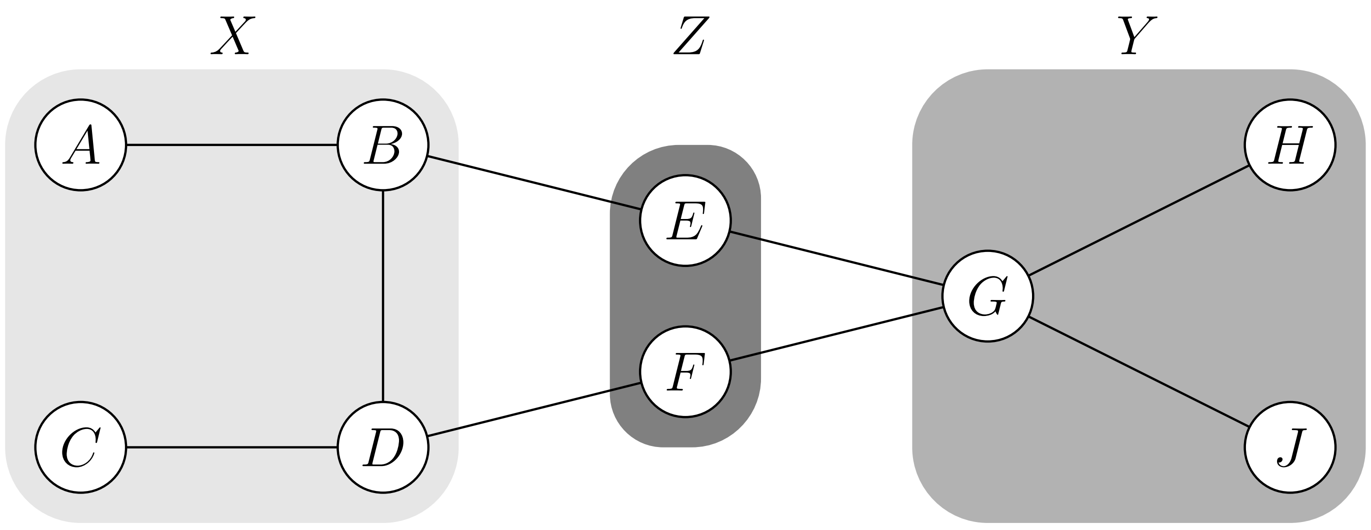
The **Moral Graph** of $\mathcal{G}$ is the undirected graph that is obtained by

1. connecting nodes that share a common child with an arbitrarily directed edge and,
2. converting all directed edges into undirected ones by dropping the arrow heads.



Moral graph of ancestral graph induced by the set $\{E, F, G\}$.

u-Separation



Let $\mathcal{G} = (V, E)$ be an undirected graph and $X, Y, Z \subseteq V$ three disjoint subsets of nodes. We agree on the following separation criteria:

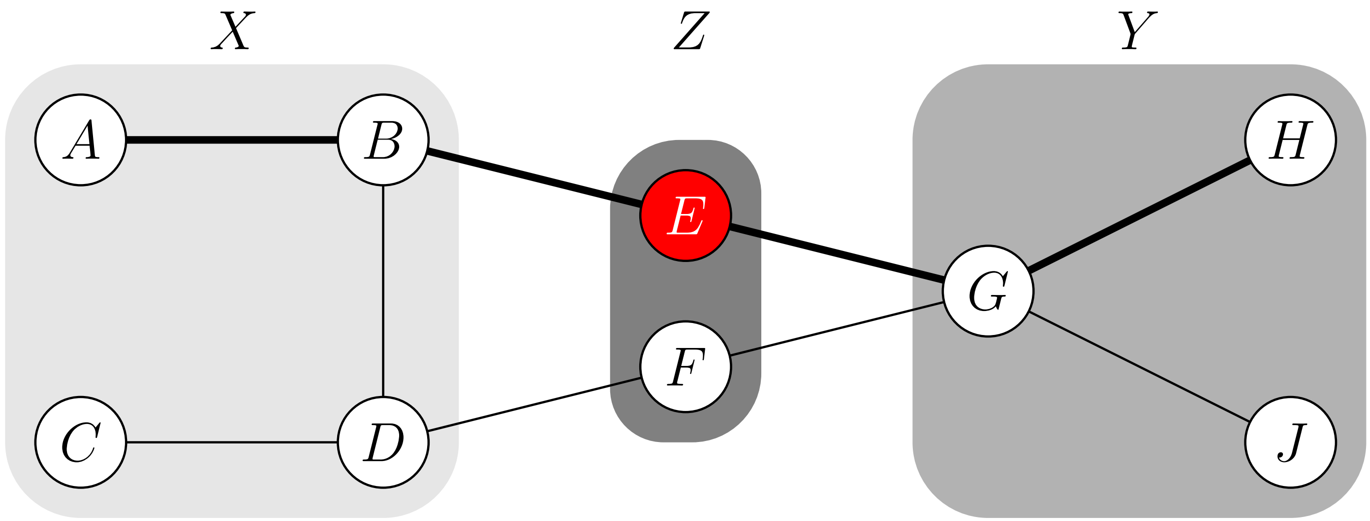
1. Z u-separates X from Y — written as

$$X \perp\!\!\!\perp_{\mathcal{G}} Y \mid Z,$$

   if every possible path from a node in X to a node in Y is blocked.

2. A path is blocked if it contains one (or more) **blocking nodes**.
3. A node is a blocking node if it lies in Z.

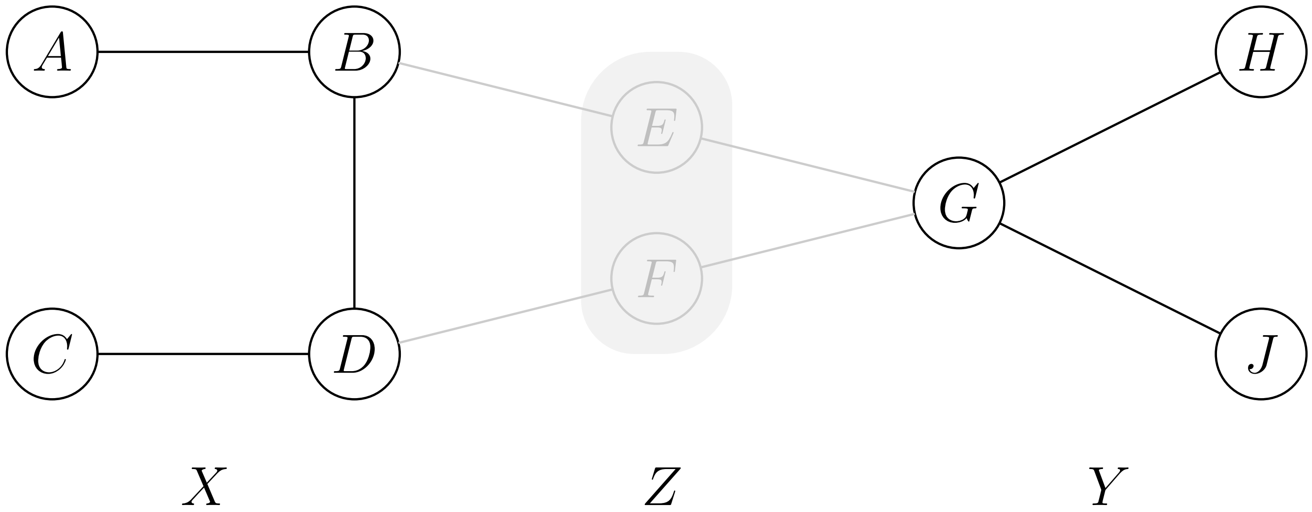
u-Separation



E. g. path $A - B - E - G - H$ is blocked by $E \in Z$. It can be easily verified, that every path from X to Y is blocked by Z. Hence we have:

$$\{A, B, C, D\} \perp\!\!\!\perp_{\mathcal{G}} \{G, H, J\} \mid \{E, F\}$$

u-Separation



Another way to check for u-separation: Remove the nodes in Z from the graph (and all the edges adjacent to these nodes). X and Y are u-separated by Z if the remaining graph is disconnected with X and Y in separate subgraphs.

Node E separates K and B in the directed graph

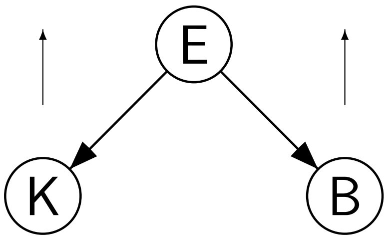
Example — Qualitative Aspects

Lecture theatre in winter: Waiting for Mr. K and Mr. B.
Not clear whether there is ice on the roads.

3 variables:

- E road condition: $\mathrm{dom}(\mathsf{E}) = \{\mathsf{ice}, \neg\mathsf{ice}\}$
- K K had an accident: $\mathrm{dom}(\mathsf{K}) = \{\mathsf{yes}, \mathsf{no}\}$
- B B had an accident: $\mathrm{dom}(\mathsf{B}) = \{\mathsf{yes}, \mathsf{no}\}$

Ignorance about these states is modelled via the observer's belief.

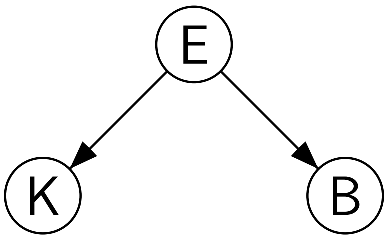


$\downarrow$ E influences K and B (the more ice the more accidents)

$\uparrow$ Knowledge about accident increases belief in ice

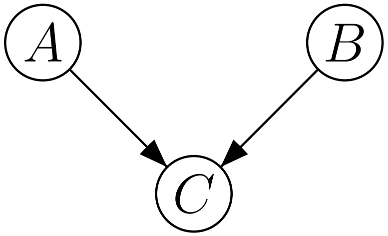
Example

A priori knowledge	Evidence	Inferences
E unknown	B has accident	$\Rightarrow$ E $=$ ice more likely
		$\Rightarrow$ K has accident more likely
E $= \neg$ice	B has accident	$\Rightarrow$ no change in belief about E
		$\Rightarrow$ no change in belief about accident of K
E unknown		K and B dependent
E known		K and B independent



Node E separates K and B in the directed graph.

Example



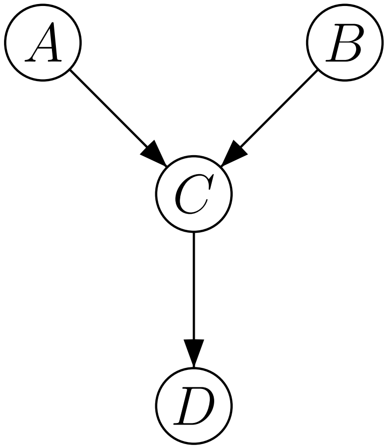
Meal quality	
A	quality of ingredients
B	cook's skill
C	meal quality

If C is not known, A and B are independent.

If C is known, then A and B become (conditionally) dependent given C.

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

Example (cont.)



Meal quality	
A	quality of ingredients
B	cook's skill
C	meal quality
D	restaurant success

If nothing is known about the restaurant success or meal quality or both, the cook's skills and quality of the ingredients are unrelated, that is, *independent*.

However, if we observe that the restaurant has no success, we can infer that the meal quality might be bad.

If we further learn that the ingredients quality is high, we will conclude that the cook's skills must be low, thus rendering both variables *dependent*.

$A \not\!\perp\!\!\!\perp B \mid D$

Now: Separation criterion for directed graphs.

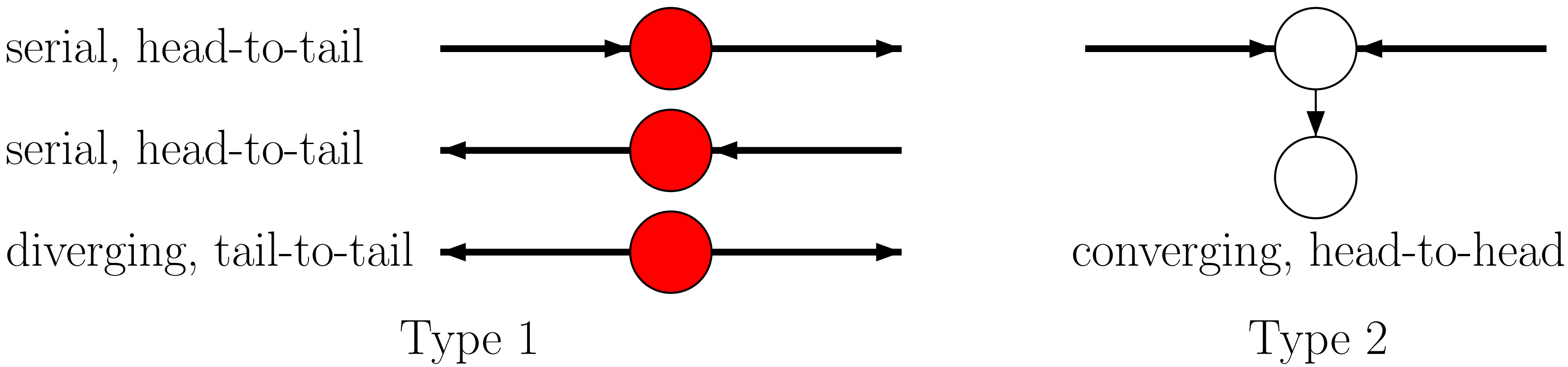
We use the same principles as for u-separation. Two modifications are necessary:

- Directed paths may lead also in reverse to the arrows.
- The blocking node condition is more sophisticated.

Blocking Node (in a directed path)

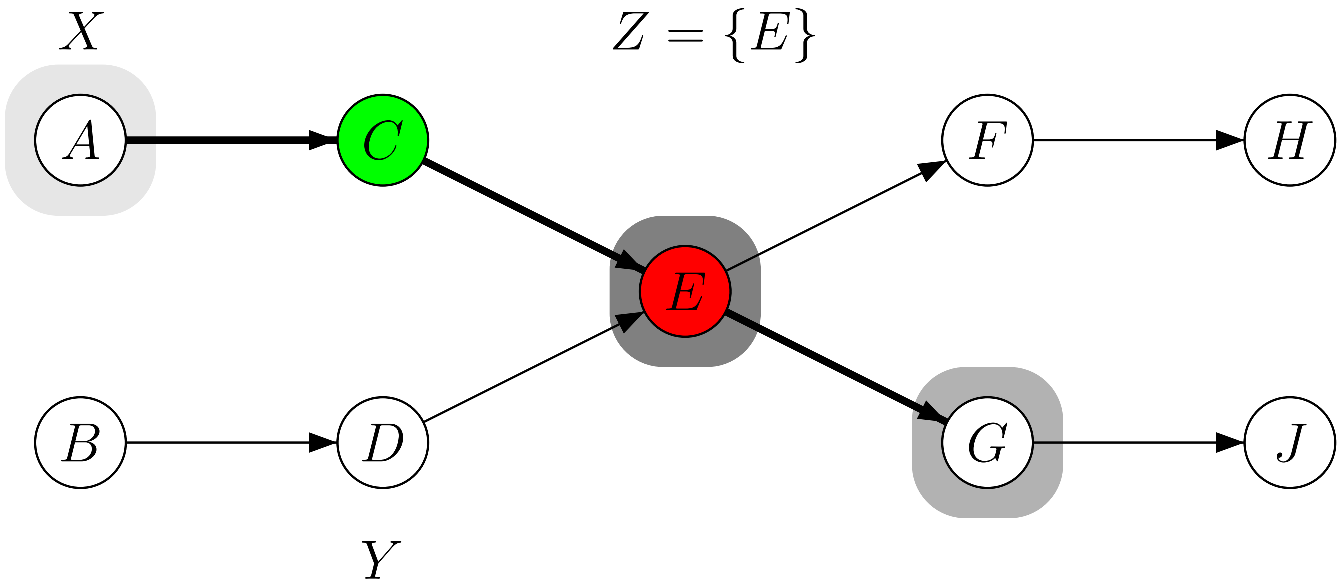
A node A is blocking if its edge directions **along the path**

- are of type 1 and $A \in Z$, or
- are of type 2 and neither A nor one of its descendants is in Z.



d-Separation

Checking path $A \to C \to E \to G$



Checking path $A \to C \to E \leftarrow D$:

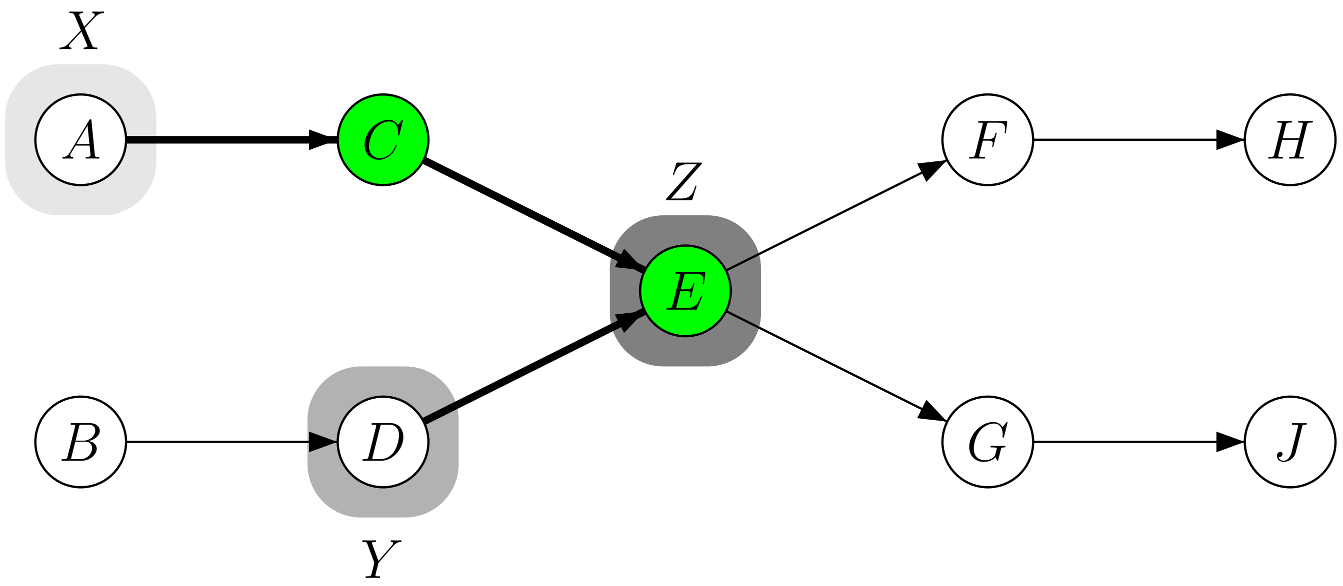
C is **serial** and not in Z: non-blocking

E is also **serial** but in Z: **blocking**

Path is blocked, no other paths between A and G are available

$$\Rightarrow A \perp\!\!\!\perp G \mid E$$

d-Separation



Checking path $A \rightarrow C \rightarrow E \leftarrow D$:

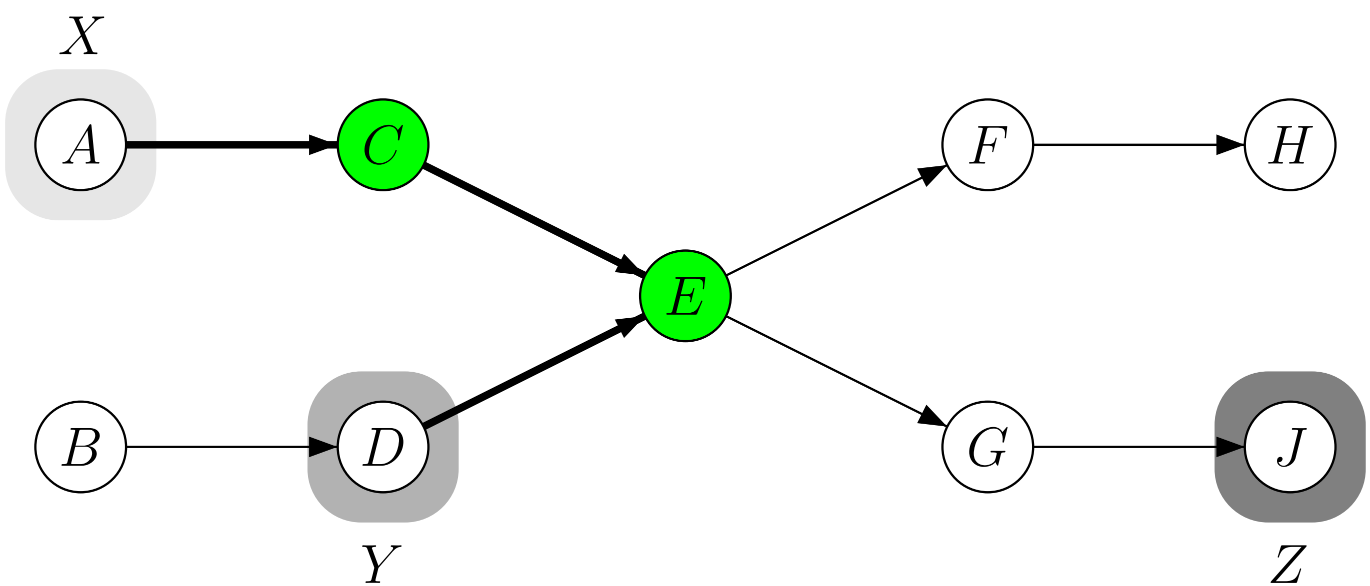
C is **serial** and not in Z: non-blocking

E is **converging** and in Z: non-blocking

$\Rightarrow$ Path is not blocked

$$A \not\perp\!\!\!\perp D \mid E$$

d-Separation



Checking path $A \to C \to E \leftarrow D$:

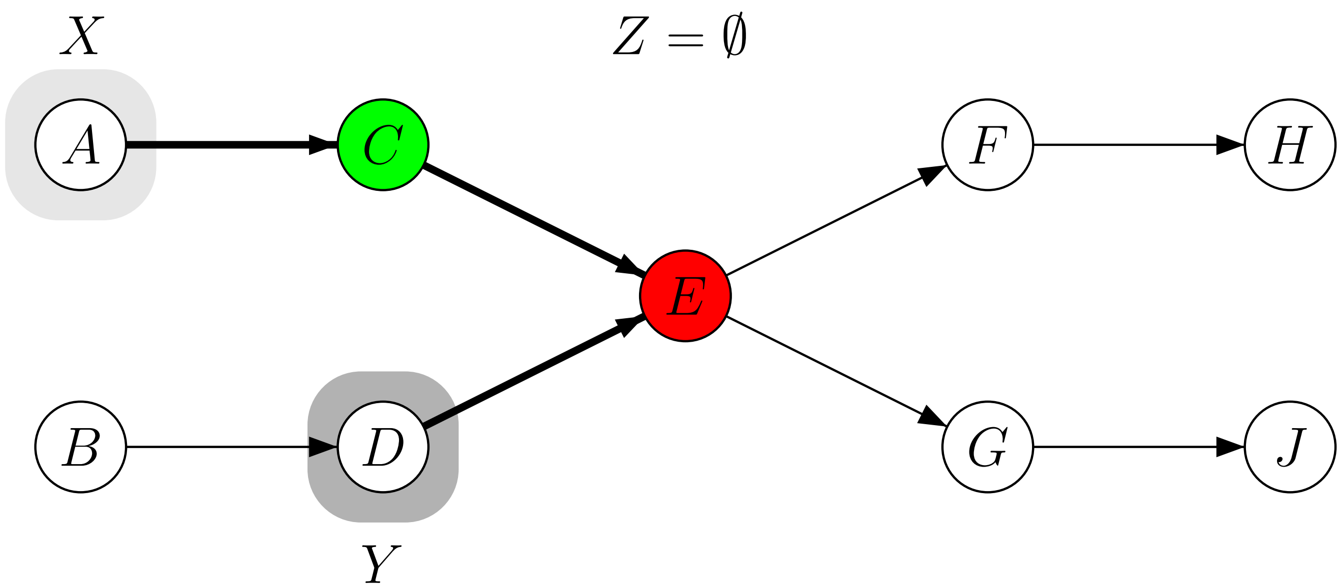
C is **serial** and not in Z: non-blocking

E is **converging** and not in Z but one of its descendants (J) is in Z: non-blocking

$\Rightarrow$ Path is not blocked

$$A \not\perp\!\!\!\perp D \mid J$$

d-Separation



Checking path $A \rightarrow C \rightarrow E \leftarrow D$:

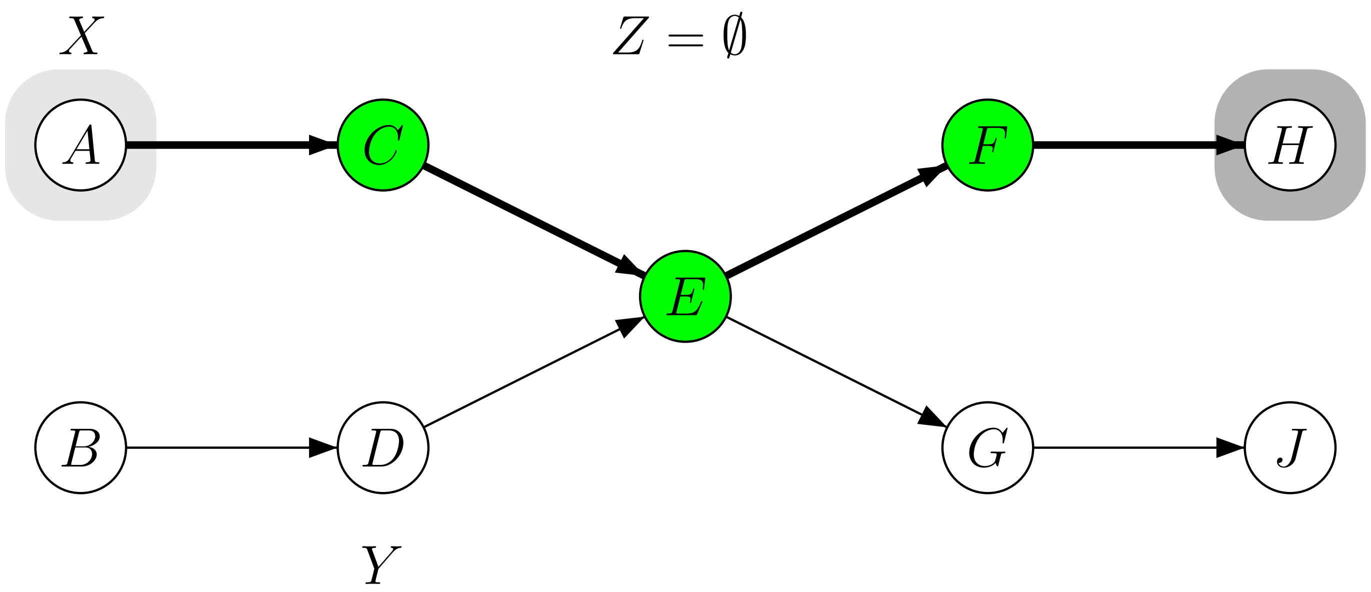
C is **serial** and not in Z: non-blocking

E is **converging** and not in Z, neither is F, G, H or J: **blocking**

$\Rightarrow$ Path is blocked

$$A \perp\!\!\!\perp D \mid \emptyset$$

d-Separation



Checking path $A \to C \to E \to F \to H$:

C is **serial** and not in Z: non-blocking

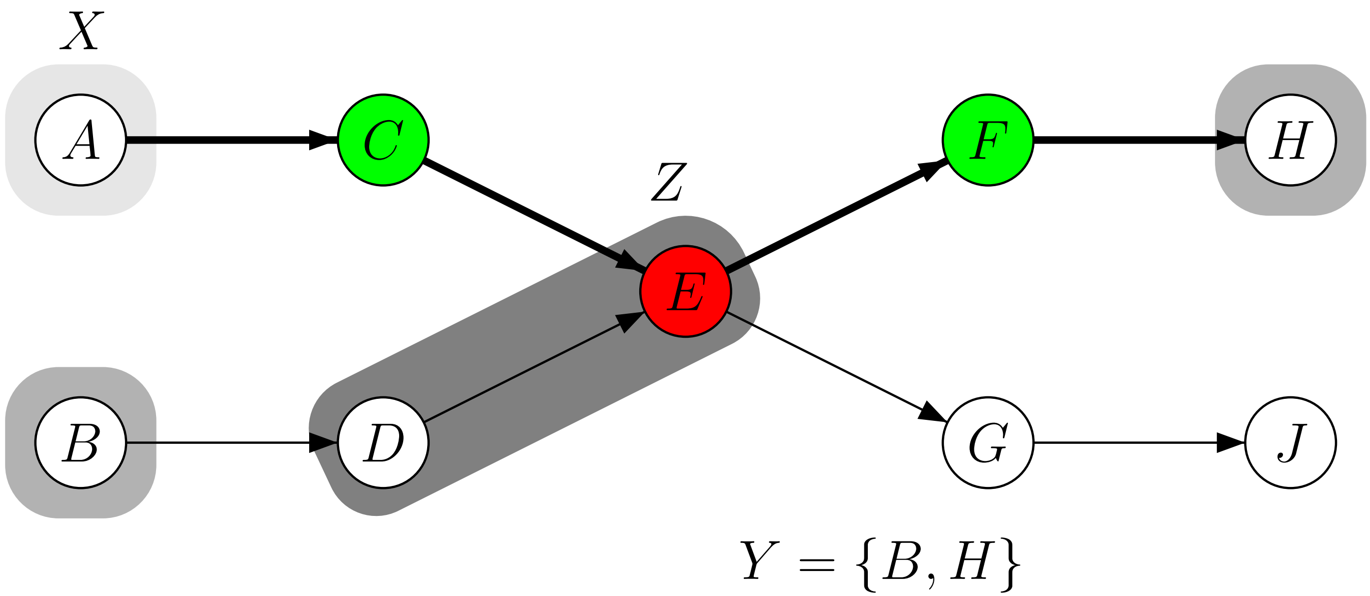
E is **serial** and not in Z: non-blocking

F is **serial** and not in Z: non-blocking

$\Rightarrow$ Path is not blocked

$$A \not\perp\!\!\!\perp H \mid \emptyset$$

d-Separation



Checking path $A \rightarrow C \rightarrow E \rightarrow F \rightarrow H$:

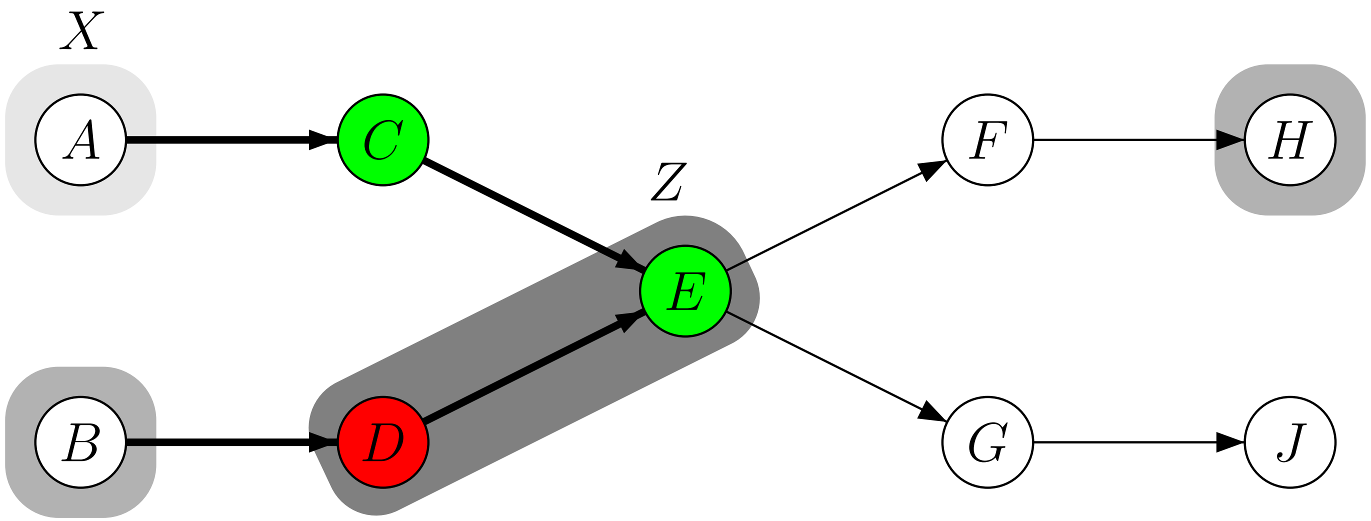
C is **serial** and not in Z: non-blocking

E is **serial** and in Z: **blocking**

F is **serial** and not in Z: non-blocking

$\Rightarrow$ Path is blocked

d-Separation



$$Y = \{B, H\}$$

Checking path $A \to C \to E \leftarrow D \to B$:

C is **serial** and not in Z: non-blocking

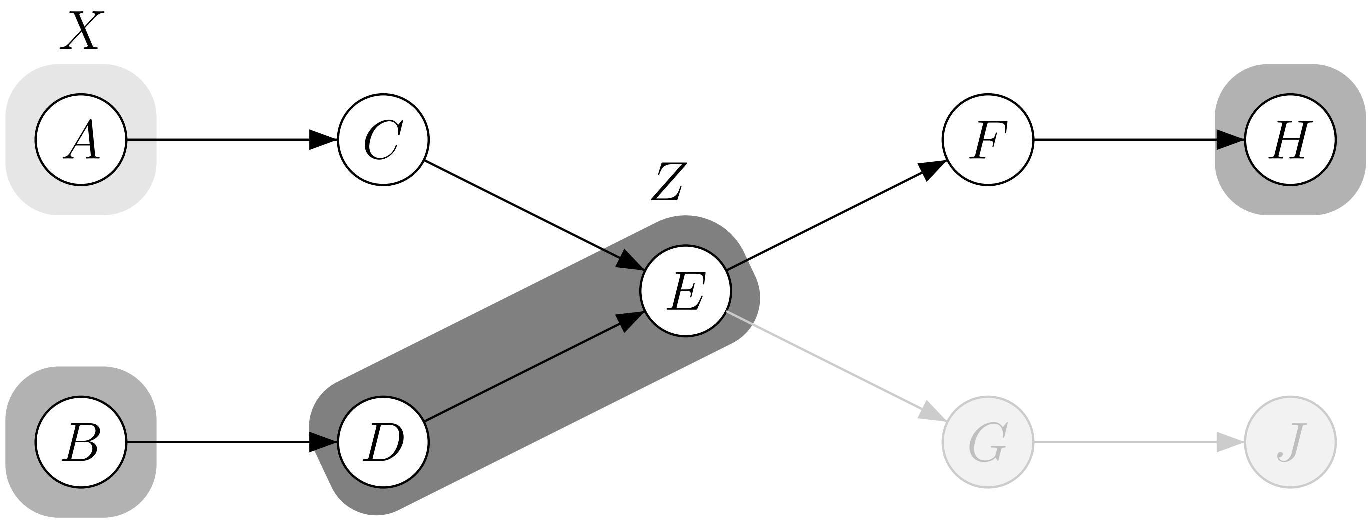
E is **converging** and in Z: non-blocking

D is **serial** and in Z: **blocking**

$\Rightarrow$ Path is blocked

$$A \perp\!\!\!\perp H, B \mid D, E$$

d-Separation: Alternative Way for Checking

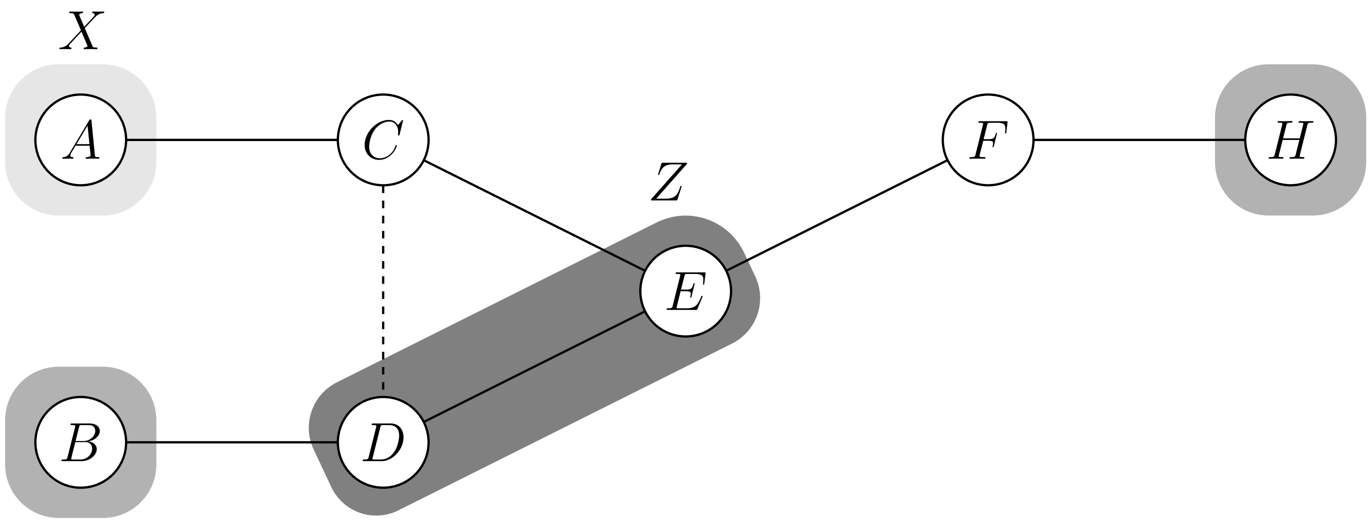


$Y = \{B, H\}$

Steps

Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$.

d-Separation: Alternative Way for Checking

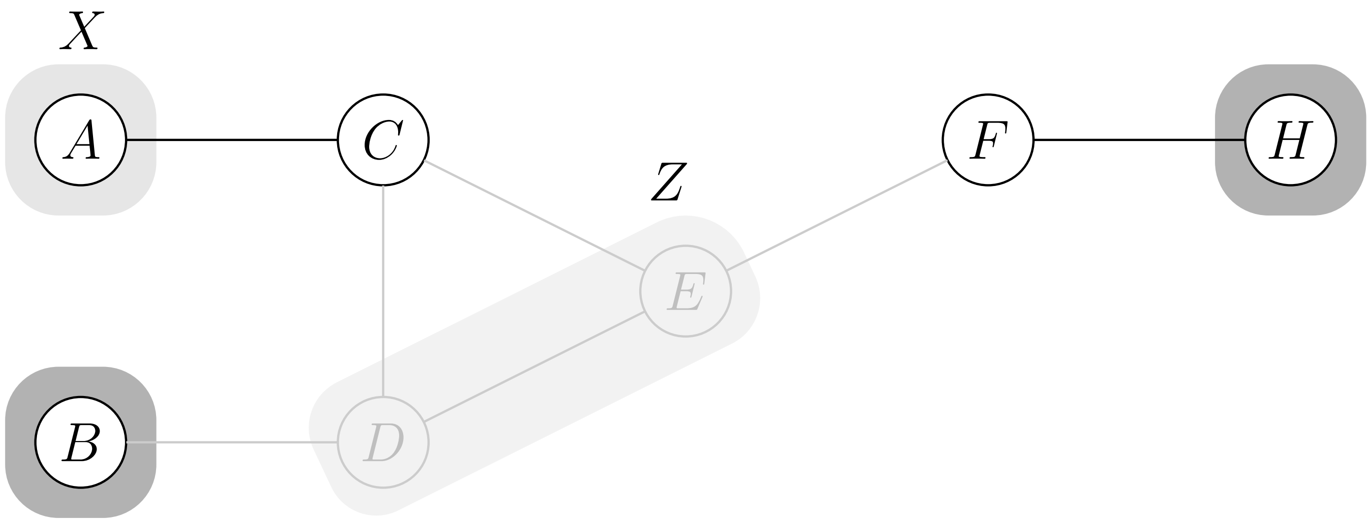


$Y = \{B, H\}$

Steps

Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$.

Moralize that subgraph.



$$Y = \{B, H\}$$

Steps:

Create the minimal ancestral subgraph induced by $X \cup Y \cup Z$.

Moralize that subgraph.

Check for u-Separation in that undirected graph.

$$A \perp\!\!\!\perp H, B \mid D, E$$

Probabilistic Graphical Models

The Big Objective(s)

In a wide variety of application fields two main problems need to be addressed over and over:

1. **How can (expert) knowledge of complex domains be efficiently represented?**

2. **How can inferences be carried out within these representations?**

3. **How can such representations be (automatically) extracted from collected data?**

We will deal with all three questions during the lecture.

Example 1: Planning in car manufacturing

Available information

"Engine type e_1 can only be combined with transmission t_2 or t_5."

"Transmission t_5 requires crankshaft c_2."

"Convertibles have the same set of radio options as SUVs."

Possible questions/inferences:

"Can a station wagon with engine e_4 be equipped with tire set y_6?"

"Supplier S_8 failed to deliver on time. What production line has to be modified and how?"

"Are there any peculiarities within the set of cars that suffered an aircondition failure?"

Example 2: Medical reasoning

Available information:

> "Malaria is much less likely than flu."
>
> "Flu causes cough and fever."
>
> "Nausea can indicate malaria as well as flu."
>
> "Nausea never indicated pneunomia before."

Possible questions/inferences

> "The patient has fever. How likely is he to have malaria?"
>
> "How much more likely does flu become if we can exclude malaria?"

Common Problems

Both scenarios share some severe problems:

Large Data Space
It is intractable to store all value combinations, i. e. all car part combinations or inter-disease dependencies.

(Example: VW Bora has 10^{200} theoretical value combinations*)

Sparse Data Space
Even if we could handle such a space, it would be extremely sparse, i. e. it would be impossible to find good estimates for all the combinations.

(Example: with 100 diseases and 200 symptoms, there would be about 10^{62} different scenarios for which we had to estimate the probability.*)

* The number of particles in the observable universe is estimated to be between 10^{78} and 10^{85}.

Idea to Solve the Problems

Given: A large (high-dimensional) distribution δ representing the domain knowledge.

Desired: A set of smaller (lower-dimensional) distributions $\{\delta_1, \ldots, \delta_s\}$ (maybe overlapping) from which the original δ *could* be reconstructed with no (or as few as possible) errors.

With such a decomposition we can draw any conclusions from $\{\delta_1, \ldots, \delta_s\}$ that could be inferred from δ — without, however, actually reconstructing it.

Example: Car Manufacturing

Let us consider a car configuration is described by three attributes:

- Engine E, $\mathrm{dom}(E) = \{e_1, e_2, e_3\}$
- Breaks B, $\mathrm{dom}(B) = \{b_1, b_2, b_3\}$
- Tires T, $\mathrm{dom}(T) = \{t_1, t_2, t_3, t_4\}$

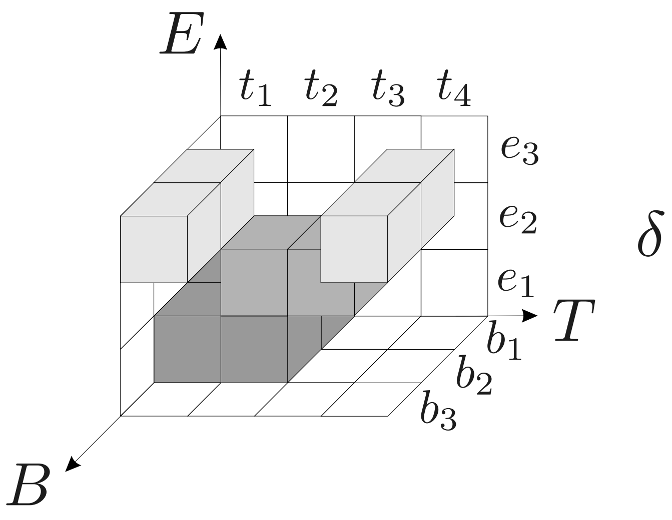
Therefore the set of all (theoretically) possible car configurations is:

$$\Omega = \mathrm{dom}(E) \times \mathrm{dom}(B) \times \mathrm{dom}(T)$$

Since not all combinations are technically possible (or wanted by marketing) a set of rules is used to cancel out invalid combinations.

Example: Car Manufacturing

Possible car configurations



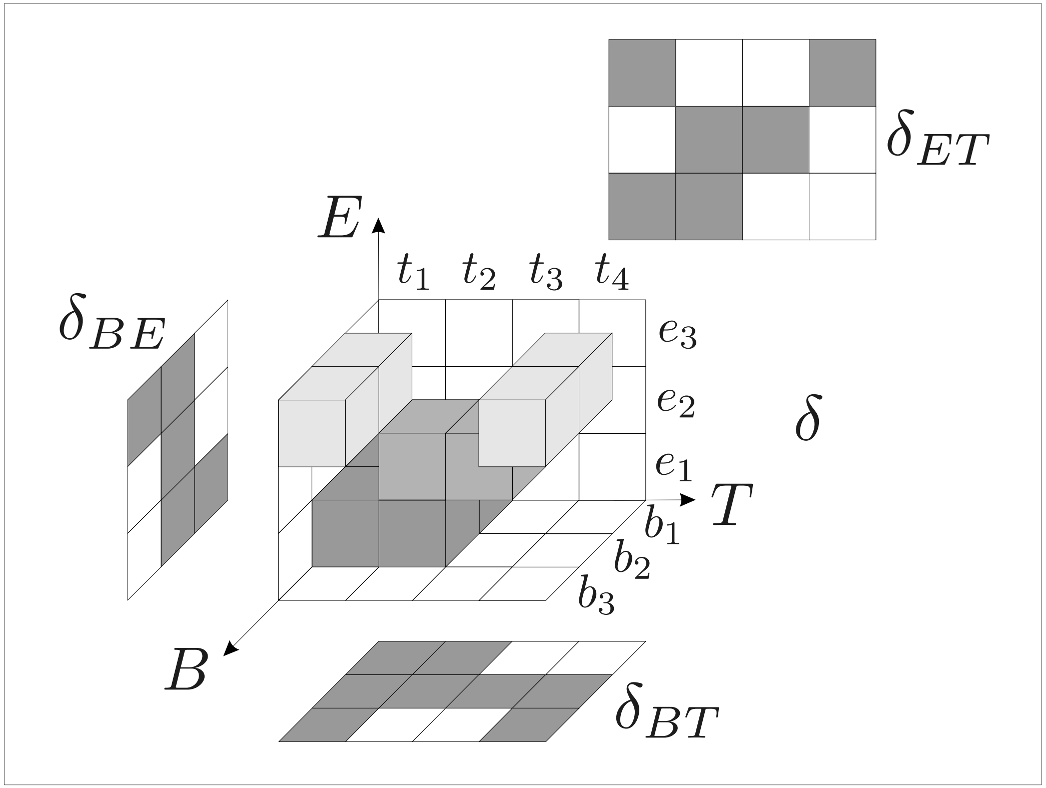
Every cube designates a valid value combination.

10 car configurations in our model.

Different colors are intended to distinguish the cubes only.

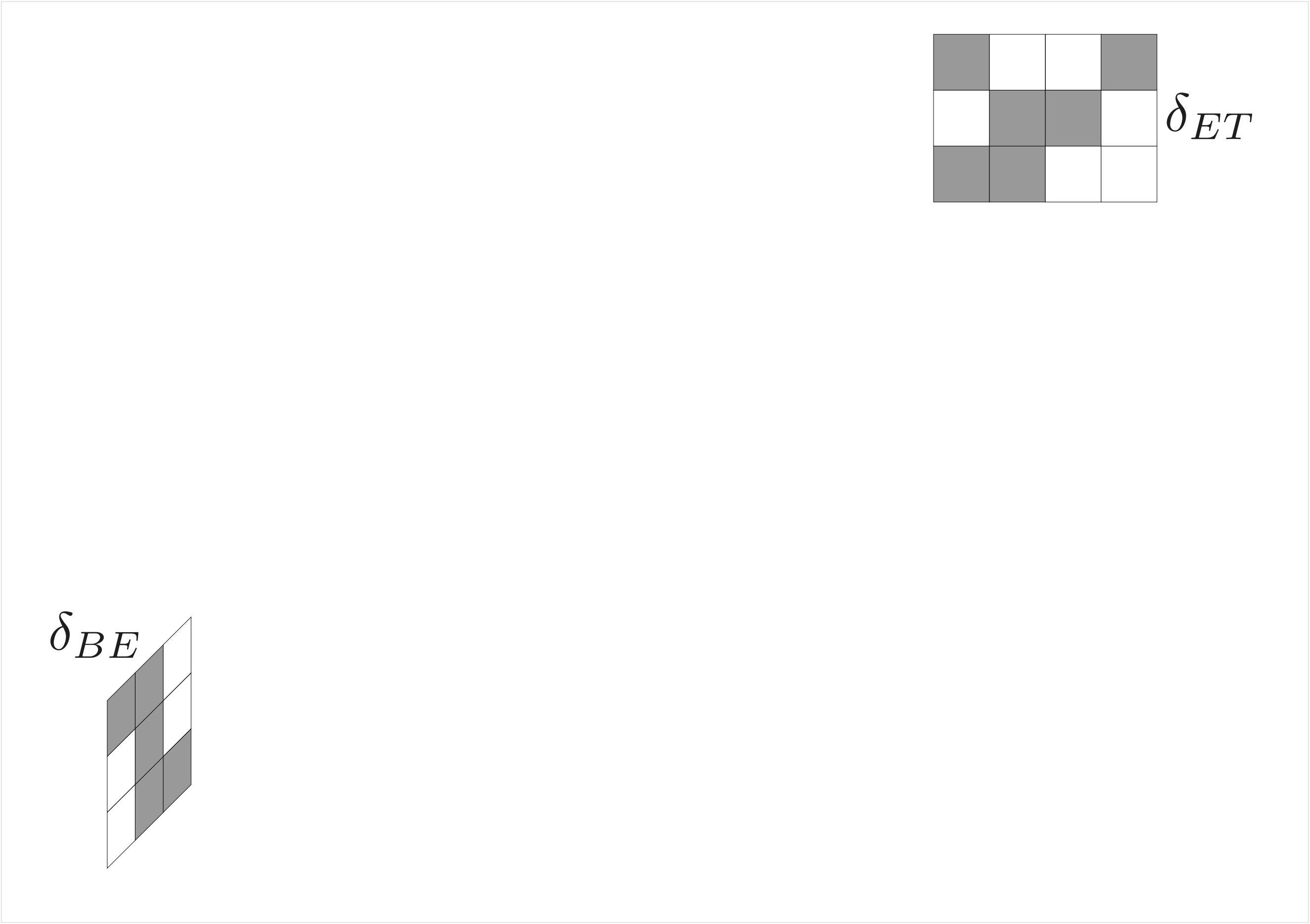
Example

2-D projections

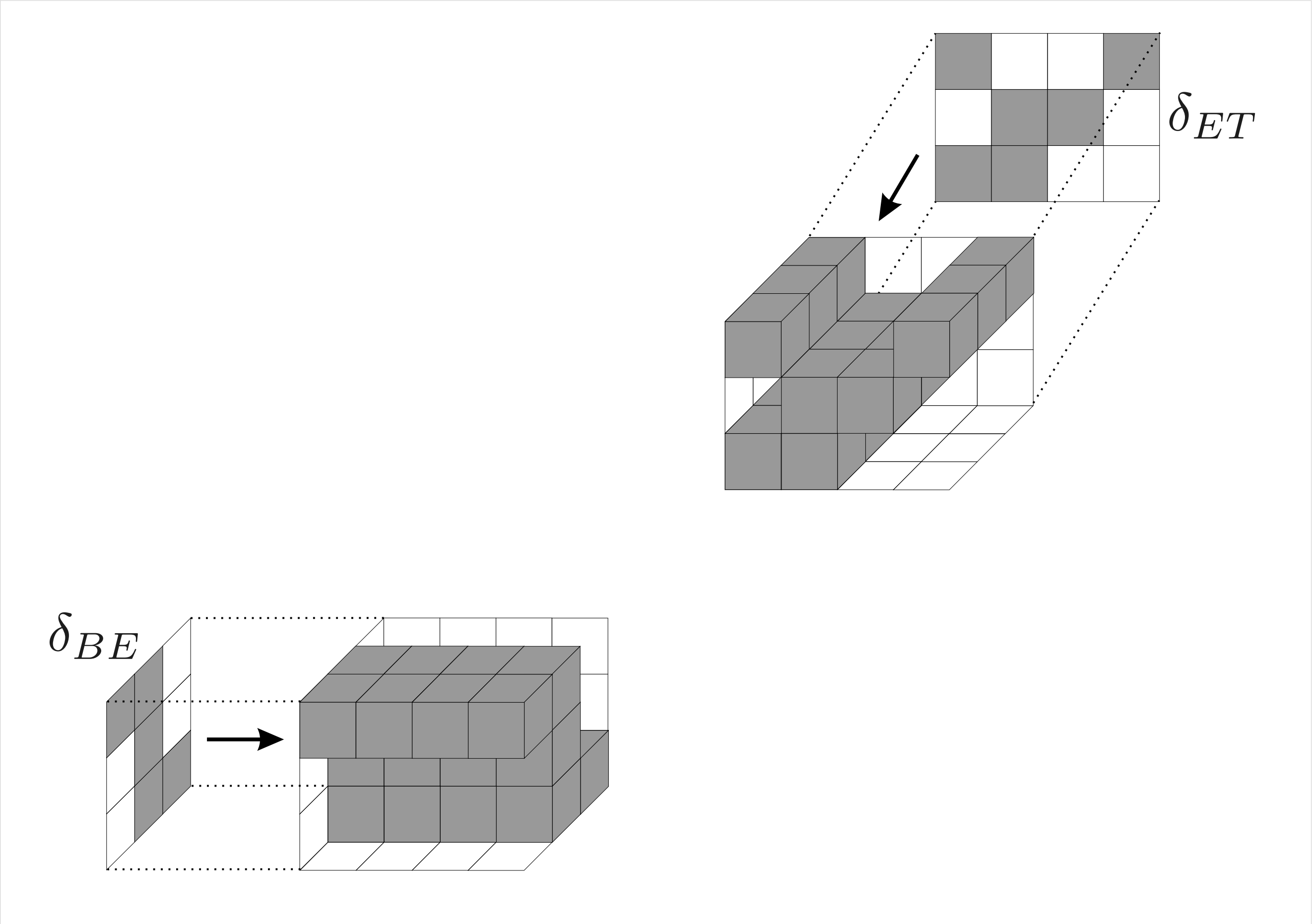


Is it possible to reconstruct δ from the δ_i?

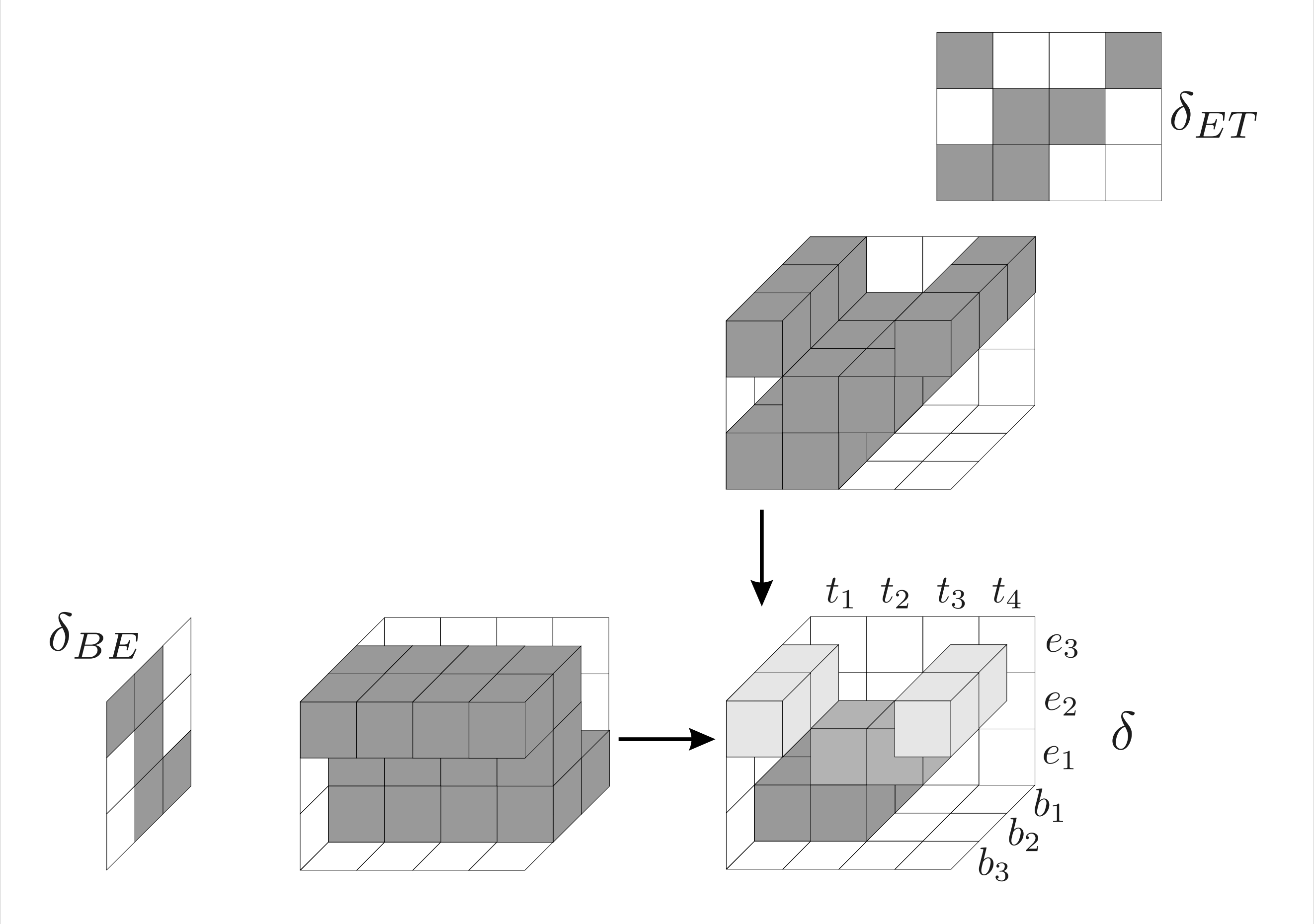
Example: Reconstruction of δ with δ_{BE} and δ_{ET}



Example: Reconstruction of δ with δ_{BE} and δ_{ET}



Example: Reconstruction of δ with δ_{BE} and δ_{ET}



Objective

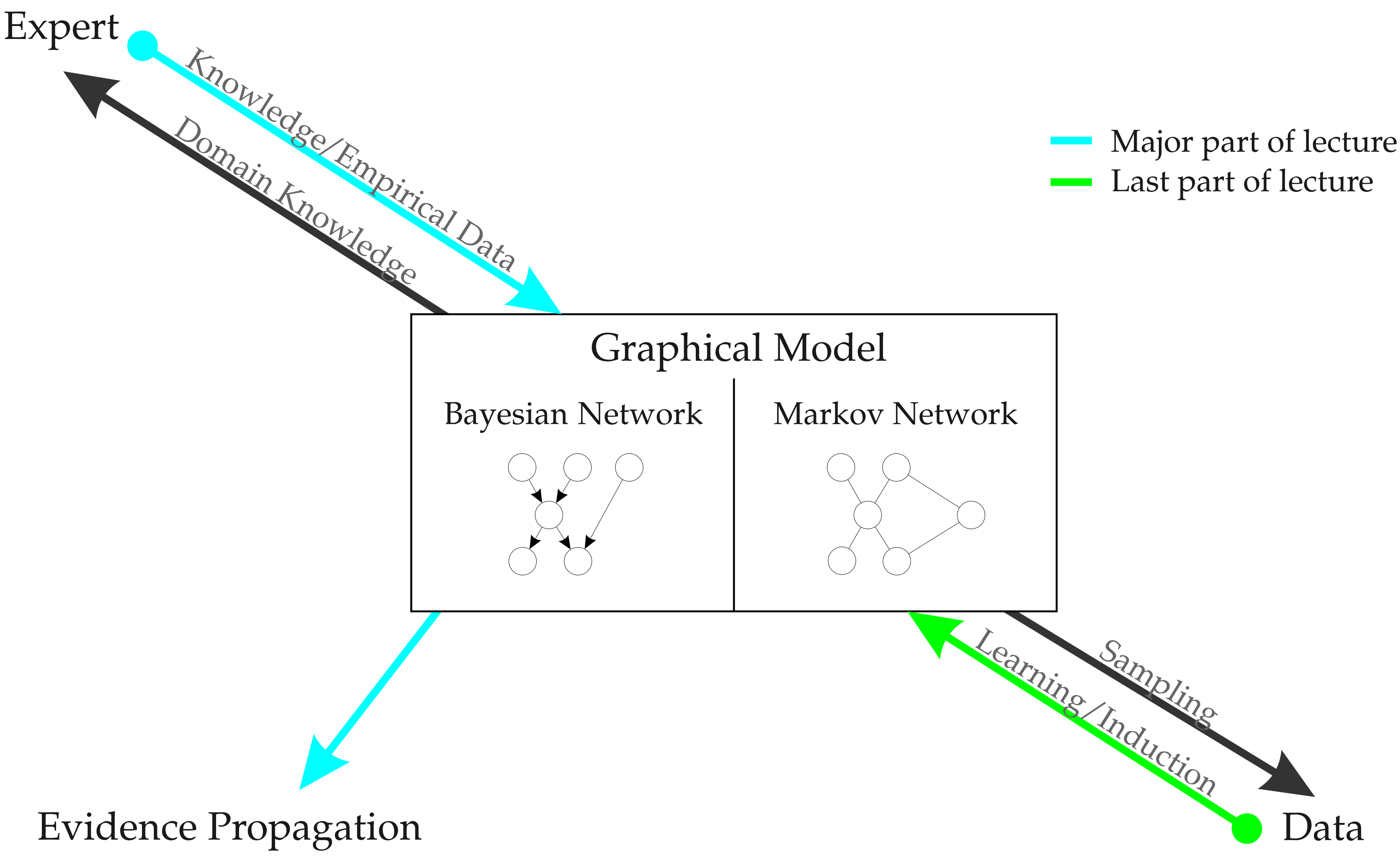
Is it possible to exploit local constraints (wherever they may come from — both structural and expert knowledge-based) in a way that allows for a decomposition of the large (intractable) distribution $P(X_1, \dots, X_n)$ into several sub-structures $\{C_1, \dots, C_m\}$ such that:

The collective size of those sub-structures is much smaller than that of the original distribution P.

The original distribution P is recomposable (with no or at least as few as possible errors) from these sub-structures in the following way:

$$P(X_1, \dots, X_n) = \prod_{i=1}^{m} \Psi_i(c_i)$$

where c_i is an instantiation of C_i and $\Psi_i(c_i) \in \mathbb{R}^+$ a *factor potential.*



Expert
Knowledge/Empirical Data
Domain Knowledge
Major part of lecture
Last part of lecture
Graphical Model
Bayesian Network
Markov Network
Sampling
Learning/Induction
Evidence Propagation
Data

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

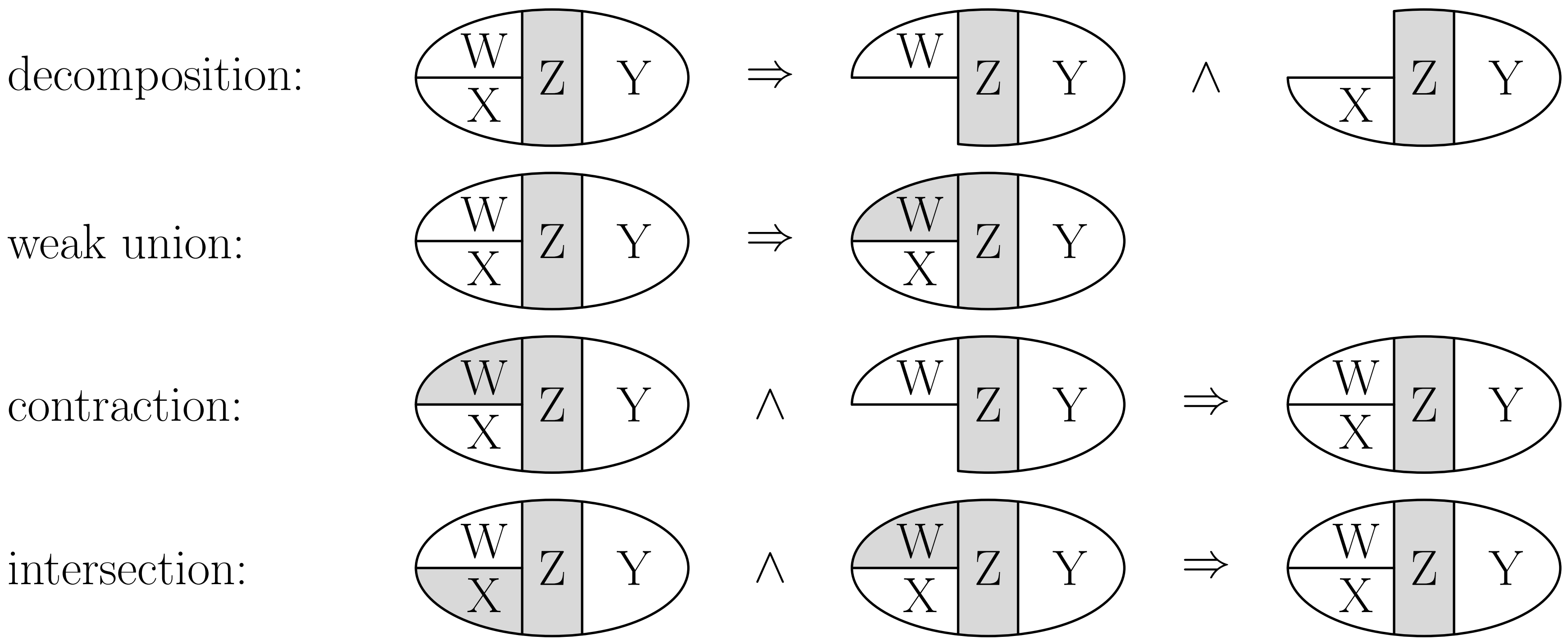
contraction: $(X \perp\!\!\!\perp Y \mid Z \cup W) \wedge (W \perp\!\!\!\perp Y \mid Z) \;\Rightarrow\; (W \cup X \perp\!\!\!\perp Y \mid Z)$

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**.

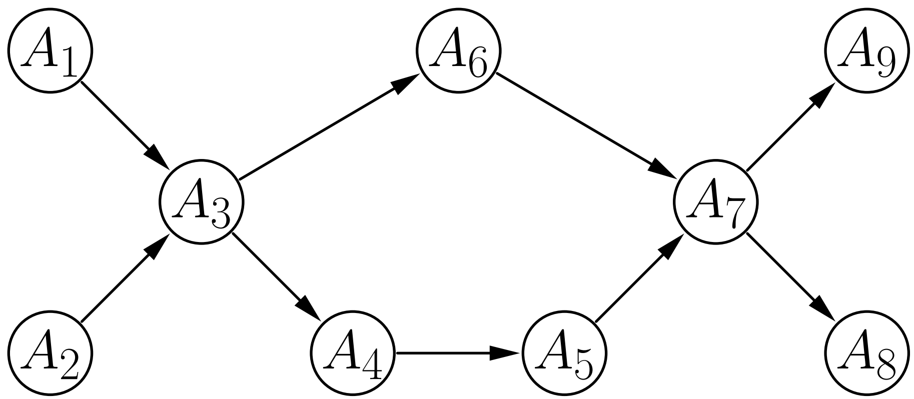
Illustration of the (Semi-)Graphoid Axioms



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

Separation in Directed Acyclic Graphs

Example Graph:



Valid Separations:

$\langle\{A_1\} \mid \{A_3\} \mid \{A_4\}\rangle$ $\qquad$ $\langle\{A_8\} \mid \{A_7\} \mid \{A_9\}\rangle$

$\langle\{A_3\} \mid \{A_4, A_6\} \mid \{A_7\}\rangle$ $\qquad$ $\langle\{A_1\} \mid \emptyset \mid \{A_2\}\rangle$

Invalid Separations:

$\langle\{A_1\} \mid \{A_4\} \mid \{A_2\}\rangle$ $\qquad$ $\langle\{A_1\} \mid \{A_6\} \mid \{A_7\}\rangle$

$\langle\{A_4\} \mid \{A_3, A_7\} \mid \{A_6\}\rangle$ $\qquad$ $\langle\{A_1\} \mid \{A_4, A_9\} \mid \{A_5\}\rangle$

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 δ 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. δ, iff for all disjoint subsets $X, Y, Z \subseteq U$ of attributes

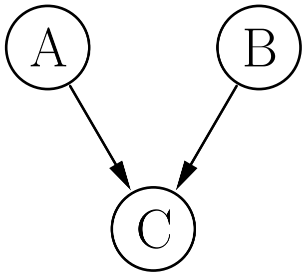
$$X \perp\!\!\!\perp_\delta Y \mid Z \;\Rightarrow\; \langle X \mid Z \mid Y \rangle_G,$$

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

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

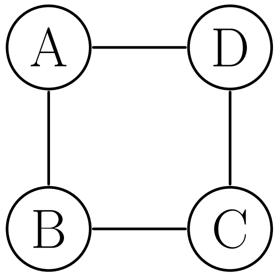
i.e., if G captures by u-separation only (conditional) independences that are valid in δ. G is said to be a **perfect map** of the conditional (in)dependences in δ, if it is both a dependence map and an independence map.

Perfect directed map, no perfect undirected map:



p_{ABC}	$A = a_1$, $B = b_1$	$A = a_1$, $B = b_2$	$A = a_2$, $B = b_1$	$A = a_2$, $B = b_2$
$C = c_1$	$^4/_{24}$	$^3/_{24}$	$^3/_{24}$	$^2/_{24}$
$C = c_2$	$^2/_{24}$	$^3/_{24}$	$^3/_{24}$	$^4/_{24}$

Perfect undirected map, no perfect directed map:



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

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 δ) the

pairwise Markov property,

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

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

local Markov property,

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

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

global Markov property,

iff in δ 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

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

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 δ) the

pairwise Markov property,

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

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

local Markov property,

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

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

global Markov property,

iff in δ 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

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

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 δ 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 δ 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.

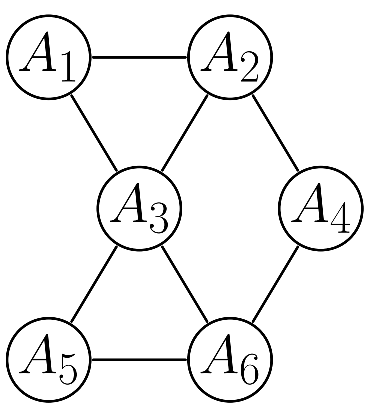
Undirected Graphs and Decompositions

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 $\phi_M : \mathcal{E}_M \to \mathbb{R}_0^+$, $M \in \mathcal{M}$, $\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$

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

Example:



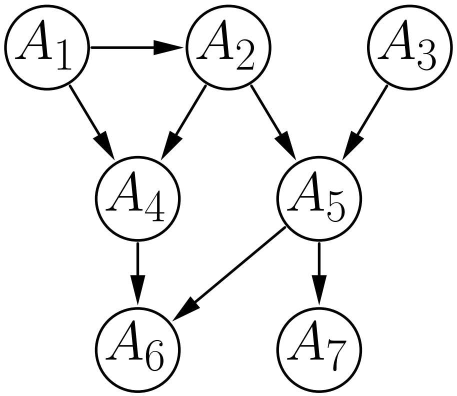
$$\begin{aligned}
&p_V(A_1 = a_1, \ldots, A_6 = a_6) \\
&\quad = \phi_{A_1A_2A_3}(A_1 = a_1, A_2 = a_2, A_3 = a_3) \\
&\quad \cdot \phi_{A_3A_5A_6}(A_3 = a_3, A_5 = a_5, A_6 = a_6) \\
&\quad \cdot \phi_{A_2A_4}(A_2 = a_2, A_4 = a_4) \\
&\quad \cdot \phi_{A_4A_6}(A_4 = a_4, A_6 = a_6).
\end{aligned}$$

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

$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$p_U\left(\bigwedge_{A_i \in U} A_i = a_i\right) = \prod_{A_i \in U} P\left(A_i = a_i \,\middle|\, \bigwedge_{A_j \in \mathrm{parents}_{\vec{G}}(A_i)} A_j = a_j\right).$$

Example:



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

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 ϕ_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.

Bayes Networks

Bayes Network

A *Bayes Network* (V, E, P) consists of a set $V = \{X_1, \ldots, X_n\}$ of random variables and a set E of directed edges between the variables.

Each variable has a finite set of mutual exclusive and collectively exhaustive states.

The variables in combination with the edges form a directed, acyclic graph.

Each variable with parent nodes $B_1, \ldots, B_m$ is assigned a table $P(A \mid B_1, \ldots, B_m)$.

Note, that the connections between the nodes not necessarily express a causal relationship.

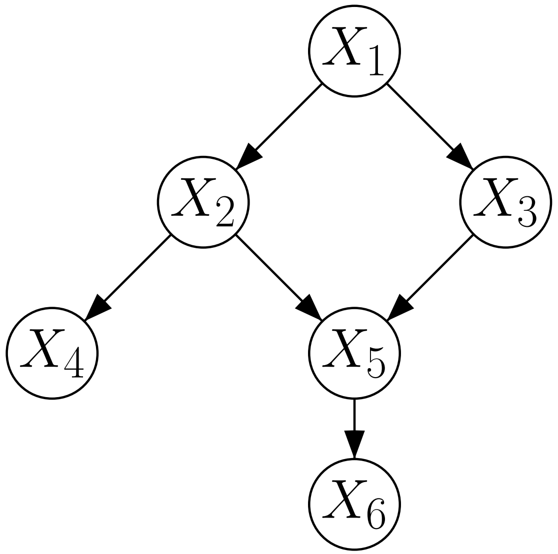
For every belief network, the following equation holds:

$$P(V) = \prod_{v \in V : P(c(v)) > 0} P(v \mid c(v))$$

with $c(v)$ being the parent nodes of v.

Probabilistic Dependency Networks

Probabilistic dependency networks are directed acyclic graphs (DAGs) where the nodes represent propositions or variables and the directed edges model a direct dependence between the connected nodes. The strength of dependence is defined by conditional probabilities.

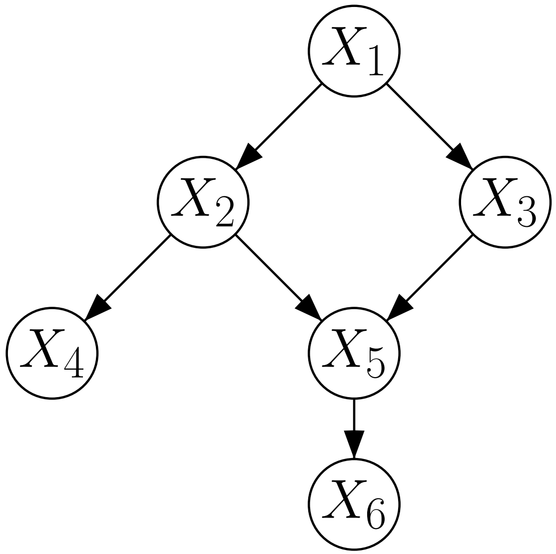


In general (according chain rule):

$$
\begin{aligned}
P(X_1, \ldots, X_6) = \; & P(X_6 \mid X_5, \ldots, X_1) \cdot \\
& P(X_5 \mid X_4, \ldots, X_1) \cdot \\
& P(X_4 \mid X_3, X_2, X_1) \cdot \\
& P(X_3 \mid X_2, X_1) \cdot \\
& P(X_2 \mid X_1) \cdot \\
& P(X_1)
\end{aligned}
$$

Probabilistic Dependency Networks

Probabilistic dependency networks are directed acyclic graphs (DAGs) where the nodes represent propositions or variables and the directed edges model a direct causal dependence between the connected nodes. The strength of dependence is defined by conditional probabilities.



According graph (independence structure):

$$\begin{aligned} P(X_1, \ldots, X_6) \ = \ & P(X_6 \mid X_5) \cdot \\ & P(X_5 \mid X_2, X_3) \cdot \\ & P(X_4 \mid X_2) \cdot \\ & P(X_3 \mid X_1) \cdot \\ & P(X_2 \mid X_1) \cdot \\ & P(X_1) \end{aligned}$$

Formal Framework

Nomenclature for the next slides:

$X_1, \ldots, X_n$	Variables (properties, attributes, random variables, propositions)
$\Omega_1, \ldots, \Omega_n$	respective finite domains (also designated with $\mathrm{dom}(X_i)$)
$\Omega = \bigtimes_{i=1}^{n} \Omega_i$	Universe of Discourse (tuples that characterize objects described by $X_1, \ldots, X_n$)
$\Omega_i = \{x_i^{(1)}, \ldots, x_i^{(n_i)}\}$	$n = 1, \ldots, n,\ n_i \in \mathbb{N}$

Formal Framework

The product space $(\Omega, 2^\Omega, P)$ is unique iff $P(\{(x_1, \ldots, x_n)\})$ is specified for all $x_i \in \{x_i^{(1)}, \ldots, x_i^{(n_i)}\}$, $i = 1, \ldots, n$.

When the distribution $P(X_1, \ldots, X_n)$ is given in tabular form, then $\prod_{i=1}^{n} |\Omega_i|$ entries are necessary.

For variables with $|\Omega_i| \geq 2$ at least 2^n entries.

The application of DAGs allows for the representation of existing (in)dependencies.

Constructing a DAG

input $P(X_1, \ldots, X_n)$
output a DAG G

1: Set the nodes of G to $\{X_1, \ldots, X_n\}$.

2: Choose a total ordering on the set of variables
(e. g. $X_1 \prec X_2 \prec \cdots \prec X_n$)

3: For X_i find the smallest (uniquely determinable) set $S_i \subseteq \{X_1, \ldots, X_n\}$ sucht that $P(X_i \mid S_i) = P(X_i \mid X_1 \ldots, X_{i-1})$.

4: Connect all nodes in S_i with X_i and store $P(X_i \mid S_i)$ as quantization of the dependencies for that node X_i (given its parents).

5: **return** G

Example

Let a_1, a_2, a_3 be three blood groups and b_1, b_2, b_3 three indications of a blood group test.

Variables:	A (blood group)	B (indication)
Domains:	$\Omega_A = \{a_1, a_2, a_3\}$	$\Omega_B = \{b_1, b_2, b_3\}$

It is conjectured that there is a causal relationship between the variables.

$P(\{(a_i, b_j)\})$	b_1	b_2	b_3	$\sum$
a_1	0.64	0.08	0.08	0.8
a_2	0.01	0.08	0.01	0.1
a_3	0.01	0.01	0.08	0.1
$\sum$	0.66	0.17	0.17	1

$$A \longrightarrow B$$

$$P(A, B) = P(B \mid A) \cdot P(A)$$

We are dealing with a belief network.

Example

Expert Knowledge

Metastatic cancer is a possible cause of brain cancer, and an explanation for elevated levels of calcium in the blood. Both phenomena together can explain that a patient falls into a coma. Severe headaches are possibly associated with a brain tumor.

Special Case

The patient has severe headaches.

Question

Will the patient is go into a coma?

Example

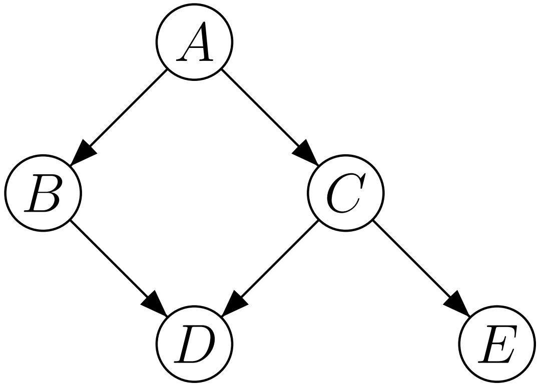
Choice of universe of discourse

	Variable	Domain
A	metastatic cancer	$\{a_1, a_2\}$
B	increased serum calcium	$\{b_1, b_2\}$
C	brain tumor	$\{c_1, c_2\}$
D	coma	$\{d_1, d_2\}$
E	headache	$\{e_1, e_2\}$

($\cdot_1$ — present, $\cdot_2$ — absent)

$$\Omega = \{a_1, a_2\} \times \cdots \times \{e_1, e_2\}$$
$$|\Omega| = 32$$

Analysis of dependencies



Example

Choice of probability parameters

$$
\begin{aligned}
P(a,b,c,d,e) &\overset{\text{abbr.}}{=} P(A=a,B=b,C=c,D=d,E=e) \\
&= P(e \mid c)P(d \mid b,c)P(c \mid a)P(b \mid a)P(a)
\end{aligned}
$$

Shorthand notation

11 values to store instead of 31

Consult experts, textbooks, case studies, surveys, etc.

Calculation of conditional probabilities

Calculation of marginal probabilities

Crux of the Matter

Knowledge acquisition (Where do the numbers come from?)
$\rightarrow$ learning strategies

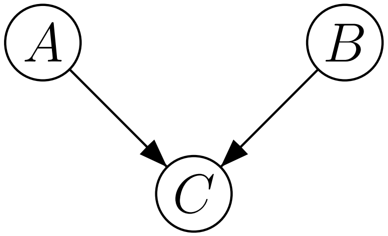
Computational complexities
$\rightarrow$ exploit independencies

Problem:

When does the independency of X and Y given Z hold in (V, E, P)?

How to determine a decomposition based of the graph structure?

Example



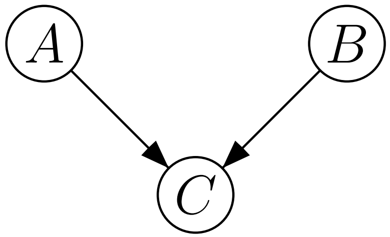
Meal quality	
A	quality of ingredients
B	cook's skill
C	meal quality

If C is not known, A and B are independent.

If C is known, then A and B become (conditionally) dependent given C.

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

Converging Connection: Marginal Independence



Decomposition according to graph:

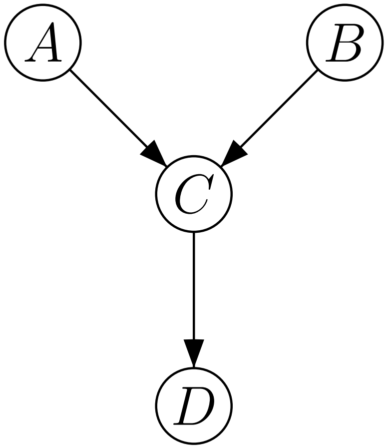
$$P(A,B,C) \;=\; P(C \mid A,B) \cdot P(A) \cdot P(B)$$

Embedded Independence:

$$P(A,B,C) \;=\; \frac{P(A,B,C)}{P(A,B)} \cdot P(A) \cdot P(B) \qquad \text{with} \quad P(A,B) \neq 0$$

$$P(A,B) \;=\; P(A) \cdot P(B)$$

$$\Rightarrow \; A \perp\!\!\!\perp B \mid \emptyset$$



Meal quality	
A	quality of ingredients
B	cook's skill
C	meal quality
D	restaurant success

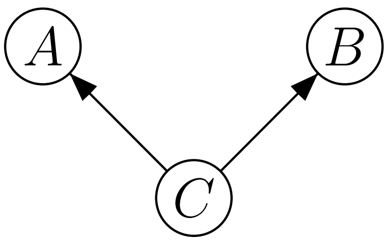
If nothing is known about the restaurant success or meal quality or both, the cook's skills and quality of the ingredients are unrelated, that is, *independent*.

However, if we observe that the restaurant has no success, we can infer that the meal quality might be bad.

If we further learn that the ingredients quality is high, we will conclude that the cook's skills must be low, thus rendering both variables *dependent*.

$A \not\perp\!\!\!\perp B \mid D$

Diverging Connection

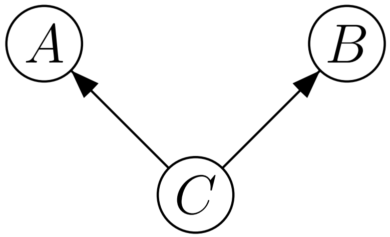


Diagnosis	
A	body temperature
B	cough
C	disease

If C is unknown, knowledge about A ist relevant for B and vice versa, i. e. A and B are marginally dependent.

However, if C is observed, A and B become conditionally independent given C.

A influences B via C. If C is known it in a way blocks the information from flowing from A to B, thus rendering A and B (conditionally) independent.



Diverging Connection: Conditional Independence

Decomposition according to graph:

$$P(A, B, C) \;=\; P(A \mid C) \cdot P(B \mid C) \cdot P(C)$$

Embedded Independence:

$$P(A, B \mid C) \;=\; P(A \mid C) \cdot P(B \mid C)$$

$$\Rightarrow\; A \perp\!\!\!\perp B \mid C$$

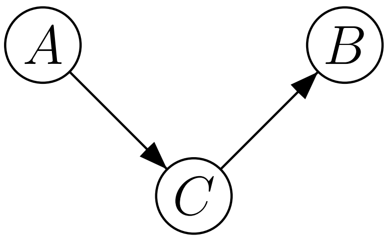
Alternative derivation:

$$P(A, B, C) \;=\; P(A \mid C) \cdot P(B, C)$$

$$P(A \mid B, C) \;=\; P(A \mid C)$$

$$\Rightarrow\; A \perp\!\!\!\perp B \mid C$$

Serial Connection



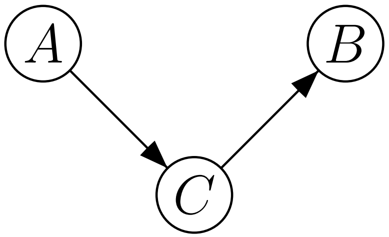
Accidents	
A	rain
B	accident risk
C	road conditions

Analog scenario to case 2

A influences C and C influences B. Thus, A influences B.
If C is known, it blocks the path between A and B.

Formal Representation

Serial Connection: Conditional Independence



Decomposition according to graph:

$$P(A,B,C) \;=\; P(B \mid C) \cdot P(C \mid A) \cdot P(A)$$

Embedded Independence:

$$P(A,B,C) \;=\; P(B \mid C) \cdot P(C,A)$$

$$P(B \mid C,A) \;=\; P(B \mid C)$$

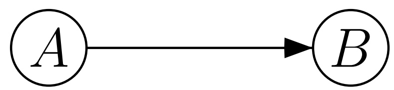
$$\Rightarrow\; A \perp\!\!\!\perp B \mid C$$

Trivial Cases:

Marginal Independence:

$$P(A, B) \;=\; P(A) \cdot P(B)$$

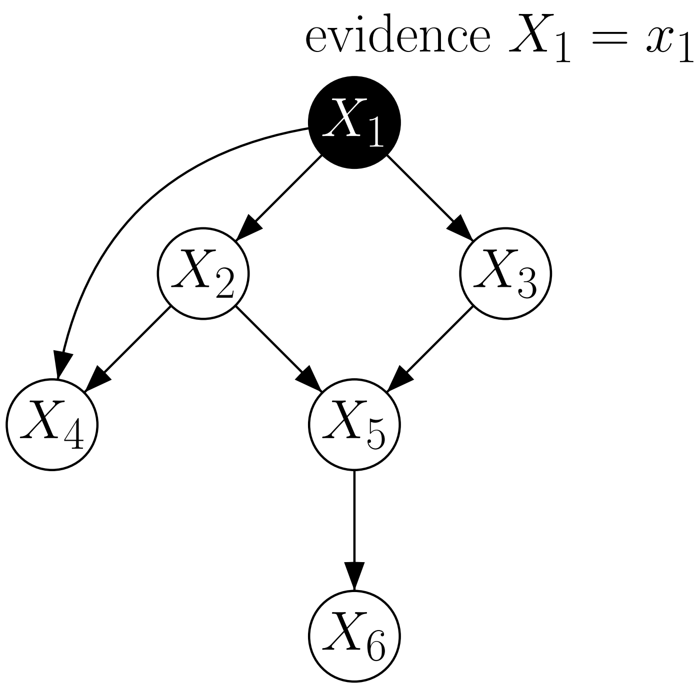
Marginal Dependence:



$$P(A, B) \;=\; P(B \mid A) \cdot P(A)$$

Question

Question: Are X_2 and X_3 independent given X_1?



Repetition: d-Separation

Let $G = (V, E)$ a DAG and $X, Y, Z \in V$ three nodes.

a) A set $S \subseteq V \backslash \{X, Y\}$ *d-separates* X and Y, if S blocks all paths between X and Y. (paths may also route in opposite edge direction)

b) A path π is d-separated by S if at least one pair of consecutive edges along π is blocked. There are the following blocking conditions:

1.	$X \leftarrow Y \rightarrow Z$	tail-to-tail
2.	$X \leftarrow Y \leftarrow Z$ $X \rightarrow Y \rightarrow Z$	head-to-tail
3.	$X \rightarrow Y \leftarrow Z$	head-to-head

c) Two edges that meet tail-to-tail or head-to-tail in node Y are blocked if $Y \in S$.

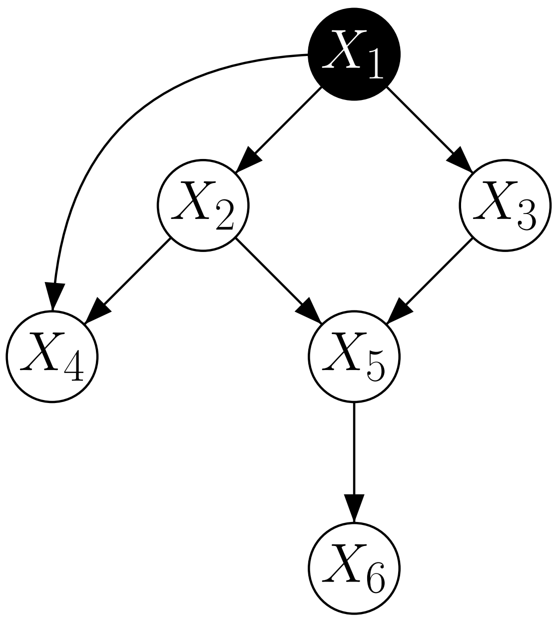
d) Two edges meeting head-to-head in Y are blocked if neither Y nor its successors are in S.

Relation to Conditional independence

If $S \subseteq V \backslash \{X, Y\}$ d-separates X and Y in a Belief network (V, E, P) then X and Y are conditionally independent given S:

$$P(X, Y \mid S) = P(X \mid S) \cdot P(Y \mid S)$$

Application to the previous example:



Paths: $\pi_1 = \langle X_2 - X_1 - X_3 \rangle$, $\pi_2 = \langle X_2 - X_5 - X_3 \rangle$
$\pi_3 = \langle X_2 - X_4 - X_1 - X_3 \rangle$, $S = \{X_1\}$

π_1 $X_2 \leftarrow X_1 \rightarrow X_3$ tail-to-tail
$X_1 \in S \;\Rightarrow\; \pi_1$ is blocked by S

π_2 $X_2 \rightarrow X_5 \leftarrow X_3$ head-to-head
$X_5, X_6 \notin S \;\Rightarrow\; \pi_2$ is blocked by S

π_3 $X_4 \leftarrow X_1 \rightarrow X_3$ tail-to-tail
$X_2 \rightarrow X_4 \leftarrow X_1$ head-to-head
both connections are blocked $\;\Rightarrow\; \pi_3$ is blocked

Example (cont.)

Answer: X_2 and X_3 are d-separated via $\{X_1\}$. Therefore X_2 and X_3 become conditionally independent given X_1.

$S = \{X_1, X_4\} \quad \Rightarrow \quad X_2$ and X_3 are d-separated by S

$S = \{X_1, X_6\} \quad \Rightarrow \quad X_2$ and X_3 are *not* d-separated by S

Algebraic structure of CI statements

Question: Is it possible to use a formal scheme to infer new conditional independence (CI) statements from a set of initial CIs?

Repetition

Let $(\Omega, \mathcal{E}, P)$ be a probability space and W, X, Y, Z disjoint subsets of variables. If X and Y are conditionally independent given Z we write:

$$X \perp\!\!\!\perp_P Y \mid Z$$

Often, the following (equivalent) notation is used:

$$I_P(X \mid Z \mid Y) \quad \text{or} \quad I_P(X, Y \mid Z)$$

If the underlying space is known the index P is omitted.

(Semi-)Graphoid-Axioms

Let $(\Omega, \mathcal{E}, P)$ be a probability space and W, X, Y and Z four disjoint subsets of random variables (over Ω). Then the propositions

a) Symmetry: $(X \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (Y \perp\!\!\!\perp_P X \mid Z)$

b) Decomposition: $(W \cup X \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (W \perp\!\!\!\perp_P Y \mid Z) \wedge (X \perp\!\!\!\perp_P Y \mid Z)$

c) Weak Union: $(W \cup X \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (X \perp\!\!\!\perp_P Y \mid Z \cup W)$

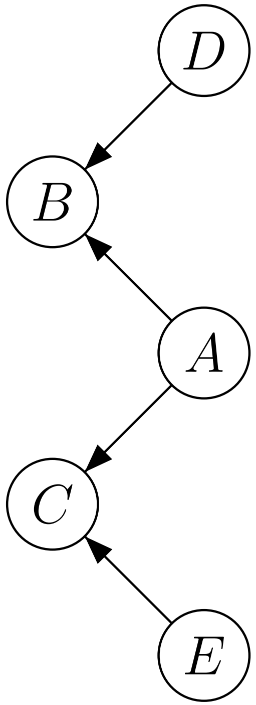
d) Contraction: $(X \perp\!\!\!\perp_P Y \mid Z \cup W) \wedge (W \perp\!\!\!\perp_P Y \mid Z) \;\Rightarrow\; (W \cup X \perp\!\!\!\perp_P Y \mid Z)$

are called the **Semi-Graphoid Axioms**. The above propositions and

e) Intersection: $(W \perp\!\!\!\perp_P Y \mid Z \cup X) \wedge (X \perp\!\!\!\perp_P Y \mid Z \cup W) \;\Rightarrow\; (W \cup X \perp\!\!\!\perp_P Y \mid Z)$

are called the **Graphoid Axioms**.

Example



Proposition: $B \perp\!\!\!\perp C \mid A$

Proof: $D \perp\!\!\!\perp A, C \mid \emptyset \quad \wedge \quad B \perp\!\!\!\perp C \mid A, D$

$\overset{\text{w. union}}{\Longrightarrow} \quad D \perp\!\!\!\perp C \mid A \quad \wedge \quad B \perp\!\!\!\perp C \mid A, D$

$\overset{\text{symm.}}{\Longleftrightarrow} \quad C \perp\!\!\!\perp D \mid A \quad \wedge \quad C \perp\!\!\!\perp B \mid A, D$

$\overset{\text{contr.}}{\Longrightarrow} \quad C \perp\!\!\!\perp B, D \mid A$

$\overset{\text{decomp.}}{\Longrightarrow} \quad C \perp\!\!\!\perp B \mid A$

$\overset{\text{symm.}}{\Longleftrightarrow} \quad B \perp\!\!\!\perp C \mid A$

Inference in Belief Trees

Inference in Belief Trees

A Bayesian Network is a complete model for the variables and their relationships.

It can be used to answer queries about them.

Typical question:
Given observes variables, what is the updated knowledge about the other variables.

There are exact inference methods for this task.

Motivation

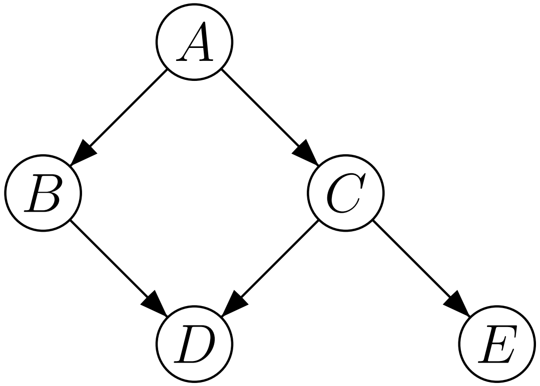
Choice of universe of discourse

	Variable	Domain
A	metastatic cancer	$\{a_1, a_2\}$
B	increased serum calcium	$\{b_1, b_2\}$
C	brain tumor	$\{c_1, c_2\}$
D	coma	$\{d_1, d_2\}$
E	headache	$\{e_1, e_2\}$

$(\cdot_1$ — present, $\cdot_2$ — absent$)$

$$\Omega \ = \{a_1, a_2\} \times \cdots \times \{e_1, e_2\}$$
$$|\Omega| = 32$$

Analysis of dependencies



Motivation

$$\left.\begin{aligned} P(e_1 \mid c_1) &= 0.8 \\ P(e_1 \mid c_2) &= 0.6 \end{aligned}\right\} \text{headaches common, but more common if tumor present}$$

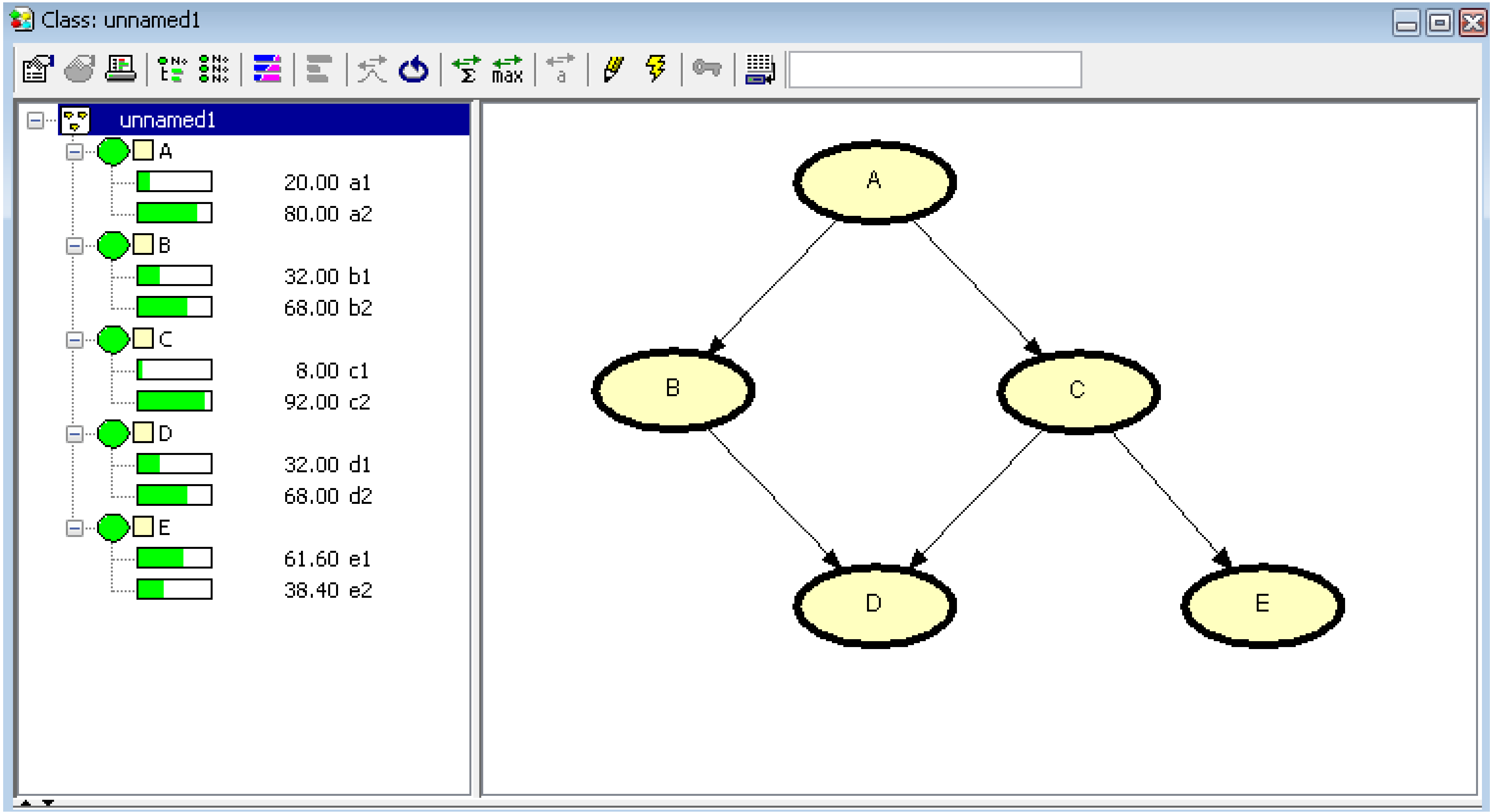
$$\left.\begin{aligned} P(d_1 \mid b_1, c_1) &= 0.8 \\ P(d_1 \mid b_1, c_2) &= 0.8 \\ P(d_1 \mid b_2, c_1) &= 0.8 \\ P(d_1 \mid b_2, c_2) &= 0.05 \end{aligned}\right\} \text{coma rare but common, if either cause is present}$$

$$\left.\begin{aligned} P(b_1 \mid a_1) &= 0.8 \\ P(b_1 \mid a_2) &= 0.2 \end{aligned}\right\} \begin{array}{c}\text{increased calcium uncommon,} \\ \text{but common consequence of metastases}\end{array}$$

$$\left.\begin{aligned} P(c_1 \mid a_1) &= 0.2 \\ P(c_1 \mid a_2) &= 0.05 \end{aligned}\right\} \text{brain tumor rare, and uncommon consequence of metastases}$$

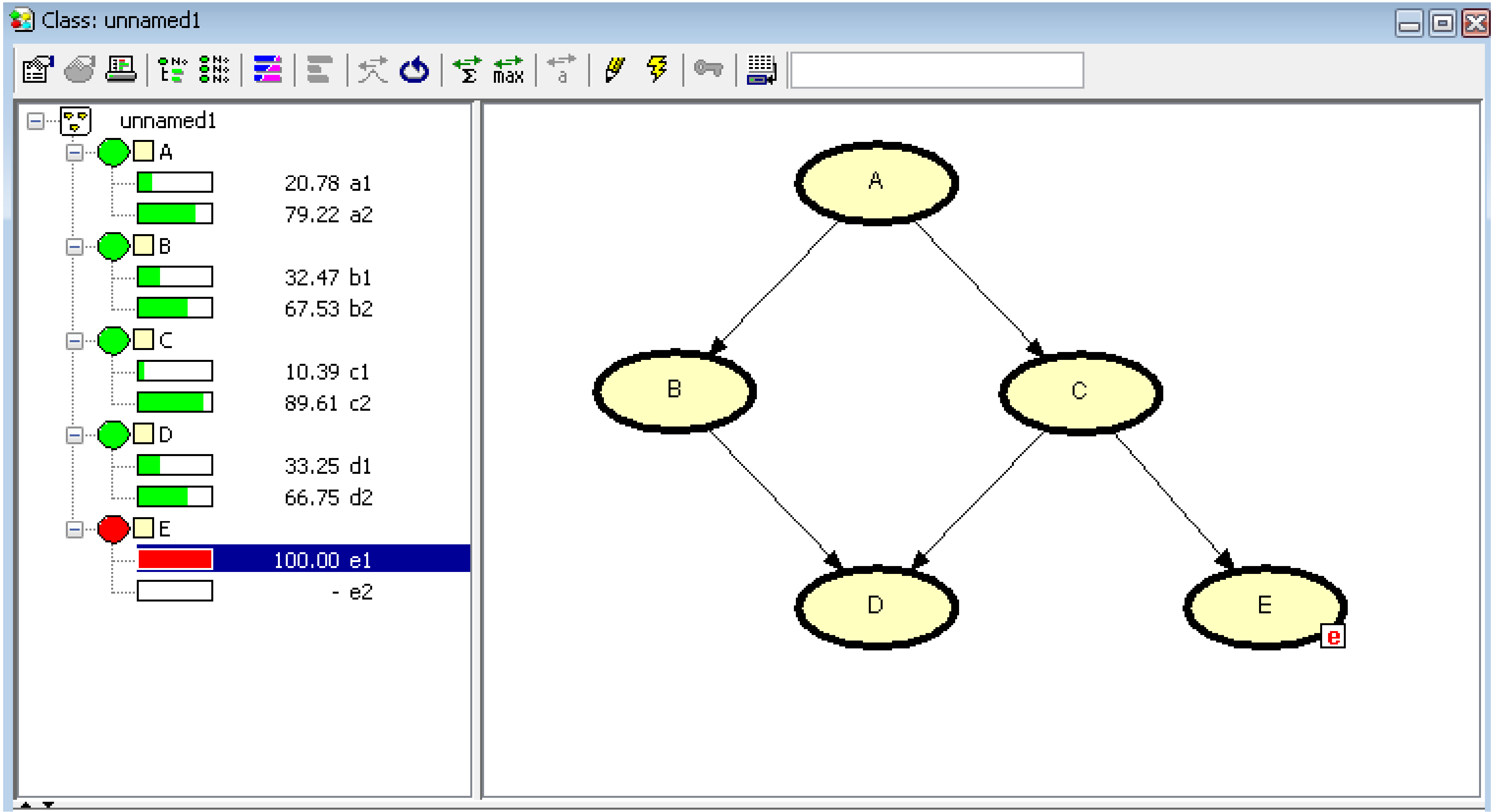
$$\left. P(a_1) = 0.2 \right\} \text{incidence of metastatic cancer in relevant clinic}$$

Motivation



Marginal distributions in the HUGIN tool.

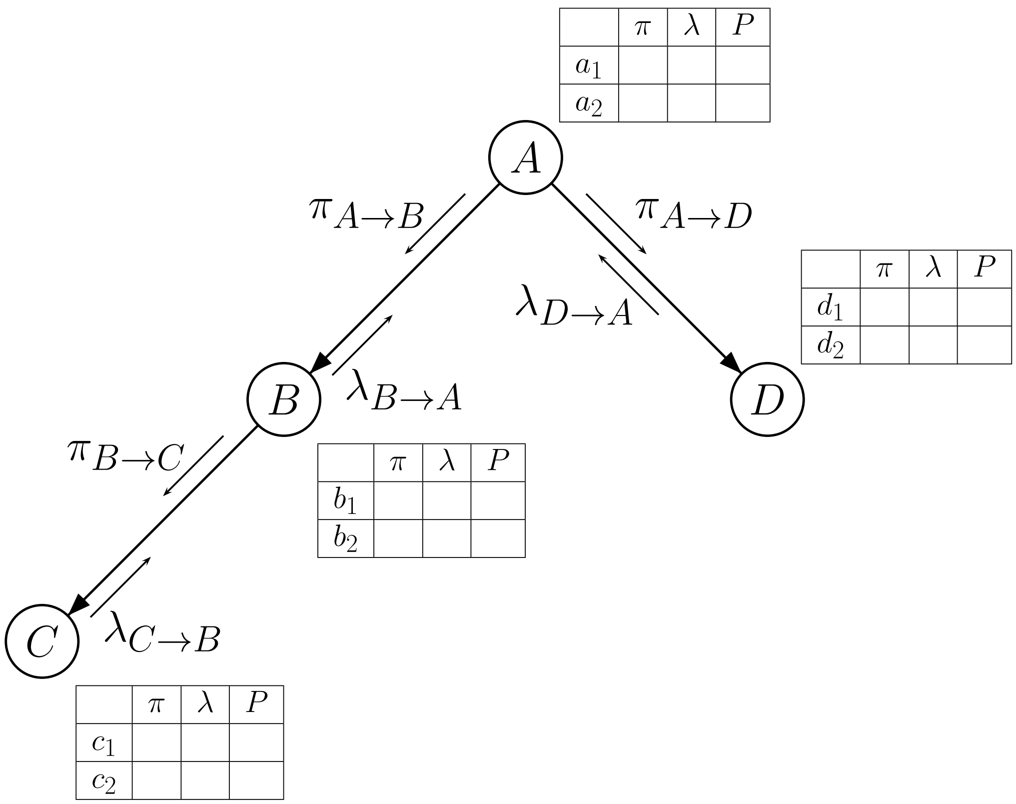
Motivation



Conditional marginal distributions with evidence $E = e_1$

Propagation in Belief Trees

Belief Tree:



Parameters:

$$P(a_1) = 0.1 \quad P(b_1 \mid a_1) = 0.7$$
$$P(b_1 \mid a_2) = 0.2$$
$$P(d_1 \mid a_1) = 0.8 \quad P(c_1 \mid b_1) = 0.4$$
$$P(d_1 \mid a_2) = 0.4 \quad P(c_1 \mid b_2) = 0.001$$

Desired:

$\forall X \forall Y \in \{A, B, C, D\}: \; P(X \mid Y) = ?$

Objective

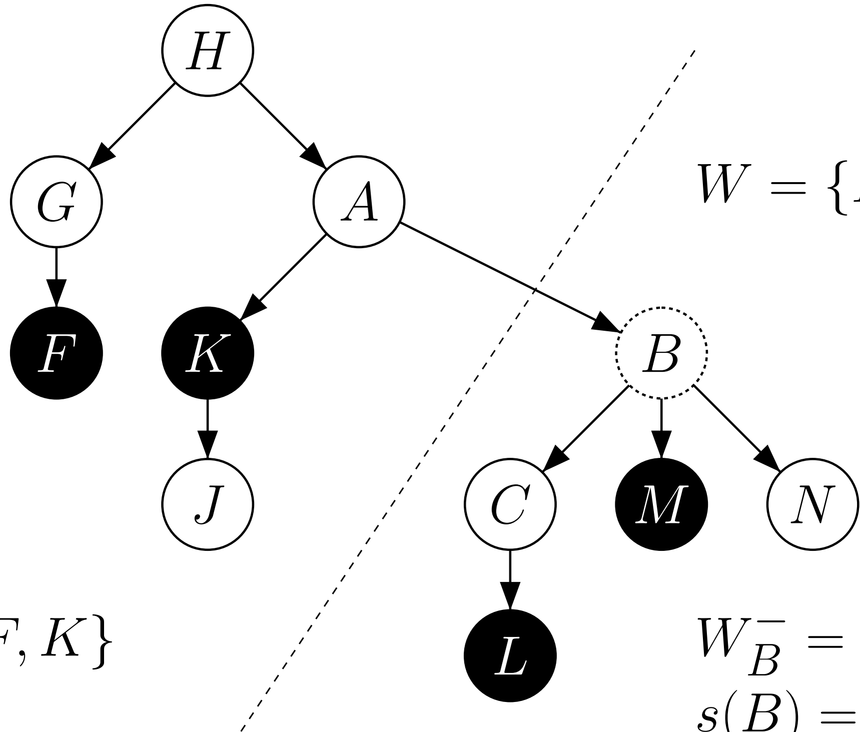
Given: Belief network (V, E, P) with tree structure and $P(V) > 0$.
Set $W \subseteq V$ of instantiated variables where
a priori knowledge $W \neq \emptyset$ is allowed

Desired: $P(B \mid W)$ for all $B \in V$

Notation:

W_B^-	subset of those variables of W that belong to the subtree of (V, E) that has root B
W_B^+	$= W \backslash W_B^-$
$s(B)$	set of direct successors of B
Ω_B	domain of B
b^*	value that B is instantiated with

Example



$$W = \{F, K, L, M\}$$

$$W_B^+ = \{F, K\}$$

$$W_B^- = \{L, M\}$$

$$s(B) = \{C, M, N\}$$

Decomposition in the Tree

$$
\begin{aligned}
P(B = b \mid W) &= P(b \mid W_B^- \cup W_B^+) \qquad \text{with } B \notin W \\
&= \frac{P(W_B^- \cup W_B^+ \cup \{b\})}{P(W_B^- \cup W_B^+)} \\
&= \frac{P(W_B^- \cup W_B^+ \mid b) P(b)}{P(W_B^- \cup W_B^+)} \\
&= \frac{P(W_B^- \mid b) P(W_B^+ \mid b) P(b)}{P(W_B^- \cup W_B^+)} \\
&= \beta_{B,W} \quad \underbrace{P(W_B^- \mid b)}_{\text{Evidence from ``below''}} \quad \underbrace{P(b \mid W_B^+)}_{\text{Evidence from ``above''}}
\end{aligned}
$$

π- and λ-Values

Since we ignore the constant $\beta_{B,W}$ for the derivations below, the following designations are used instead of $P(\cdot)$:

π-values and λ-values

Let $B \in V$ be a variable and $b \in \Omega_B$ a value of its domain. We define the π- and λ-values as follows:

$$\lambda(b) = \begin{cases} P(W_B^- \mid b) & \text{if } B \notin W \\ \quad 1 & \text{if } B \in W \wedge b^* = b \\ \quad 0 & \text{if } B \in W \wedge b^* \neq b \end{cases}$$

$$\pi(b) = P(b \mid W_B^+)$$

π- and λ-Values

$$\lambda(b) = \prod_{C \in s(B)} P(W_C^- \mid b) \qquad \text{if } B \notin W$$

$$\lambda(b) = 1 \qquad \text{if } B \text{ leaf in } (V, E)$$

$$\pi(b) = P(b) \qquad \text{if } B \text{ root in } (V, E)$$

$$P(b \mid W) = \alpha_{B,W} \cdot \lambda(b) \cdot \pi(b)$$

λ-message

Let $B \in V$ be an attribute and $C \in s(B)$ its direct children with the respective domains $\mathrm{dom}(B) = \{B_1, \ldots, b_i, \ldots, b_k\}$ and $\mathrm{dom}(C) = \{c_1, \ldots, c_j, \ldots, c_m\}$.

$$\lambda_{C \to B}(b_i) \stackrel{\text{Def}}{=} \sum_{j=1}^{m} P(c_j \mid b_i) \cdot \lambda(c_j), \qquad i = 1, \ldots, k$$

The vector

$$\vec{\lambda}_{C \to B} \stackrel{\text{Def}}{=} \Big(\lambda_{C \to B}(b_i)\Big)_{i=1}^{k}$$

is called λ-*message* from C to B.

λ-Message

Let $B \in V$ an attribute an $b \in \mathrm{dom}(B)$ a value of its domain.
Then

$$\lambda(b) = \begin{cases} \rho_{B,W} \cdot \prod\limits_{C \in s(B)} \lambda_{C \to B}(b) & \text{if } B \notin W \\ 1 & \text{if } B \in W \wedge b = b^* \\ 0 & \text{if } B \in W \wedge b \neq b^* \end{cases}$$

with $\rho_{B,W}$ being a positive constant.

π-message

Let $B \in V$ be a non-root node in (V, E) and $A \in V$ its parent with domain $\mathrm{dom}(A) = \{a_1, \ldots, a_j, \ldots, a_m\}$.

$$j = 1, \ldots, m :$$
$$\pi_{A \to B}(a_j) \stackrel{\text{Def}}{=} \begin{cases} \pi(a_j) \cdot \prod\limits_{C \in s(A) \setminus \{B\}} \lambda_{C \to B}(a_j) & \text{if } A \notin W \\ 1 & \text{if } A \in W \wedge a = a^* \\ 0 & \text{if } A \in W \wedge a \neq a^* \end{cases}$$

The vector

$$\vec{\pi}_{A \to B} \stackrel{\text{Def}}{=} \Big(\pi_{A \to B}(a_j)\Big)_{j=1}^{m}$$

is called π-*message* from A to B.

π-Message

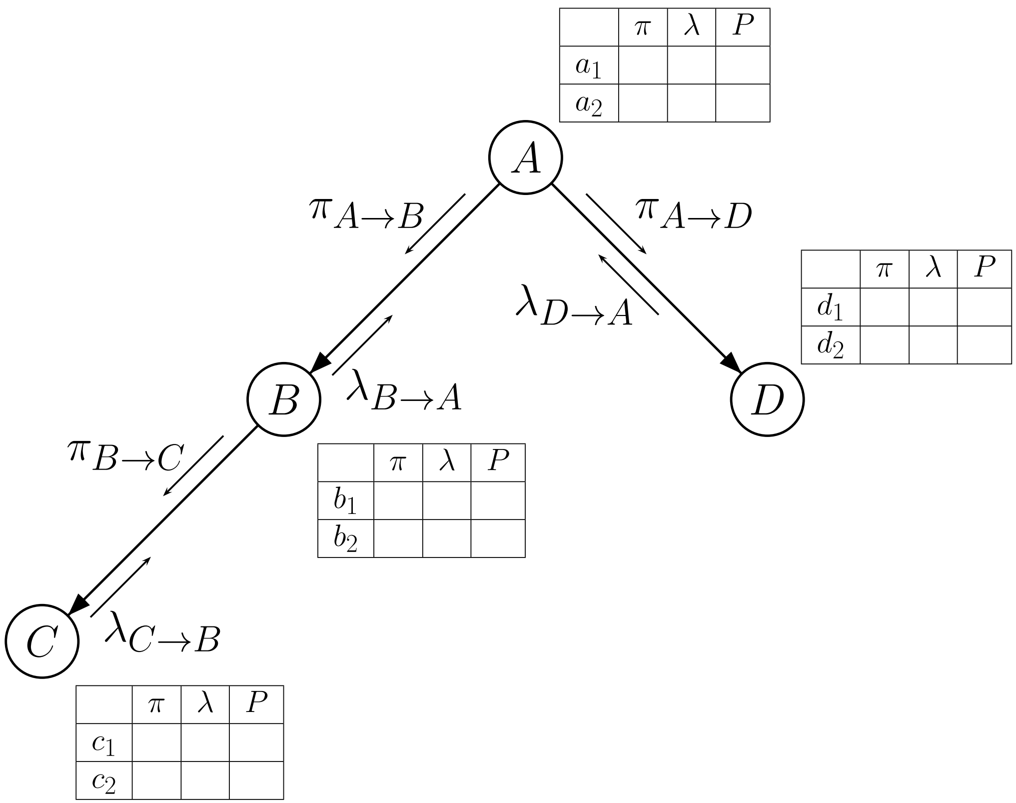
Let $B \in V$ be a non-root node in (V, E) and A the parent node of B. Further let $b \in \mathrm{dom}(B)$ be a value of B's domain.

$$\pi(b) \;=\; \mu_{B,W} \;\cdot \sum_{a \in \mathrm{dom}(A)} P(b \mid a) \cdot \pi_{A \to B}(a)$$

Let $A \notin W$ a non-instantiated attribute and $P(V) > 0$.

$$\begin{aligned}
\pi_{A \to B}(a_j) \;&=\; \pi(a_j) \;\cdot \prod_{C \in s(A) \setminus \{B\}} \lambda_{C \to A}(a_j) \\
&=\; \tau_{B,W} \cdot \frac{P(a_j \mid W)}{\lambda_{B \to A}(a_j)}
\end{aligned}$$

Belief Tree:



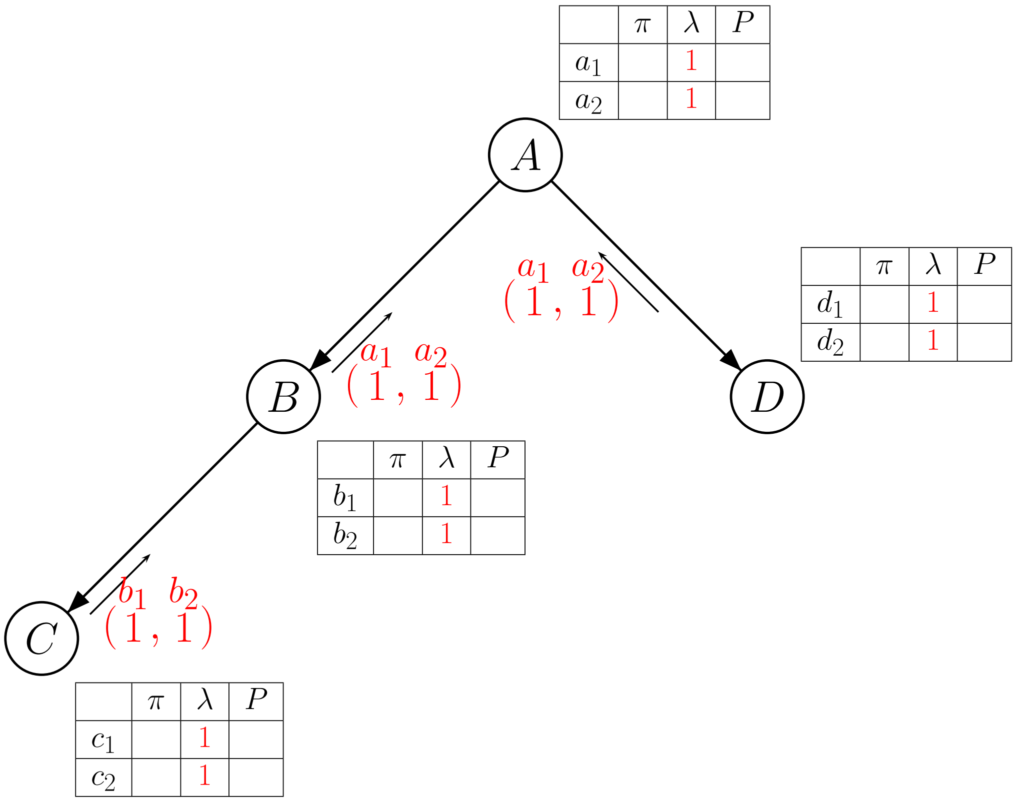
Parameters:

$$P(a_1) = 0.1 \quad P(b_1 \mid a_1) = 0.7$$
$$P(b_1 \mid a_2) = 0.2$$
$$P(d_1 \mid a_1) = 0.8 \quad P(c_1 \mid b_1) = 0.4$$
$$P(d_1 \mid a_2) = 0.4 \quad P(c_1 \mid b_2) = 0.001$$

Desired:

$$\forall X \in \{A, B, C, D\} : \; P(X \mid \emptyset) = ?$$

Belief Tree:



	π	λ	P
a_1		1	
a_2		1	

	π	λ	P
b_1		1	
b_2		1	

	π	λ	P
c_1		1	
c_2		1	

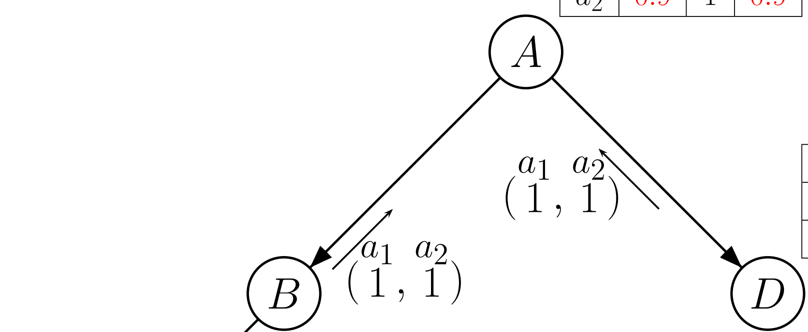
	π	λ	P
d_1		1	
d_2		1	

Initialization Phase:

Set all λ-messages and λ-values to 1.

Propagation in Belief Trees (3)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1		1	
d_2		1	

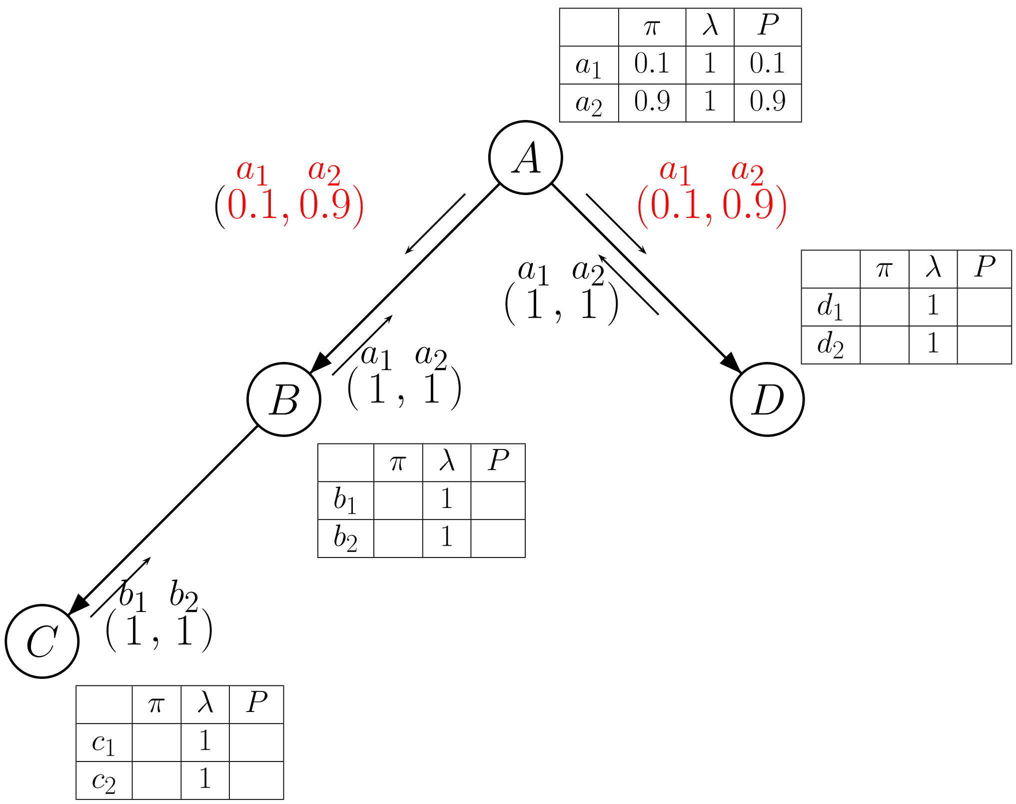
	π	λ	P
b_1		1	
b_2		1	

	π	λ	P
c_1		1	
c_2		1	

Initialization Phase:

Set all λ-messages and λ-values to 1.
$\pi(a_1) = P(a_1)$ and
$\pi(a_2) = P(a_2)$

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
b_1		1	
b_2		1	

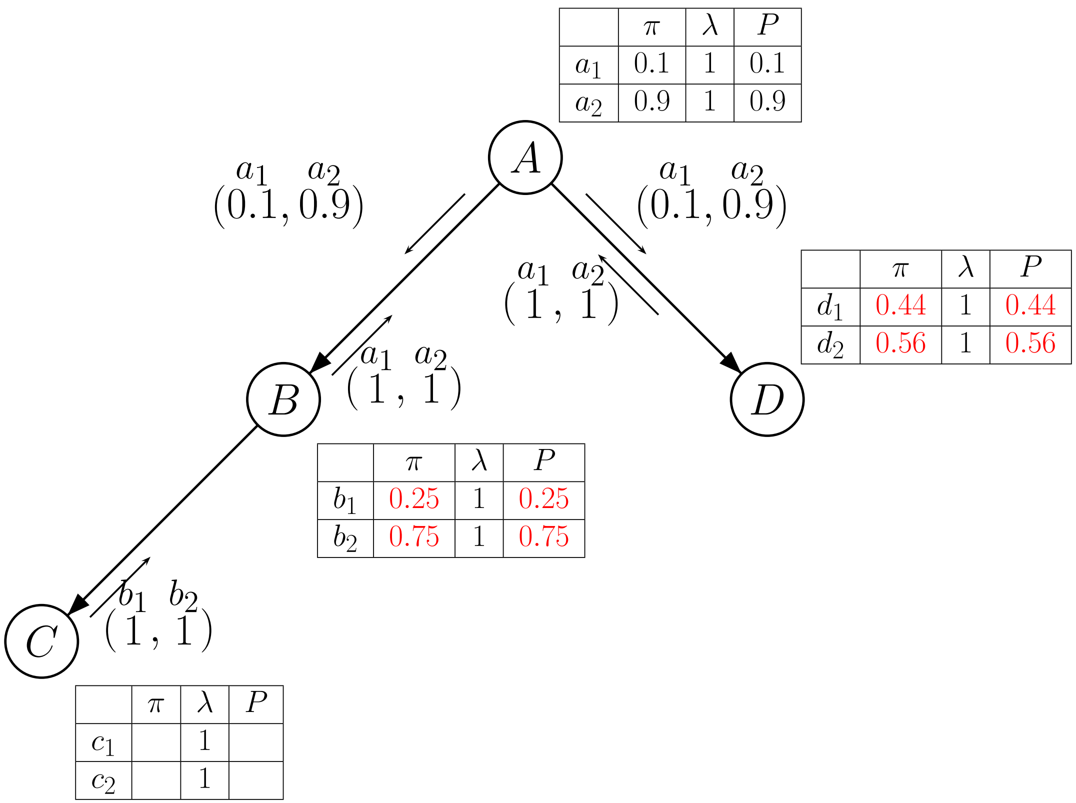
	π	λ	P
c_1		1	
c_2		1	

	π	λ	P
d_1		1	
d_2		1	

Initialization Phase:

Set all λ-messages and λ-values to 1.
$\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
A sends π-messages to B and D.

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1	0.44	1	0.44
d_2	0.56	1	0.56

	π	λ	P
b_1	0.25	1	0.25
b_2	0.75	1	0.75

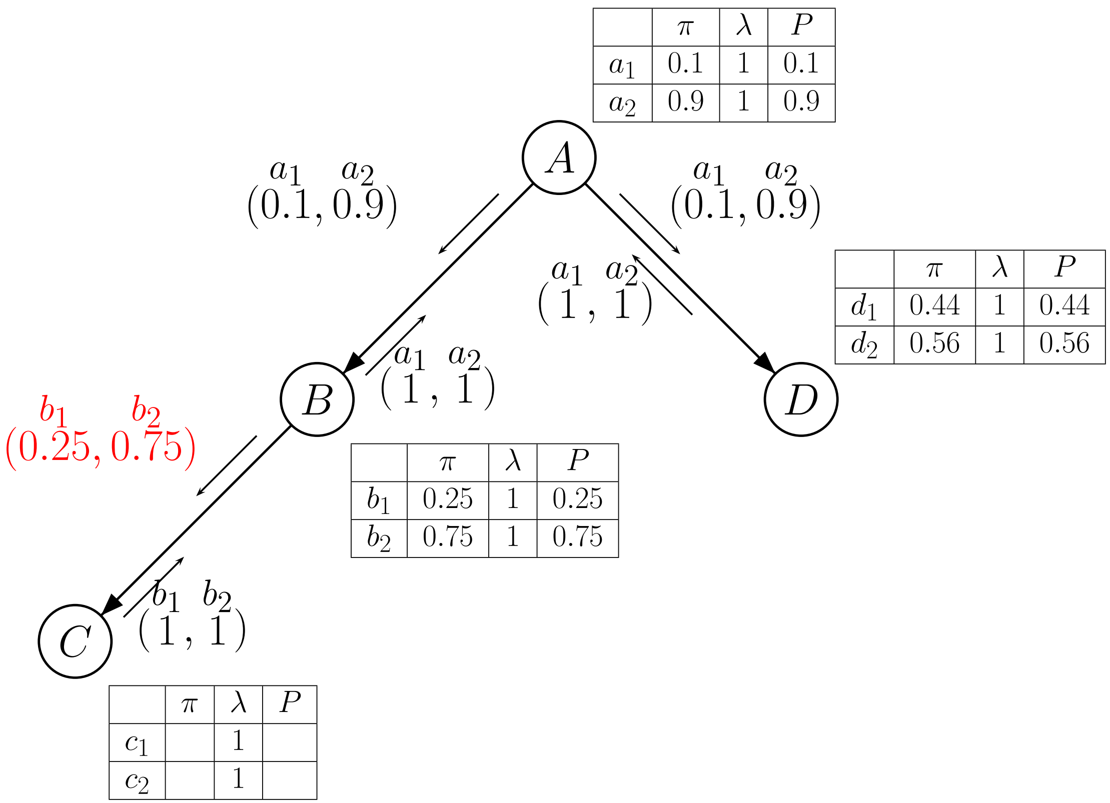
	π	λ	P
c_1		1	
c_2		1	

Initialization Phase:

Set all λ-messages and λ-values to 1.
$\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
A sends π-messages to B and D.
B and D update their π-values.

Propagation in Belief Trees (6)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1	0.44	1	0.44
d_2	0.56	1	0.56

	π	λ	P
b_1	0.25	1	0.25
b_2	0.75	1	0.75

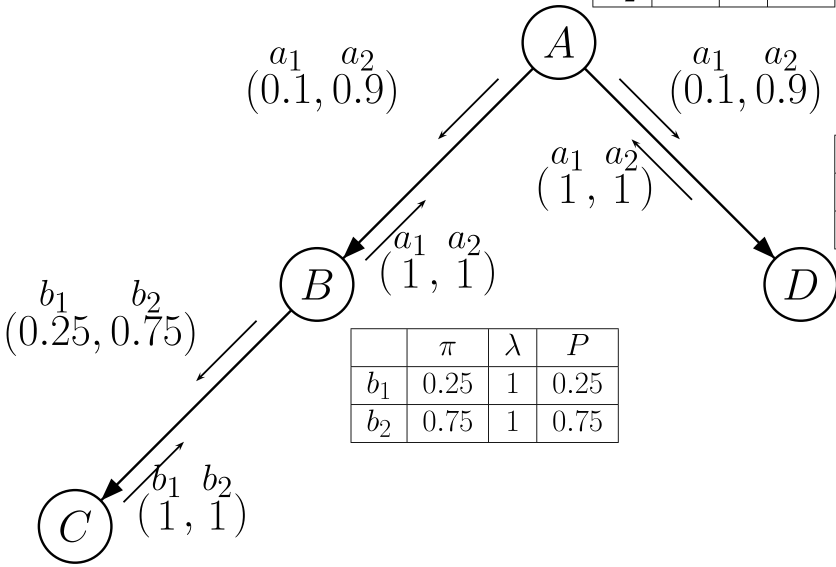
	π	λ	P
c_1		1	
c_2		1	

Initialization Phase:

Set all λ-messages and λ-values to 1.
$\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
A sends π-messages to B and D.
B and D update their π-values.
B sends π-message to C.

Propagation in Belief Trees (7)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1	0.44	1	0.44
d_2	0.56	1	0.56

	π	λ	P
b_1	0.25	1	0.25
b_2	0.75	1	0.75

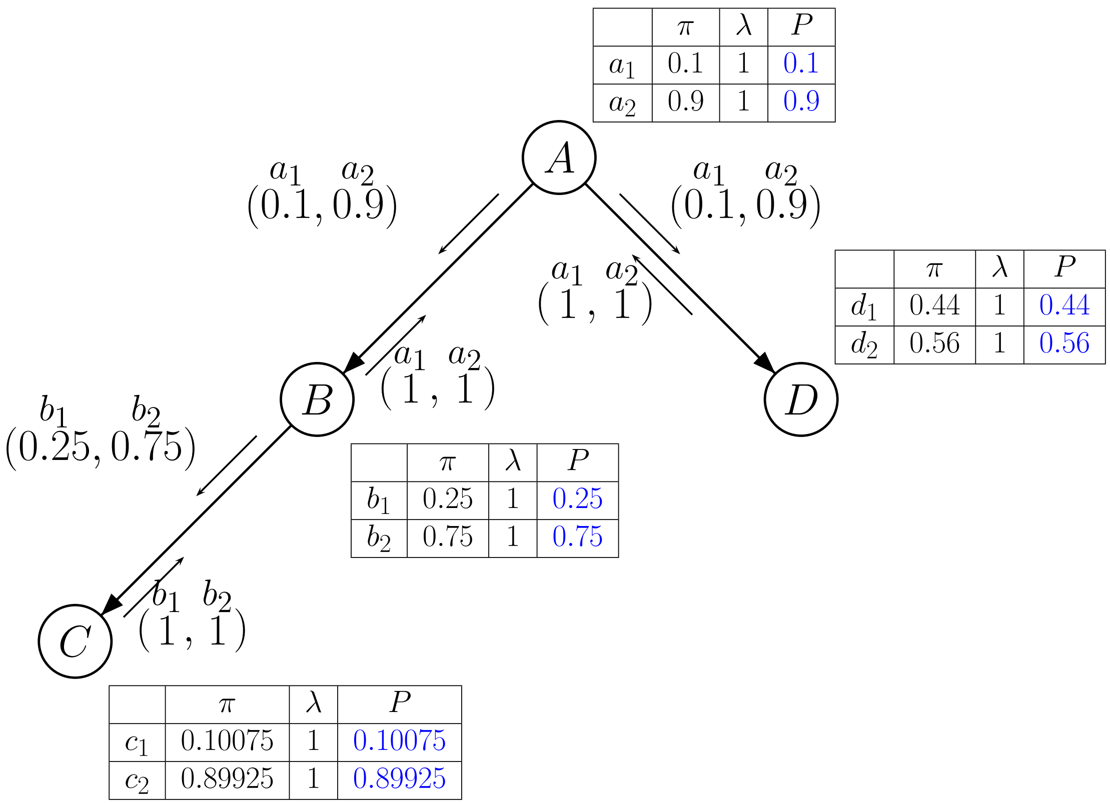
	π	λ	P
c_1	0.10075	1	0.10075
c_2	0.89925	1	0.89925

Initialization Phase:

Set all λ-messages and λ-values to 1.
$\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
A sends π-messages to B and D.
B and D update their π-values.
B sends π-message to C.
C updates it π-value.

Propagation in Belief Trees (8)

Belief Tree:



	π	λ	P
a_1	0.1	1	0.1
a_2	0.9	1	0.9

	π	λ	P
d_1	0.44	1	0.44
d_2	0.56	1	0.56

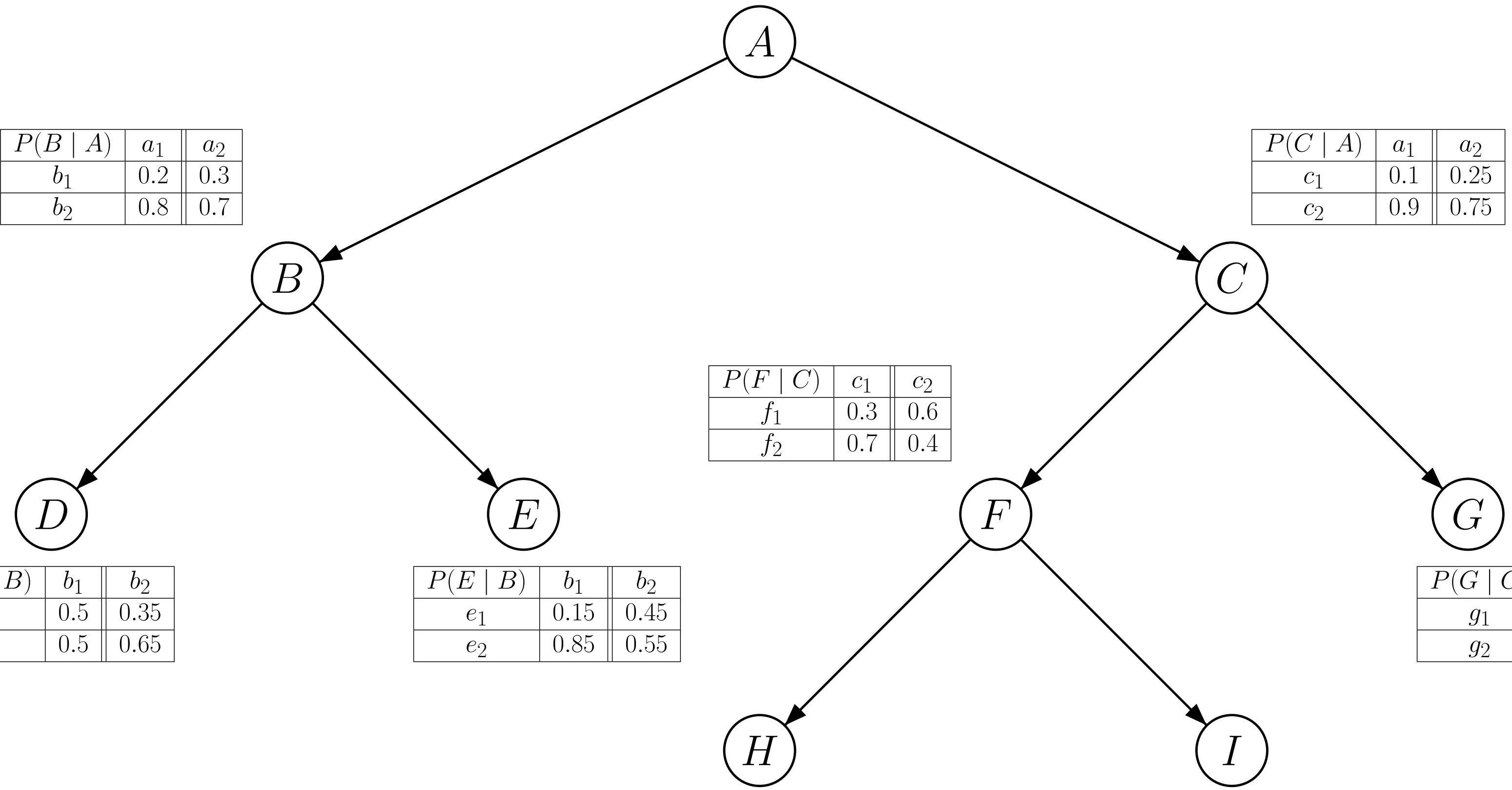
	π	λ	P
b_1	0.25	1	0.25
b_2	0.75	1	0.75

	π	λ	P
c_1	0.10075	1	0.10075
c_2	0.89925	1	0.89925

Initialization Phase:

Set all λ-messages and λ-values to 1.
$\pi(a_1) = P(a_1)$ and $\pi(a_2) = P(a_2)$.
A sends π-messages to B and D.
B and D update their π-values.
B sends π-message to C.
C updates it π-value.
Initialization finished.

Larger Network (1): Parameters



$P(A)$	$\emptyset$
a_1	0.4
a_2	0.6

$P(B \mid A)$	a_1	a_2
b_1	0.2	0.3
b_2	0.8	0.7

$P(C \mid A)$	a_1	a_2
c_1	0.1	0.25
c_2	0.9	0.75

$P(F \mid C)$	c_1	c_2
f_1	0.3	0.6
f_2	0.7	0.4

$P(D \mid B)$	b_1	b_2
d_1	0.5	0.35
d_2	0.5	0.65

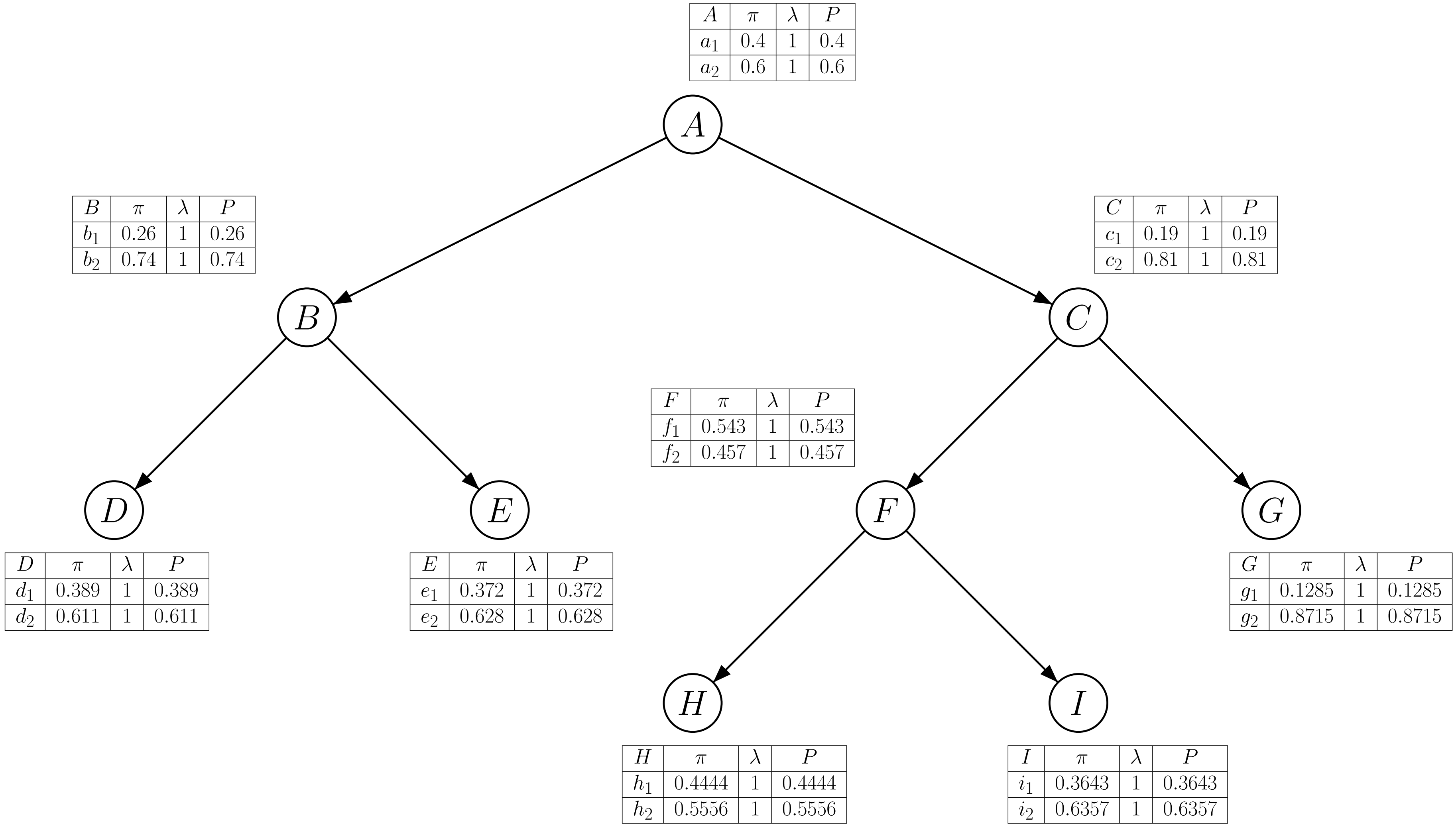
$P(E \mid B)$	b_1	b_2
e_1	0.15	0.45
e_2	0.85	0.55

$P(G \mid C)$	c_1	c_2
g_1	0.25	0.1
g_2	0.75	0.9

$P(H \mid F)$	f_1	f_2
h_1	0.65	0.2
h_2	0.35	0.8

$P(I \mid F)$	f_1	f_2
i_1	0.25	0.5
i_2	0.75	0.5

Larger Network (2): After Initialization



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	1	0.26
b_2	0.74	1	0.74

C	π	λ	P
c_1	0.19	1	0.19
c_2	0.81	1	0.81

F	π	λ	P
f_1	0.543	1	0.543
f_2	0.457	1	0.457

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

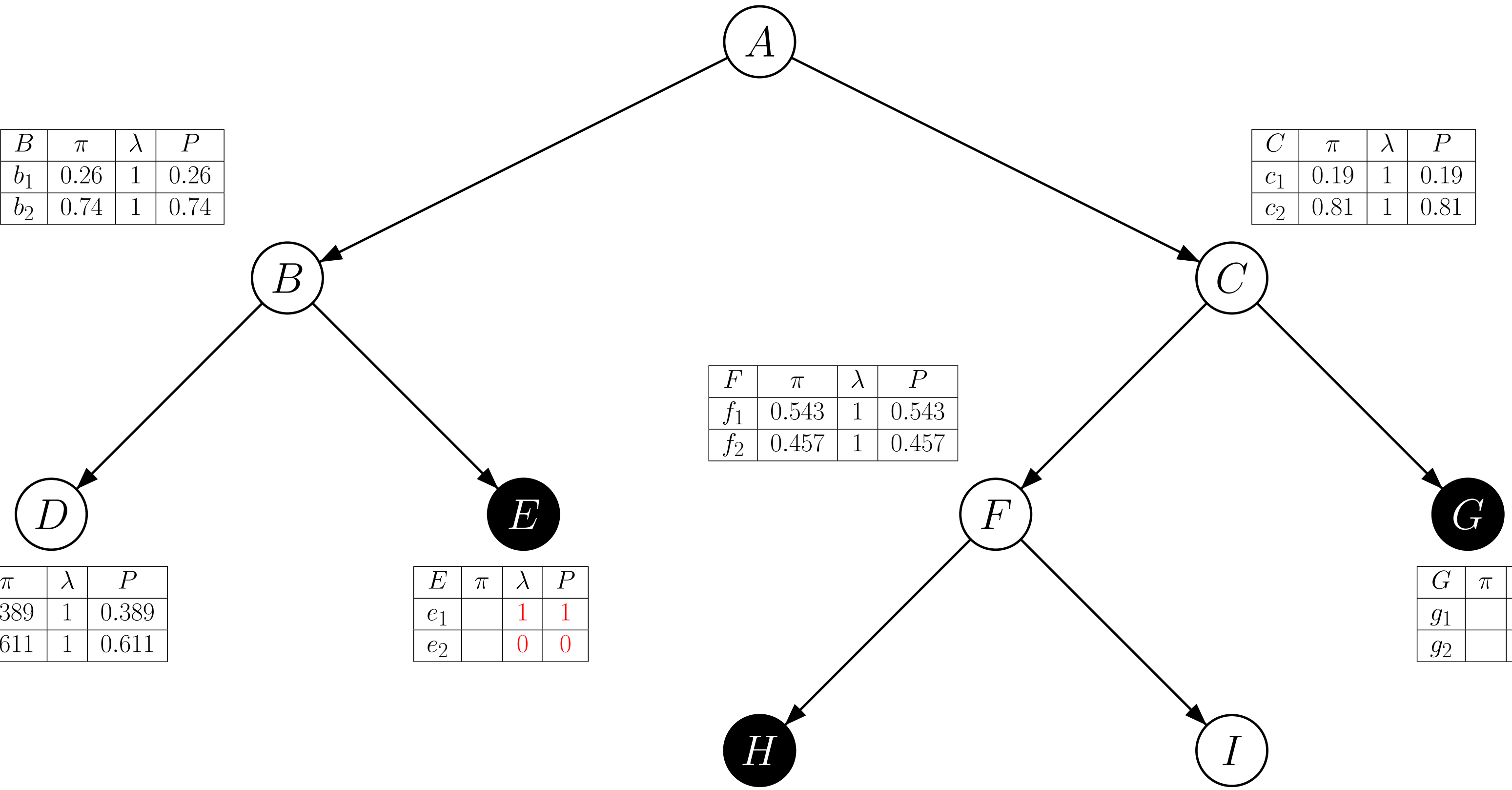
E	π	λ	P
e_1	0.372	1	0.372
e_2	0.628	1	0.628

G	π	λ	P
g_1	0.1285	1	0.1285
g_2	0.8715	1	0.8715

H	π	λ	P
h_1	0.4444	1	0.4444
h_2	0.5556	1	0.5556

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (3): Set Evidence e_1, g_1, h_1



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	1	0.26
b_2	0.74	1	0.74

C	π	λ	P
c_1	0.19	1	0.19
c_2	0.81	1	0.81

F	π	λ	P
f_1	0.543	1	0.543
f_2	0.457	1	0.457

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

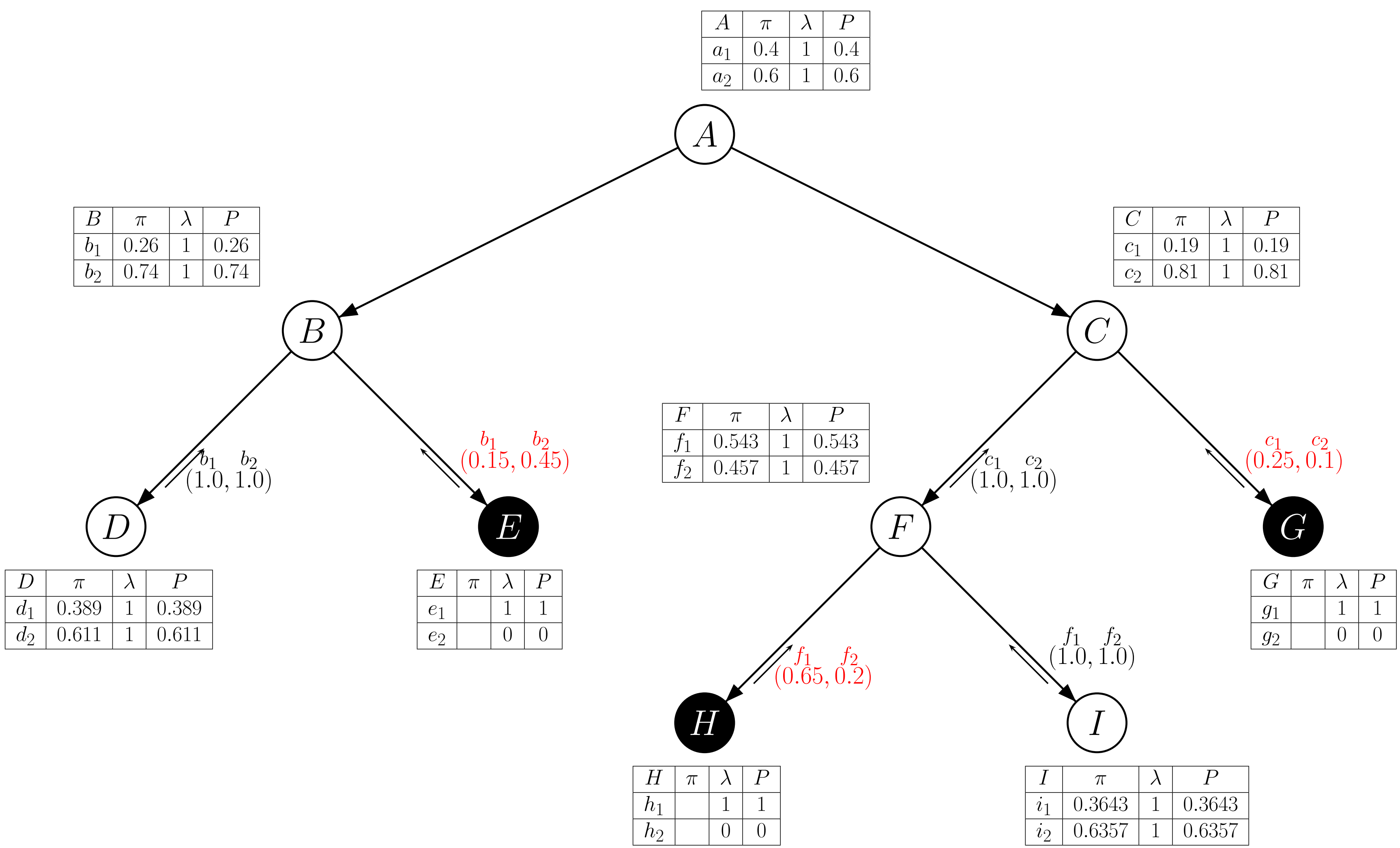
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (4): Propagate Evidence



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	1	0.26
b_2	0.74	1	0.74

C	π	λ	P
c_1	0.19	1	0.19
c_2	0.81	1	0.81

F	π	λ	P
f_1	0.543	1	0.543
f_2	0.457	1	0.457

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

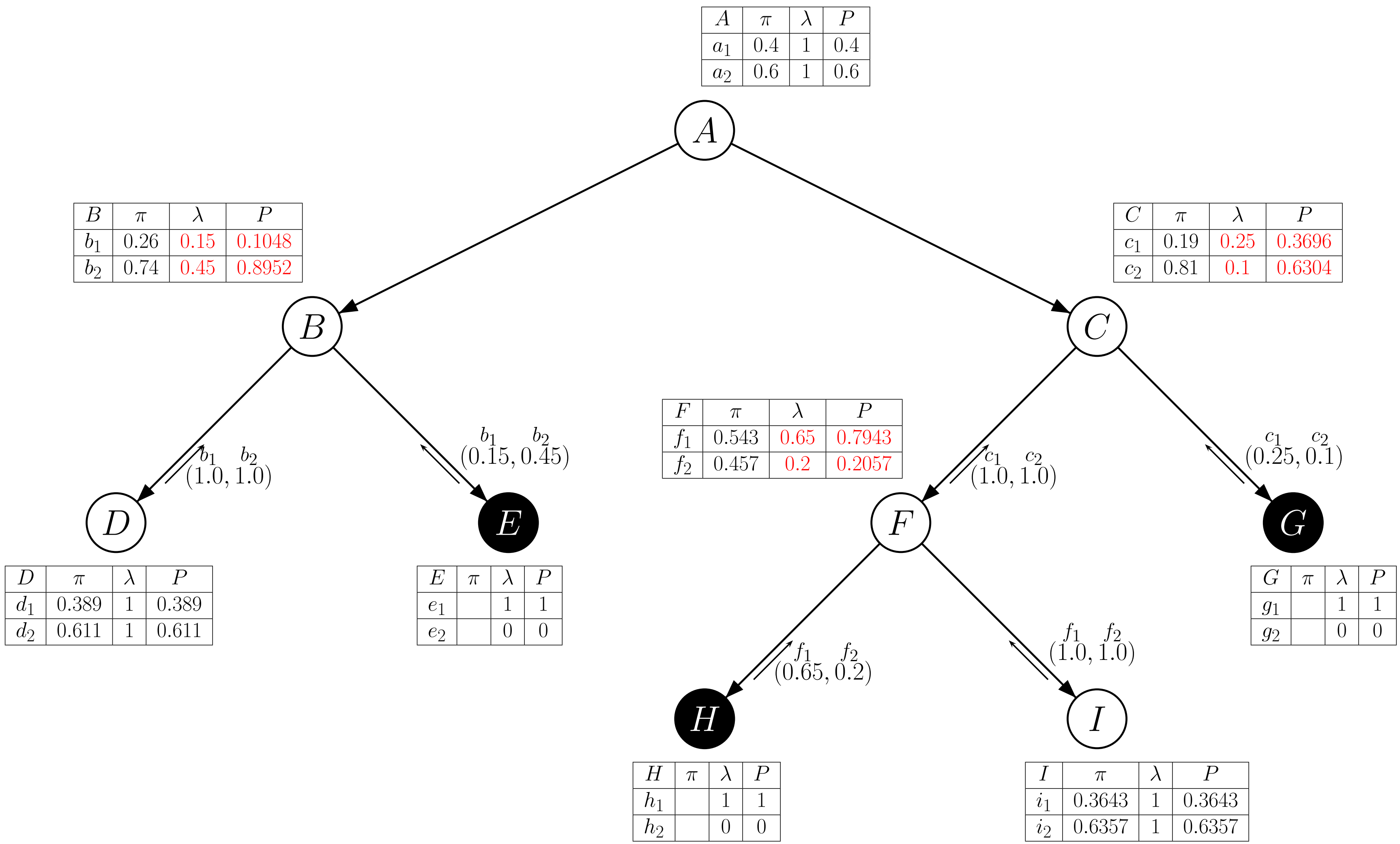
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (5): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.25	0.3696
c_2	0.81	0.1	0.6304

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

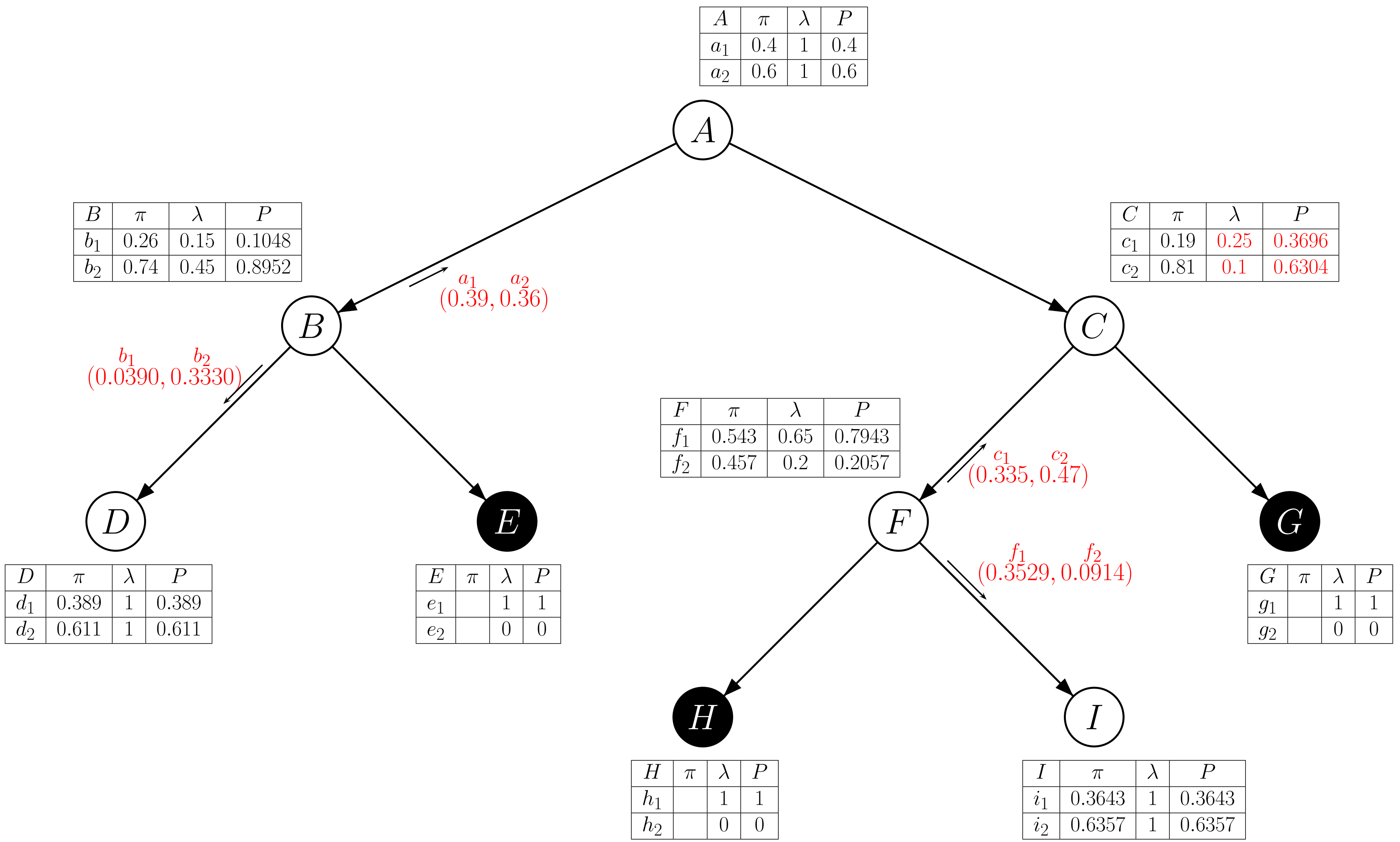
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (6): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	1	0.4
a_2	0.6	1	0.6

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.25	0.3696
c_2	0.81	0.1	0.6304

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.389	1	0.389
d_2	0.611	1	0.611

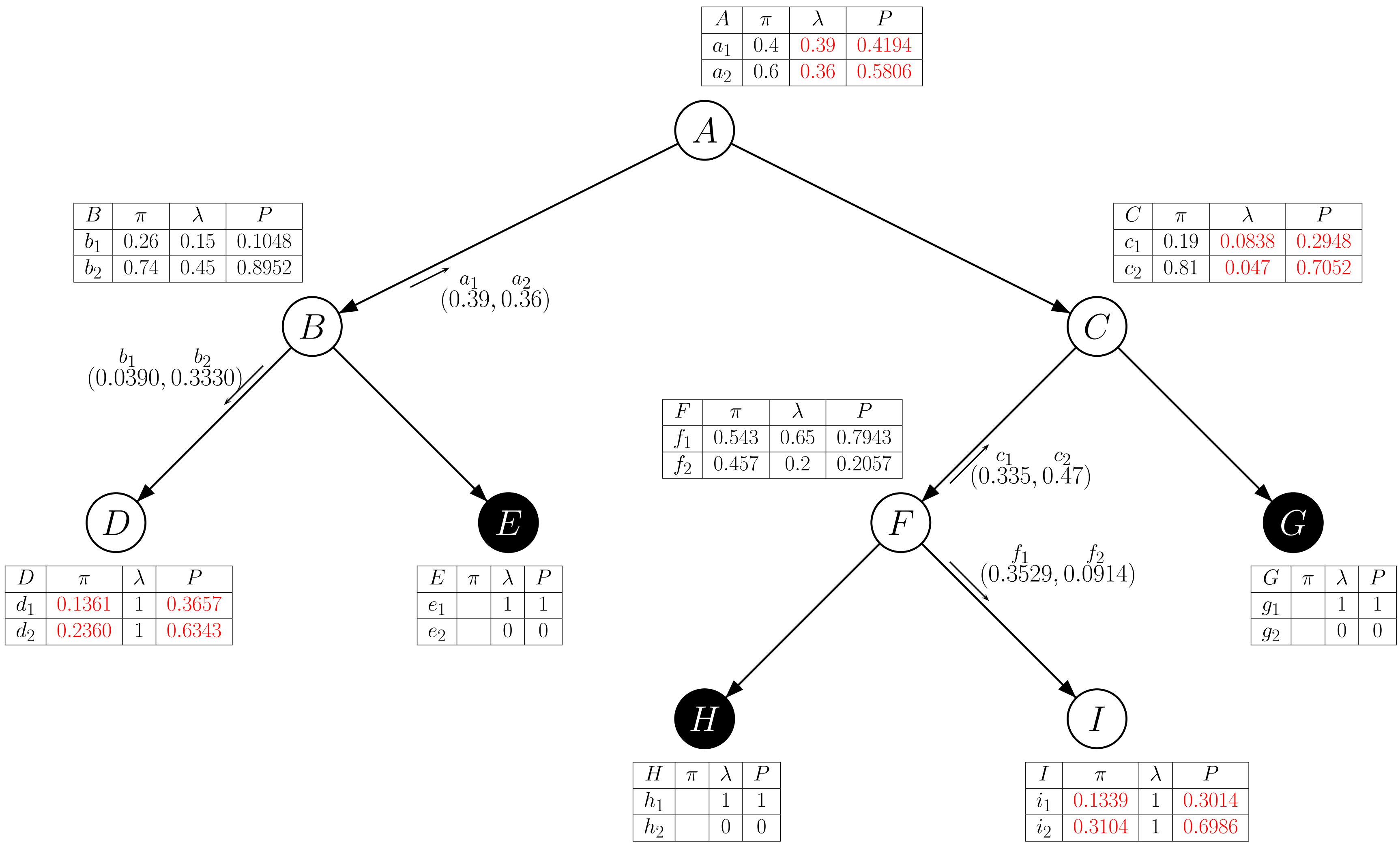
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.3643	1	0.3643
i_2	0.6357	1	0.6357

Larger Network (7): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.39	0.4194
a_2	0.6	0.36	0.5806

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.1361	1	0.3657
d_2	0.2360	1	0.6343

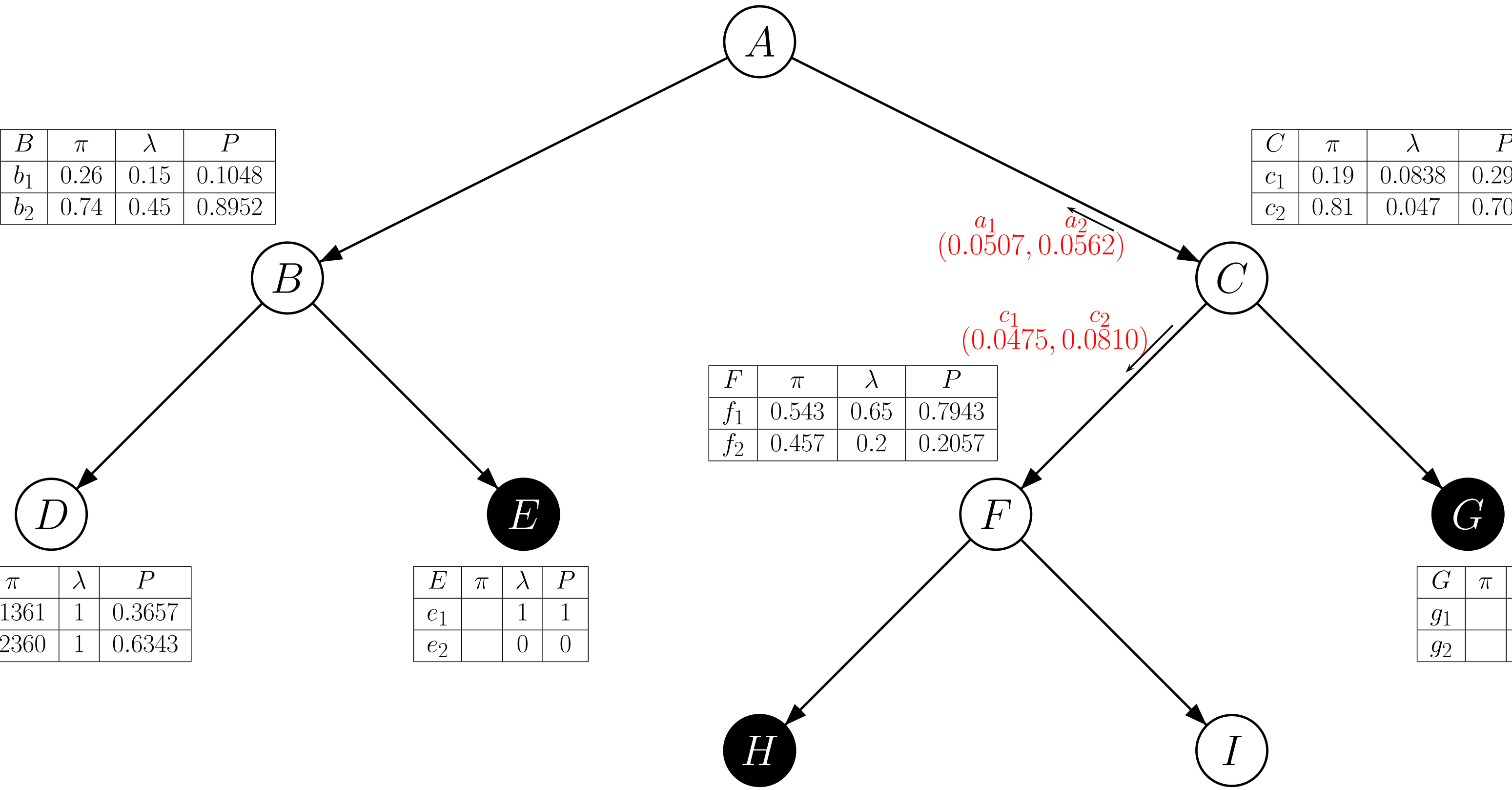
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.1339	1	0.3014
i_2	0.3104	1	0.6986

Larger Network (8): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.39	0.4194
a_2	0.6	0.36	0.5806

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.543	0.65	0.7943
f_2	0.457	0.2	0.2057

D	π	λ	P
d_1	0.1361	1	0.3657
d_2	0.2360	1	0.6343

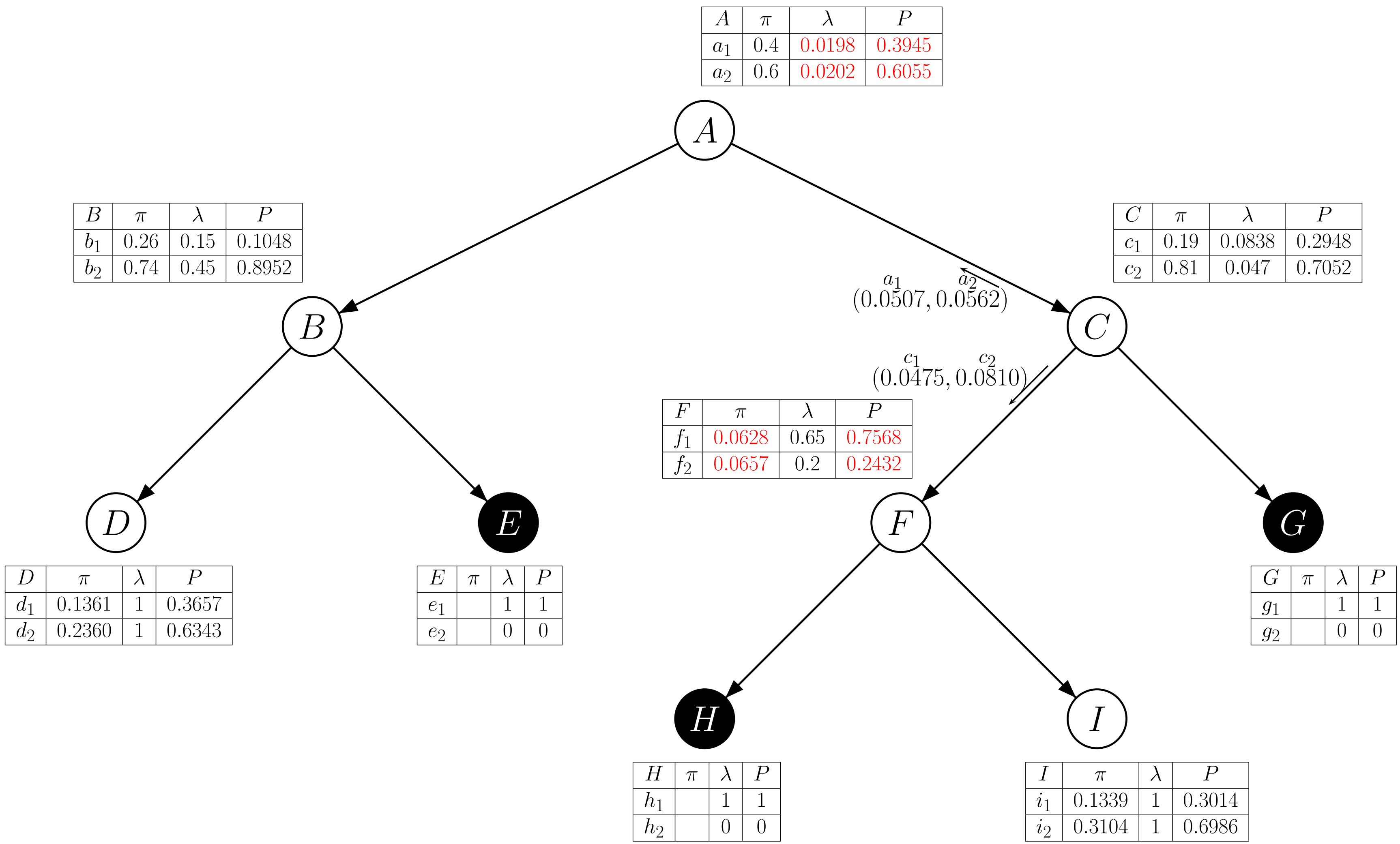
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.1339	1	0.3014
i_2	0.3104	1	0.6986

Larger Network (9): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.0628	0.65	0.7568
f_2	0.0657	0.2	0.2432

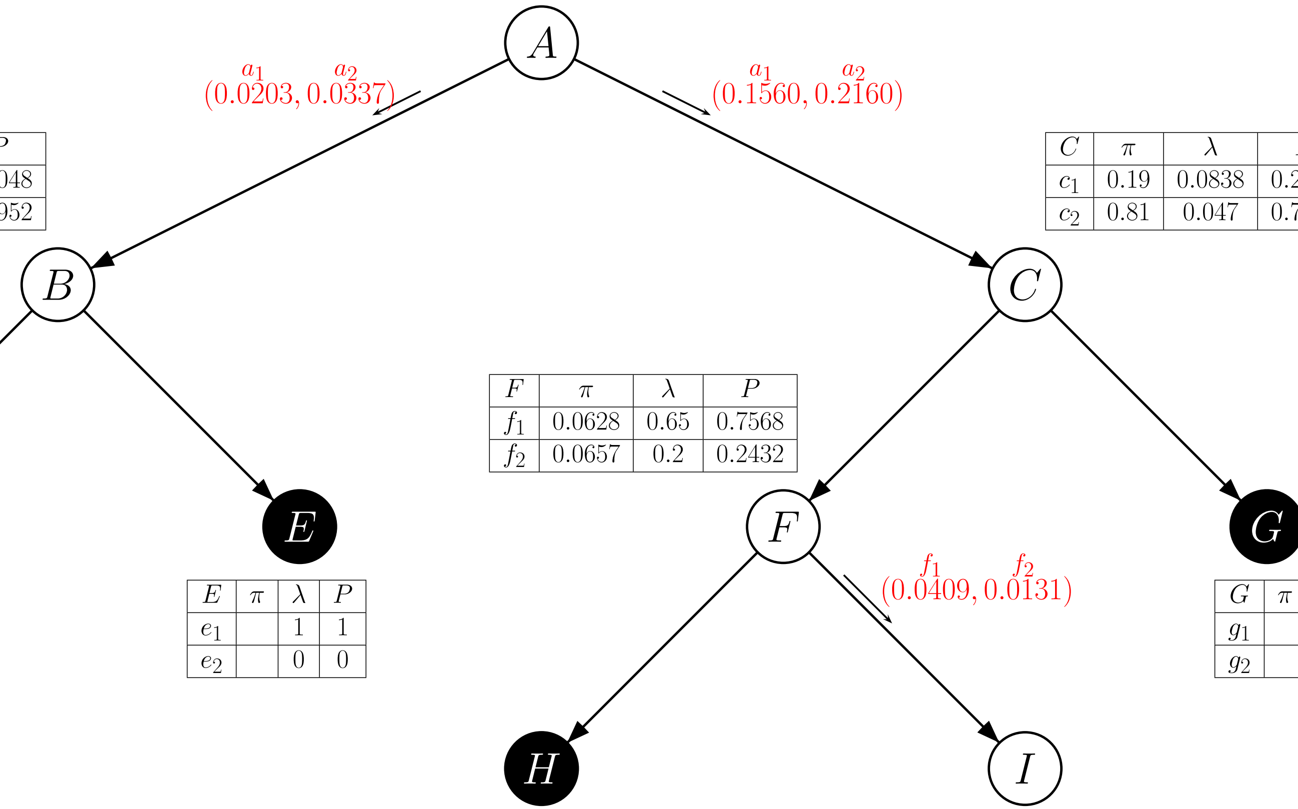
D	π	λ	P
d_1	0.1361	1	0.3657
d_2	0.2360	1	0.6343

E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.1339	1	0.3014
i_2	0.3104	1	0.6986



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.26	0.15	0.1048
b_2	0.74	0.45	0.8952

C	π	λ	P
c_1	0.19	0.0838	0.2948
c_2	0.81	0.047	0.7052

F	π	λ	P
f_1	0.0628	0.65	0.7568
f_2	0.0657	0.2	0.2432

D	π	λ	P
d_1	0.1361	1	0.3657
d_2	0.2360	1	0.6343

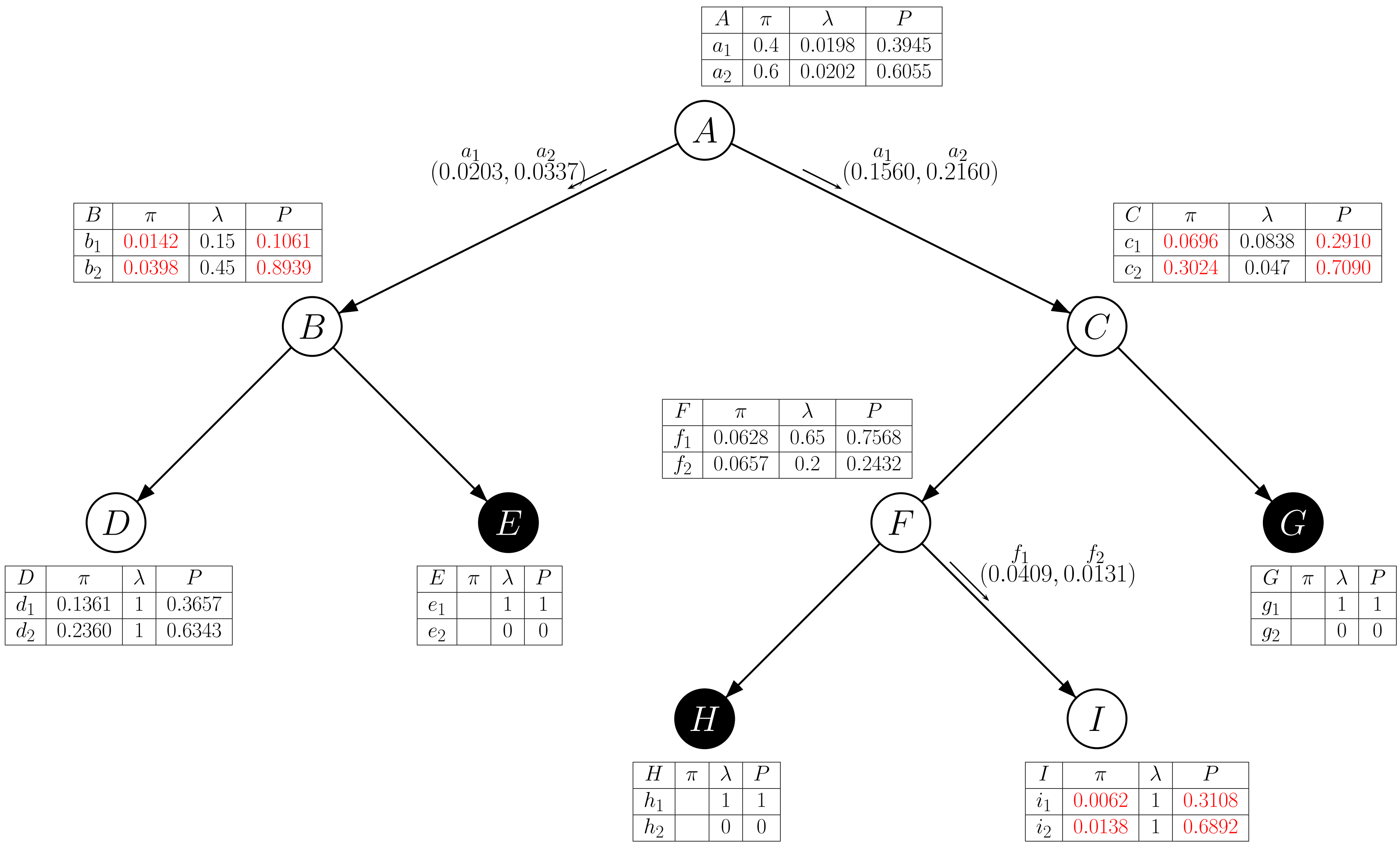
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.1339	1	0.3014
i_2	0.3104	1	0.6986

Larger Network (11): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.0142	0.15	0.1061
b_2	0.0398	0.45	0.8939

C	π	λ	P
c_1	0.0696	0.0838	0.2910
c_2	0.3024	0.047	0.7090

F	π	λ	P
f_1	0.0628	0.65	0.7568
f_2	0.0657	0.2	0.2432

D	π	λ	P
d_1	0.1361	1	0.3657
d_2	0.2360	1	0.6343

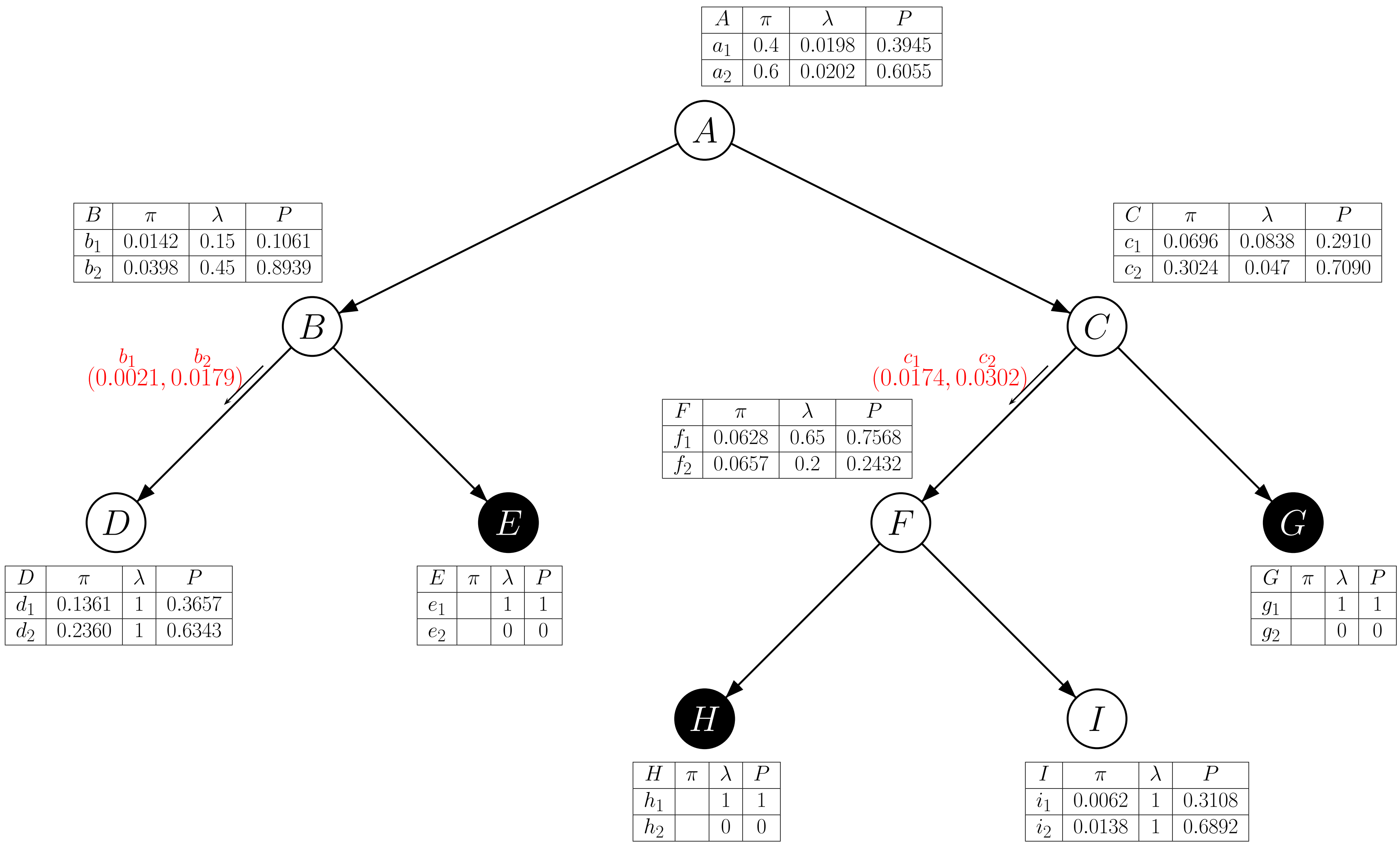
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.0062	1	0.3108
i_2	0.0138	1	0.6892

Larger Network (12): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.0142	0.15	0.1061
b_2	0.0398	0.45	0.8939

C	π	λ	P
c_1	0.0696	0.0838	0.2910
c_2	0.3024	0.047	0.7090

F	π	λ	P
f_1	0.0628	0.65	0.7568
f_2	0.0657	0.2	0.2432

D	π	λ	P
d_1	0.1361	1	0.3657
d_2	0.2360	1	0.6343

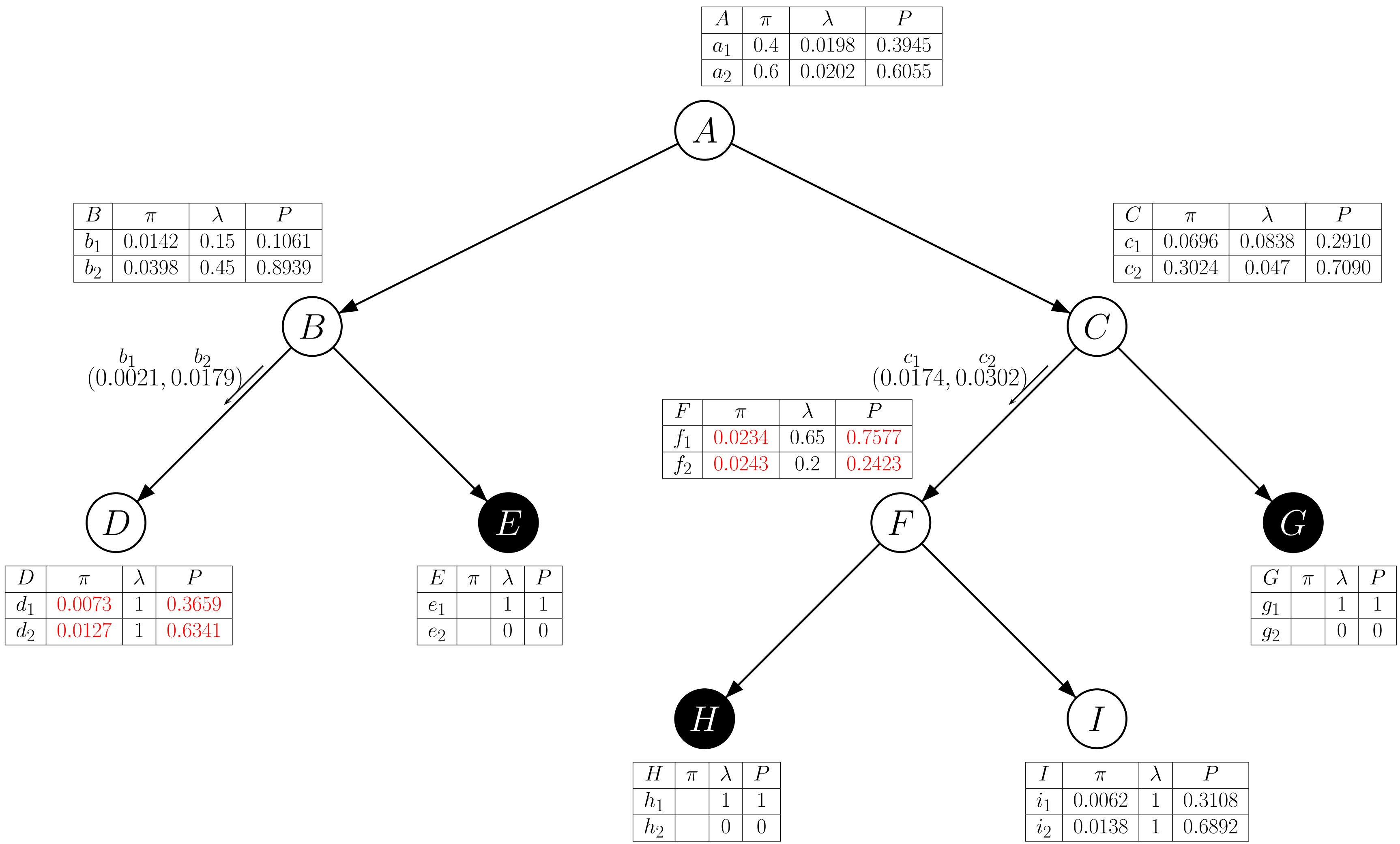
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.0062	1	0.3108
i_2	0.0138	1	0.6892

Larger Network (13): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.0142	0.15	0.1061
b_2	0.0398	0.45	0.8939

C	π	λ	P
c_1	0.0696	0.0838	0.2910
c_2	0.3024	0.047	0.7090

F	π	λ	P
f_1	0.0234	0.65	0.7577
f_2	0.0243	0.2	0.2423

D	π	λ	P
d_1	0.0073	1	0.3659
d_2	0.0127	1	0.6341

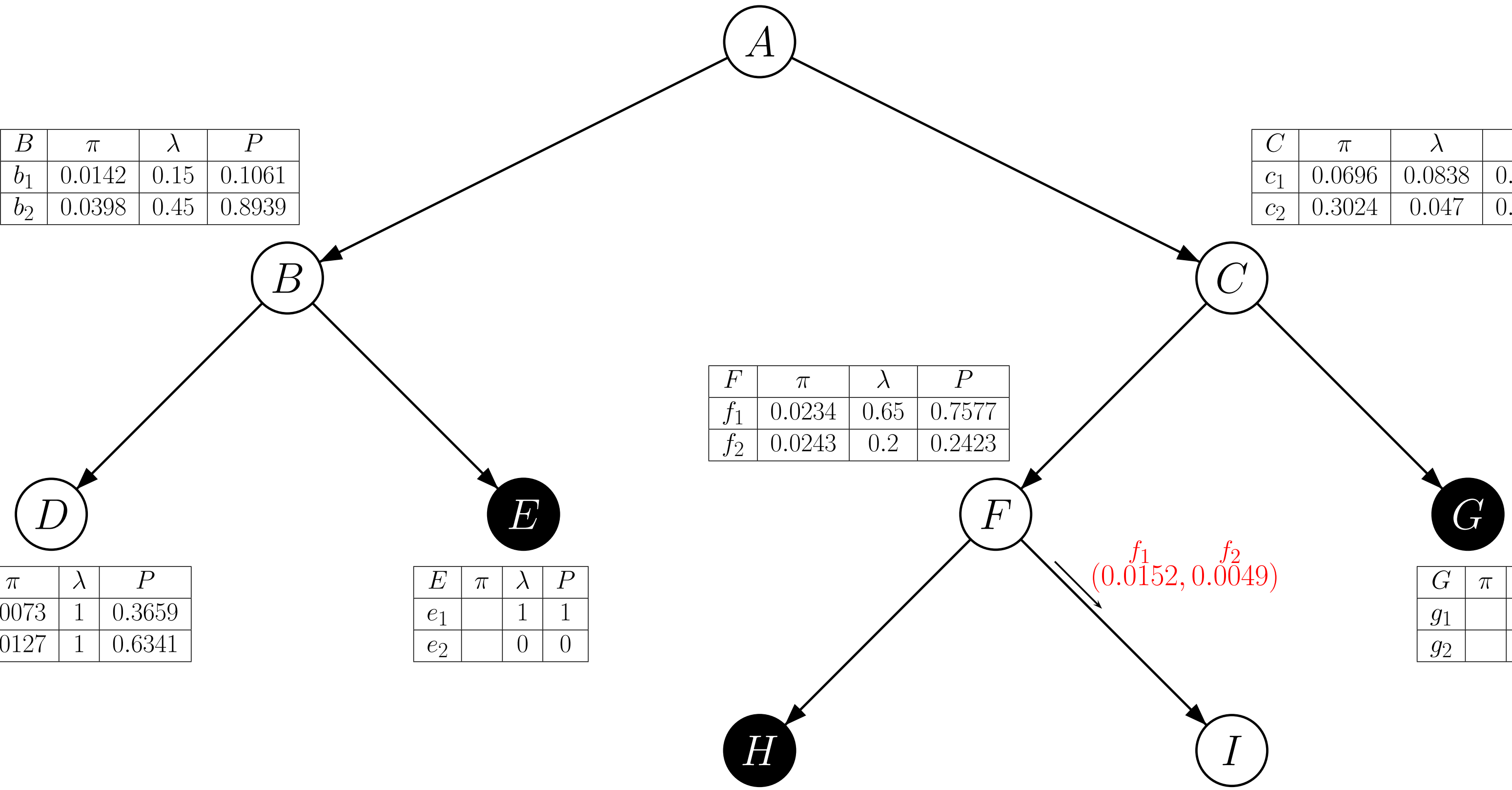
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.0062	1	0.3108
i_2	0.0138	1	0.6892

Larger Network (14): Propagate Evidence, cont.



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.0142	0.15	0.1061
b_2	0.0398	0.45	0.8939

C	π	λ	P
c_1	0.0696	0.0838	0.2910
c_2	0.3024	0.047	0.7090

F	π	λ	P
f_1	0.0234	0.65	0.7577
f_2	0.0243	0.2	0.2423

D	π	λ	P
d_1	0.0073	1	0.3659
d_2	0.0127	1	0.6341

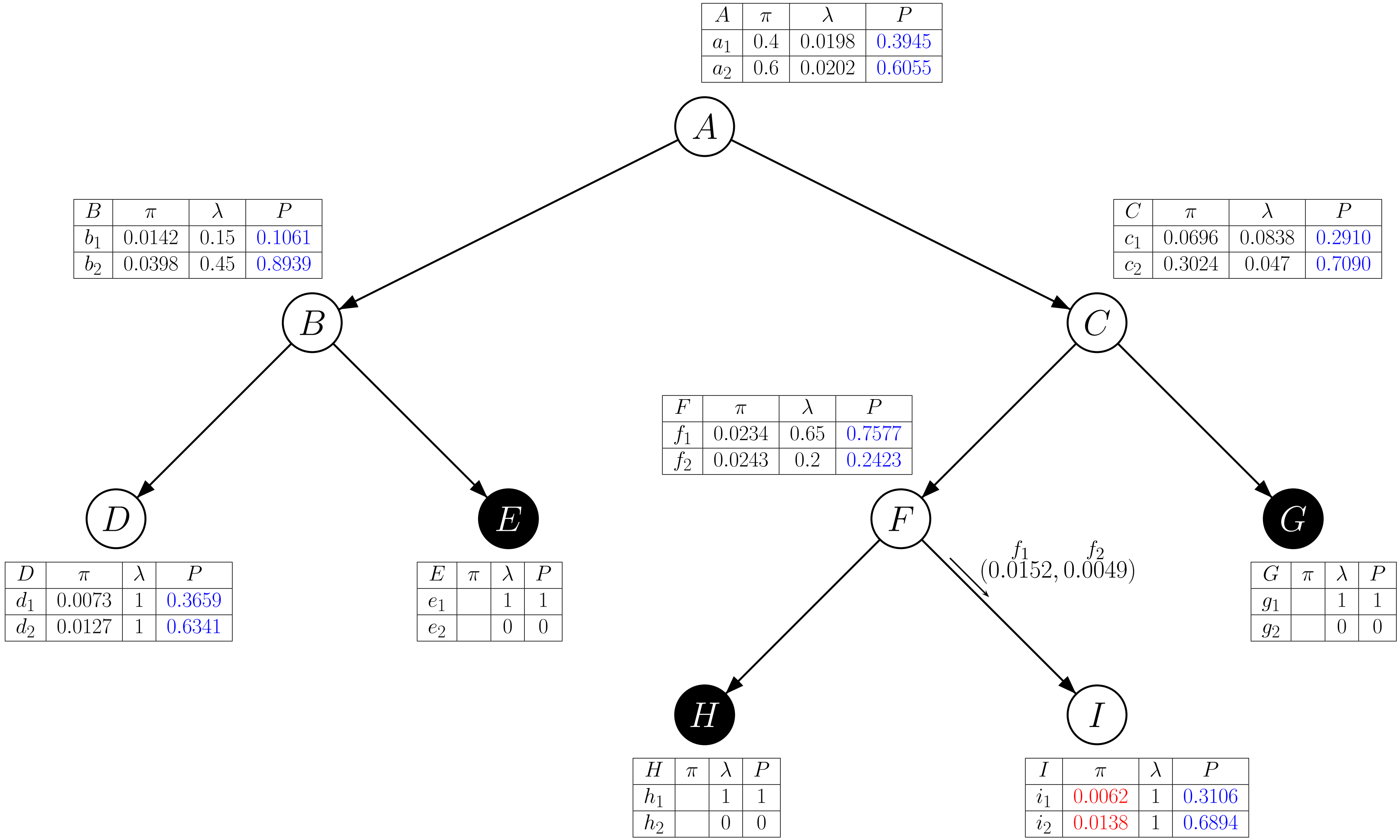
E	π	λ	P
e_1		1	1
e_2		0	0

G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.0062	1	0.3108
i_2	0.0138	1	0.6892

Larger Network (15): Finished



A	π	λ	P
a_1	0.4	0.0198	0.3945
a_2	0.6	0.0202	0.6055

B	π	λ	P
b_1	0.0142	0.15	0.1061
b_2	0.0398	0.45	0.8939

C	π	λ	P
c_1	0.0696	0.0838	0.2910
c_2	0.3024	0.047	0.7090

F	π	λ	P
f_1	0.0234	0.65	0.7577
f_2	0.0243	0.2	0.2423

D	π	λ	P
d_1	0.0073	1	0.3659
d_2	0.0127	1	0.6341

E	π	λ	P
e_1		1	1
e_2		0	0

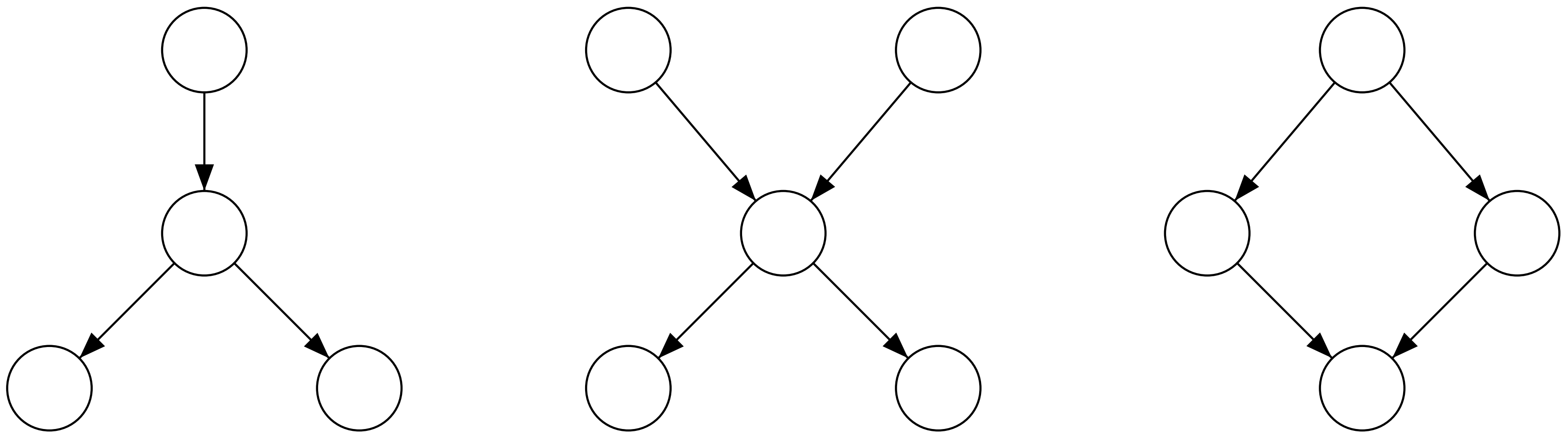
G	π	λ	P
g_1		1	1
g_2		0	0

H	π	λ	P
h_1		1	1
h_2		0	0

I	π	λ	P
i_1	0.0062	1	0.3106
i_2	0.0138	1	0.6894

Inference in Bayes-Networks and Markov-Networks

Problems



The propagation algorithm as presented can only deal with *trees*.

Can be extended to *polytrees* (i. e. singly connected graphs with multiple parents per node).

However, it cannot handle networks that contain loops!

Idea

Main Objectives:

Transform the cyclic directed graph into a secondary structure without cycles.

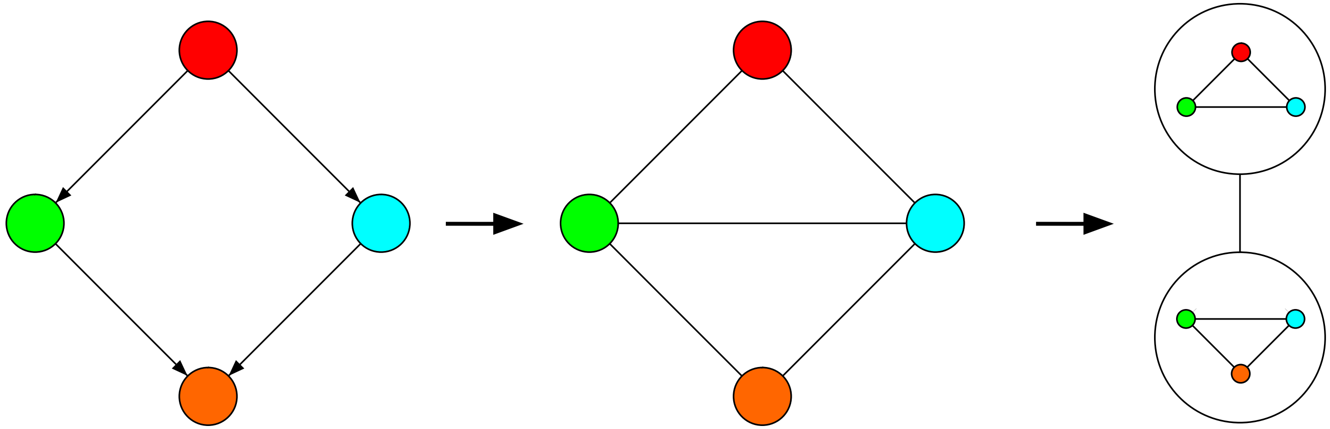
Find a decomposition of the underlying joint distribution.

Task:

Combine nodes of the original (primary) graph structure.

These groups form the nodes of a secondary structure.

Find a transformation that yields tree structure.



Idea (2)

Secondary Structure:

We will generate an undirected graph mimicking (some of) the conditional independence statements of the cyclic directed graph.

Maximal cliques are identified and form the nodes of the secondary structure.

Specify a so-called potential function for every clique such that the product of all potentials yields the initial joint distribution.

In order to propagate evidence, create a **tree** from the clique nodes such that the following property is satisfied:

> If two cliques have some attributes in common, then these attributes have to be contained in every clique of the path connecting the two cliques. (called the **running intersection property**, **RIP**)

Justification:

Tree: Unique path of evidence propagation.

RIP: Update of an attribute reaches all cliques which contain it.

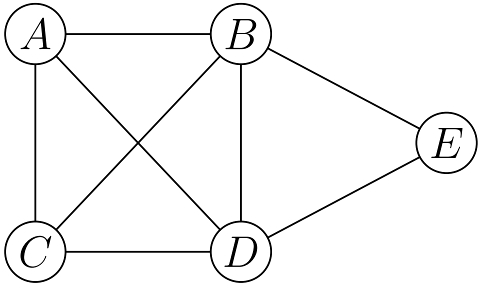
Complete Graph

An undirected Graph $G = (V, E)$ is called *complete*, if every pair of (distinct) nodes is connected by an edge.

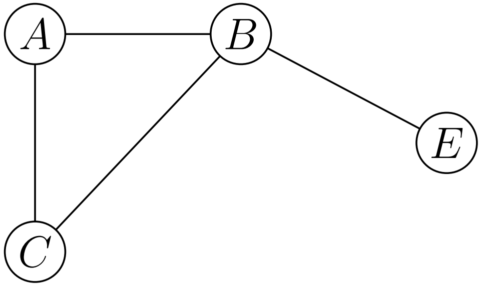
Induced Subgraph

Let $G = (V, E)$ be an undirected graph and $W \subseteq V$ a selection of nodes. Then, $G_W = (W, E_W)$ is called the *subgraph of G induced by W* with E_W being

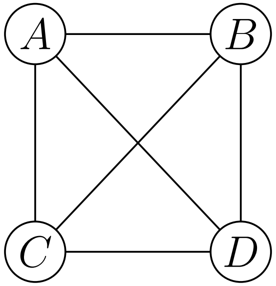
$$E_W = \{(u, v) \in E \mid u, v \in W\}.$$



Incomplete graph



Subgraph (W, E_W)
with $W = \{A, B, C, E\}$



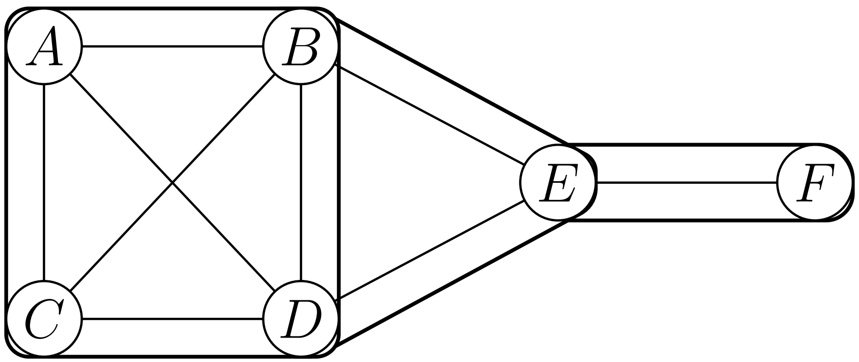
Complete (sub)graph

Complete Set, Clique

Let $G = (V, E)$ be an undirected graph. A set $W \subseteq V$ is called *complete* iff it induces a complete subgraph. It is further called a *clique*, iff W is maximal, i. e. it is not possible to add a node to W without violating the completeness condition.

a) W is complete $\Leftrightarrow$ W induces a complete subgraph

b) W is a clique $\Leftrightarrow$ W is complete and maximal



3 cliques

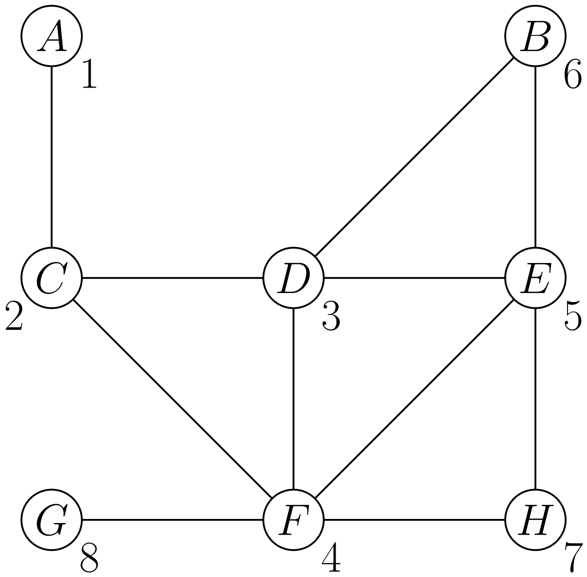
$$C_1 = \{A, B, C, D\}$$
$$C_2 = \{B, D, E\}$$
$$C_3 = \{E, F\}$$

Perfect Ordering

Let $G = (V, E)$ be an undirected graph with n nodes and $\alpha = \langle v_1, \ldots, v_n \rangle$ a total ordering on V. Then, α is called *perfect*, if the following sets

$$\mathrm{adj}(v_i) \cap \{v_1, \ldots, v_{i-1}\} \qquad i = 1, \ldots, n$$

are complete, where $\mathrm{adj}(v_i) = \{w \mid (v_i, w) \in E\}$ returns the adjacent nodes of v_i.



$\alpha = \langle A,\ C,\ D,\ F,\ E,\ B,\ H,\ G \rangle$

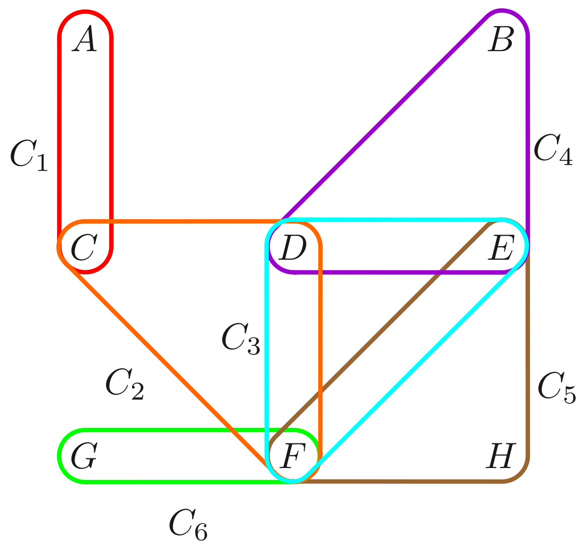
i	$\mathrm{adj}(v_i)$	$\mathrm{adj}(v_i) \cap \{v_1, \ldots, v_{i-1}\}$		
1	$\{C\}$	$\{C\} \cap \emptyset$	$= \emptyset$	complete
2	$\{A, D, F\}$	$\{A\} \cap \{A, D, F\}$	$= \{A\}$	complete
3	$\{C, B, E, F\}$	$\{A, C\} \cap \{C, B, E, F\}$	$= \{C\}$	complete
4	$\{G, C, D, E, H\}$	$\{A, C, D\} \cap \{G, C, D, E, H\}$	$= \{C, D\}$	complete
5	$\{B, D, F, H\}$	$\{A, C, D, F\} \cap \{B, D, F, H\}$	$= \{D, F\}$	complete
6	$\{D, E\}$	$\{A, C, D, F, E\} \cap \{D, E\}$	$= \{D, E\}$	complete
7	$\{F, E\}$	$\{A, C, D, F, E, B\} \cap \{F, E\}$	$= \{F, E\}$	complete
8	$\{F\}$	$\{A, C, D, F, E, B, H\} \cap \{F\}$	$= \{F\}$	complete

α is a perfect ordering

Running Intersection Property

Let $G = (V, E)$ be an undirected graph with p cliques. An ordering of these cliques has the *running intersection property (RIP)*, if for every $j > 1$ there exists an $i < j$ such that:

$$C_j \;\cap\; \left(C_1 \cup \dots \cup C_{j-1}\right) \;\subseteq\; C_i$$

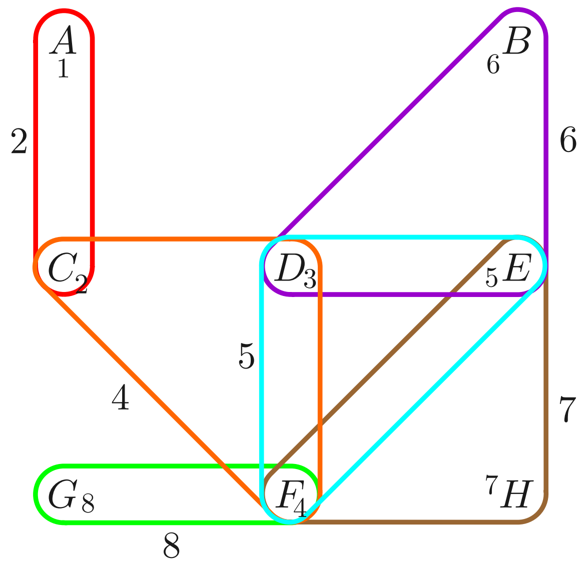


$\xi = \langle C_1,\ C_2,\ C_3,\ C_4,\ C_5,\ C_6 \rangle$

j				i
2	$C_2 \cap C_1$	$= \{C\}$	$\subseteq C_1$	1
3	$C_3 \cap (C_1 \cup C_2)$	$= \{D, F\}$	$\subseteq C_2$	2
4	$C_4 \cap (C_1 \cup C_2 \cup C_3)$	$= \{D, E\}$	$\subseteq C_3$	3
5	$C_5 \cap (C_1 \cup C_2 \cup C_3 \cup C_4)$	$= \{E, F\}$	$\subseteq C_3$	3
6	$C_6 \cap (C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5)$	$= \{F\}$	$\subseteq C_5$	5

ξ has running intersection property

If a node ordering α of an undirected graph $G = (V, E)$ is perfect and the cliques of G are ordered according to the highest rank (w. r. t. α) of the containing nodes, then this clique ordering has RIP.

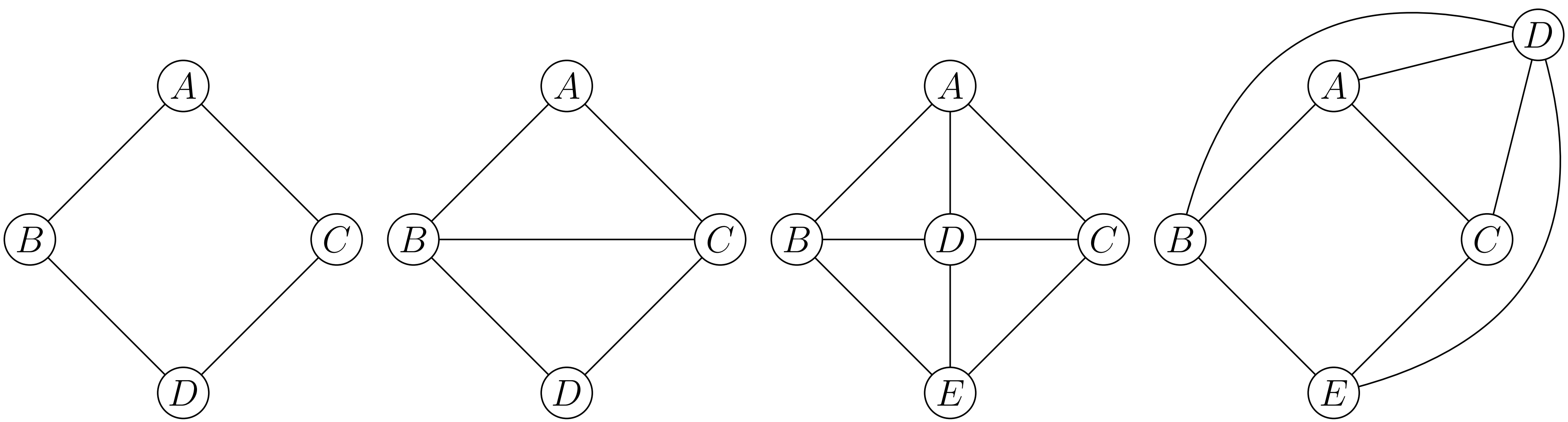


Clique	Rank		
$\{A, C\}$	$\max\{\alpha(A), \alpha(C)\}$	$= 2$	$\rightarrow C_1$
$\{C, D, F\}$	$\max\{\alpha(C), \alpha(D), \alpha(F)\}$	$= 4$	$\rightarrow C_2$
$\{D, E, F\}$	$\max\{\alpha(D), \alpha(E), \alpha(F)\}$	$= 5$	$\rightarrow C_3$
$\{B, D, E\}$	$\max\{\alpha(B), \alpha(D), \alpha(E)\}$	$= 6$	$\rightarrow C_4$
$\{F, E, H\}$	$\max\{\alpha(F), \alpha(E), \alpha(H)\}$	$= 7$	$\rightarrow C_5$
$\{F, G\}$	$\max\{\alpha(F), \alpha(G)\}$	$= 8$	$\rightarrow C_6$

How to get a perfect ordering?

Triangulated Graph

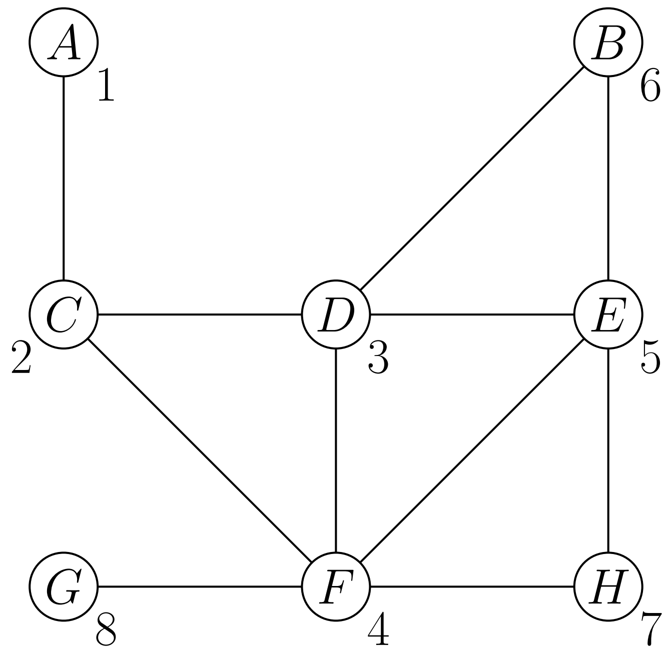
An undirected graph is called *triangulated* if every simple loop (i. e. path with identical start and end node but with any other node occurring at most once) of length greater 3 has a chord.



not triangulated   triangulated   not triangulated   no chord for $\langle A, B, E, C\rangle$

Maximum Cardinality Search

Let $G = (V, E)$ be an undirected graph. An ordering according *maximum cardinality search (MCS)* is obtained by first assigning 1 to an arbitray node. If n numbers are assigned the node that is connected to most of the nodes already numbered gets assigned number $n + 1$.



3 can be assigned to D or F

6 can be assigned to H or B

Triangulated Graphs (3)

An undirected graph is triangulated iff the ordering obtained by MCS is perfect.

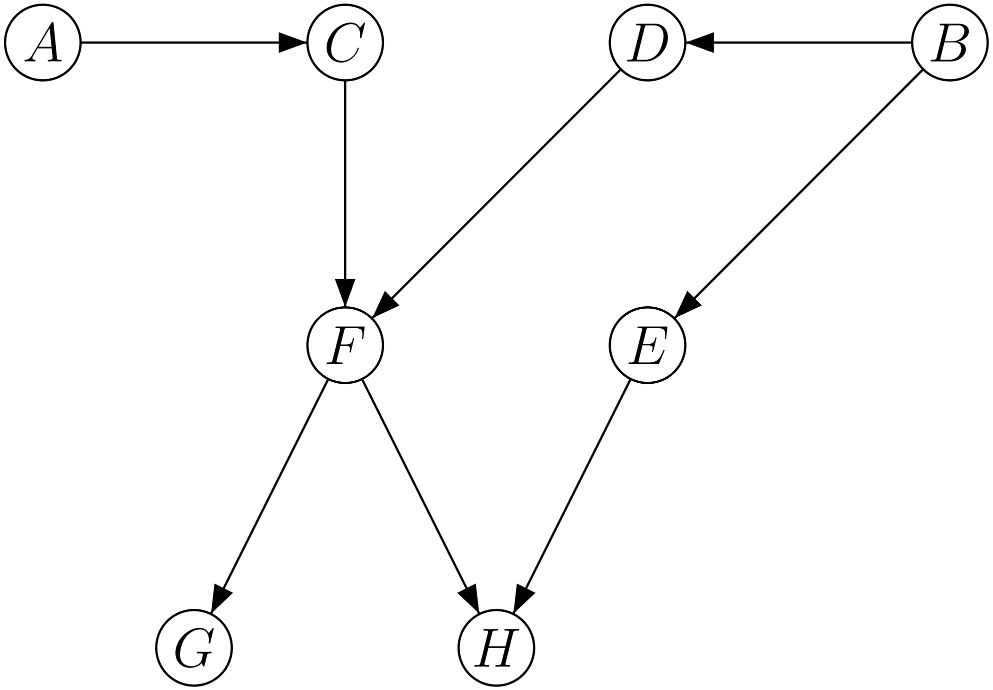
To check whether a graph is triangulated is efficient to implement. The optimization problem that is related to the triangulation task is NP-hard. However, there are good heuristics.

Moral Graph (Repetition)

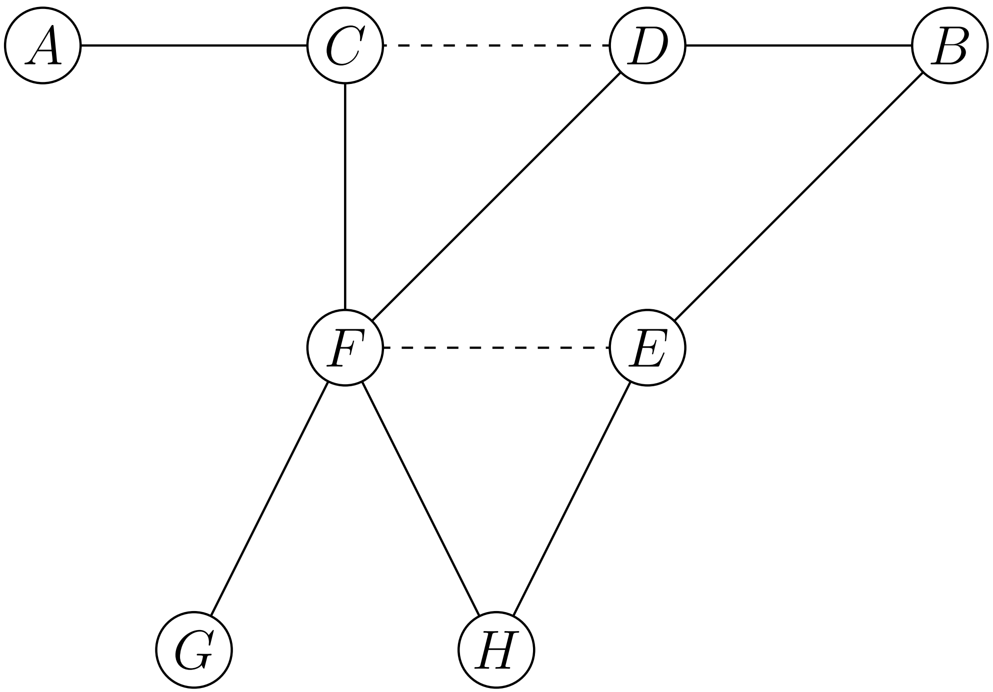
Let $G = (V, E)$ be a directed acyclic graph. If $u, w \in W$ are parents of $v \in V$ connect u and w with an (arbitrarily oriented) edge. After the removal of all edge directions the resulting graph $G_m = (V, E')$ is called the *moral graph* of G.

Join-Tree Construction (1)

Given directed graph.

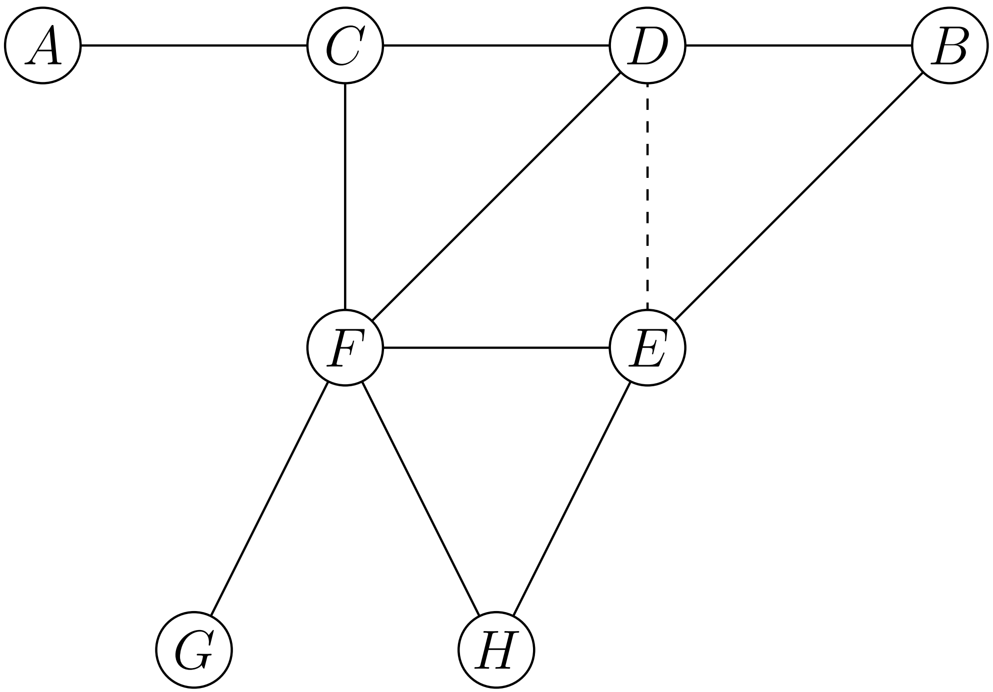


Join-Tree Construction (2)



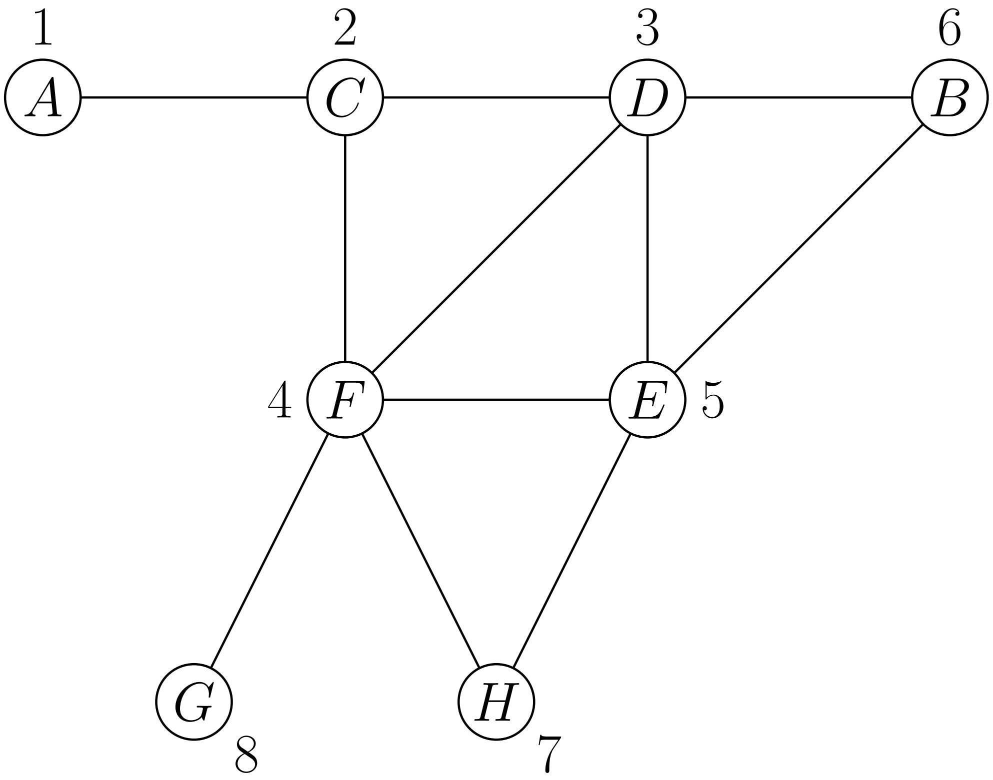
- Moral graph

Join-Tree Construction (3)



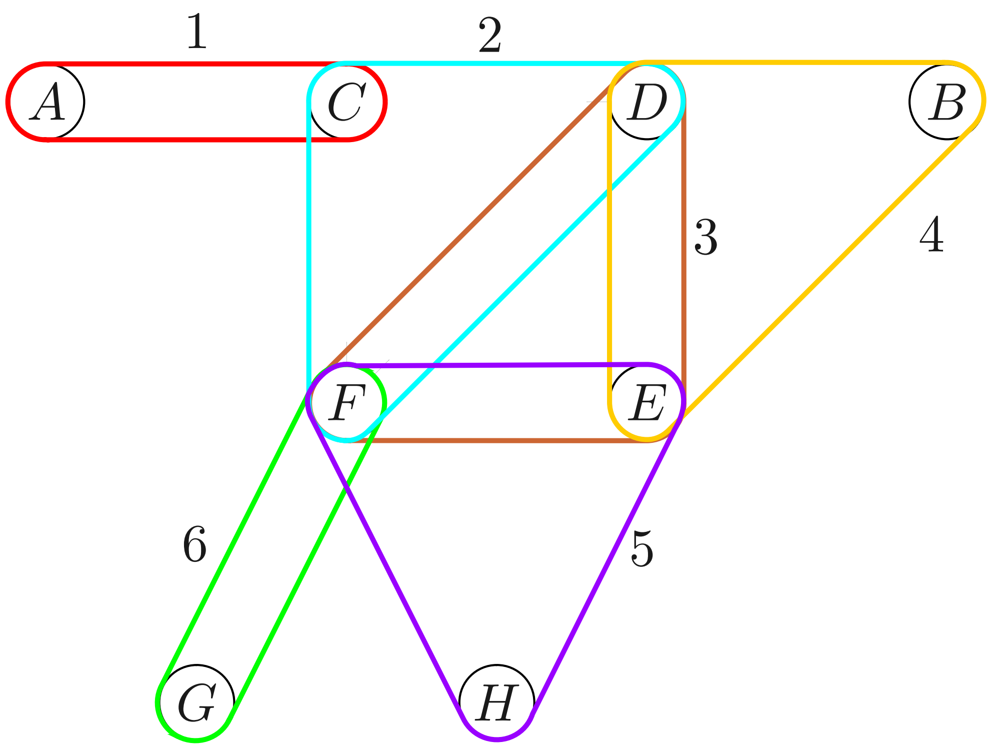
- Moral graph
- Triangulated graph

Join-Tree Construction (4)



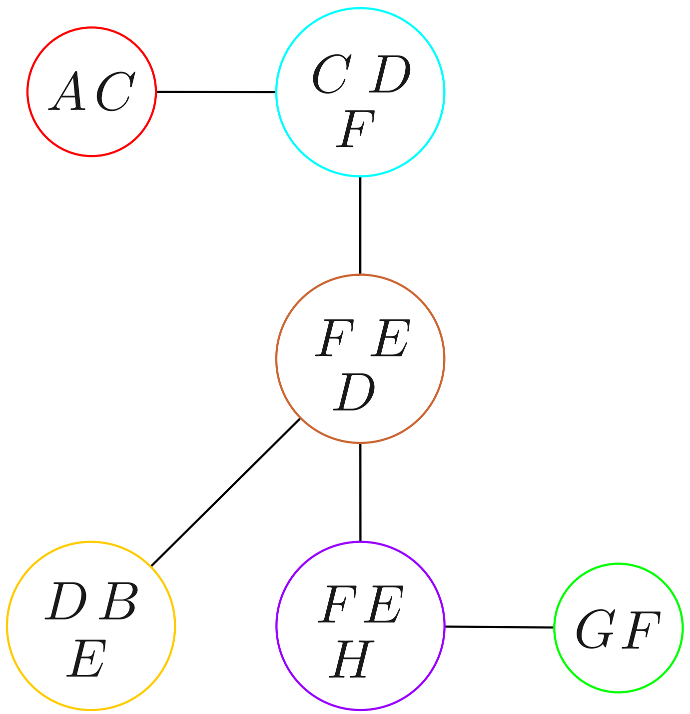
- Moral graph
- Triangulated graph
- MCS yields perfect ordering

Join-Tree Construction (5)



- Moral graph
- Triangulated graph
- MCS yields perfect ordering
- Clique order has RIP

Join-Tree Construction (6)



- Moral graph
- Triangulated graph
- MCS yields perfect ordering
- Clique order has RIP
- Form a join-tree

  Two cliques can be connected if they have a non-empty intersection. The generation of the tree follows the RIP. In case of a tie, connect cliques with the largest intersection. (e. g. DBE—FED instead of DBE—CFD) Break remaining ties arbitrarily.

Example: Expert Knowledge

Qualitative knowledge:

Metastatic cancer is a possible cause of brain tumor, and is also an explanation for increased total serum calcium. In turn, either of these could explain a patient falling into a coma. Severe headache is also possibly associated with a brain tumor.

Special case:

The patient has heavy headache.

Query:

Will the patient fall into coma?

Example: Choice of State Space

Attribute		Possible Values	
A	metastatic cancer	$\mathrm{dom}(A) = \{a_1, a_2\}$	$\cdot_1 =$ existing
B	increased total serum calcium	$\mathrm{dom}(B) = \{b_1, b_2\}$	$\cdot_2 =$ notexisting
C	brain tumor	$\mathrm{dom}(C) = \{c_1, c_2\}$	
D	coma	$\mathrm{dom}(D) = \{d_1, d_2\}$	
E	severe headache	$\mathrm{dom}(E) = \{e_1, e_2\}$	

Exhaustive state space:

$$\Omega = \mathrm{dom}(A) \times \mathrm{dom}(B) \times \mathrm{dom}(C) \times \mathrm{dom}(D) \times \mathrm{dom}(E)$$

Marginal and conditional probabilities have to be specified!

Example: Qualitative Knowledge

$$\left.\begin{aligned} P(e_1 \mid c_1) &= 0.8 \\ P(e_1 \mid c_2) &= 0.6 \end{aligned}\right\} \text{headaches common, but more common if tumor present}$$

$$\left.\begin{aligned} P(d_1 \mid b_1, c_1) &= 0.8 \\ P(d_1 \mid b_1, c_2) &= 0.8 \\ P(d_1 \mid b_2, c_1) &= 0.8 \\ P(d_1 \mid b_2, c_2) &= 0.05 \end{aligned}\right\} \text{coma rare but common, if either cause is present}$$

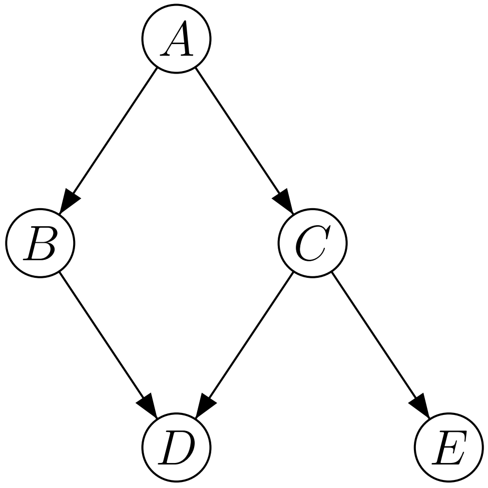
$$\left.\begin{aligned} P(b_1 \mid a_1) &= 0.8 \\ P(b_1 \mid a_2) &= 0.2 \end{aligned}\right\} \begin{array}{c}\text{increased calcium uncommon,} \\ \text{but common consequence of metastases}\end{array}$$

$$\left.\begin{aligned} P(c_1 \mid a_1) &= 0.2 \\ P(c_1 \mid a_2) &= 0.05 \end{aligned}\right\} \text{brain tumor rare, and uncommon consequence of metastases}$$

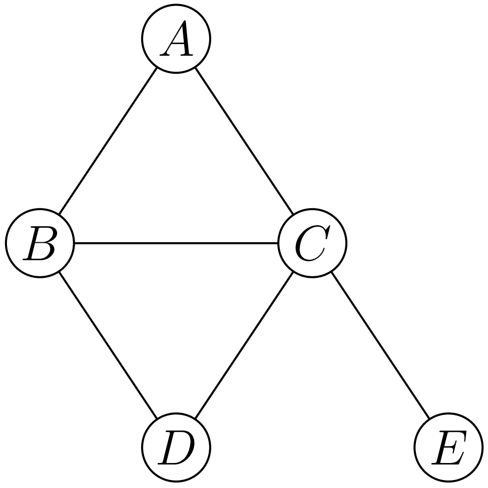
$$P(a_1) = 0.2 \quad \} \text{ incidence of metastatic cancer in relevant clinic}$$

Propagation on Cliques (1)

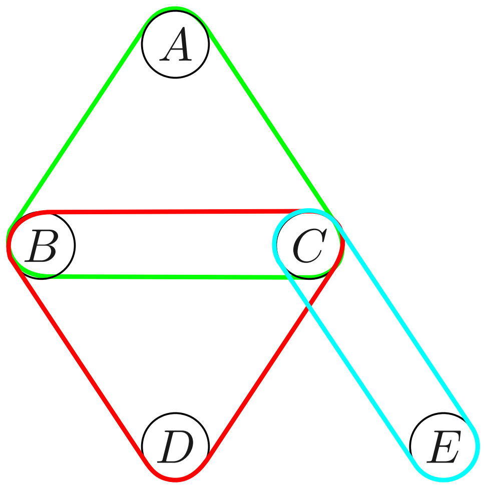
Example: Metastatic Cancer



Dependencies



Moralization/Triangulation



MCS, hyper graph

Clique tree with separator sets

Propagation on Cliques (3)

Quantitative knowledge:

(a, b, c)	$P(a, b, c)$
a_1, b_1, c_1	0.032
a_2, b_1, c_1	0.008
$\vdots$	$\vdots$
a_2, b_2, c_2	0.608

(b, c, d)	$P(b, c, d)$
b_1, c_1, d_1	0.032
b_2, c_1, d_1	0.032
$\vdots$	$\vdots$
b_2, c_2, d_2	0.608

(c, e)	$P(c, e)$
c_1, e_1	0.064
c_2, e_1	0.552
c_1, e_2	0.016
c_2, e_2	0.368

Potential representation:

$$\begin{aligned} P(A, B, C, D, E,) &= P(A \mid \emptyset)P(B \mid A)P(C \mid A)P(D \mid BC)P(E \mid C) \\ &= \frac{P(A, B, C)P(B, C, D), P(C, E)}{P(BC)P(C)} \end{aligned}$$

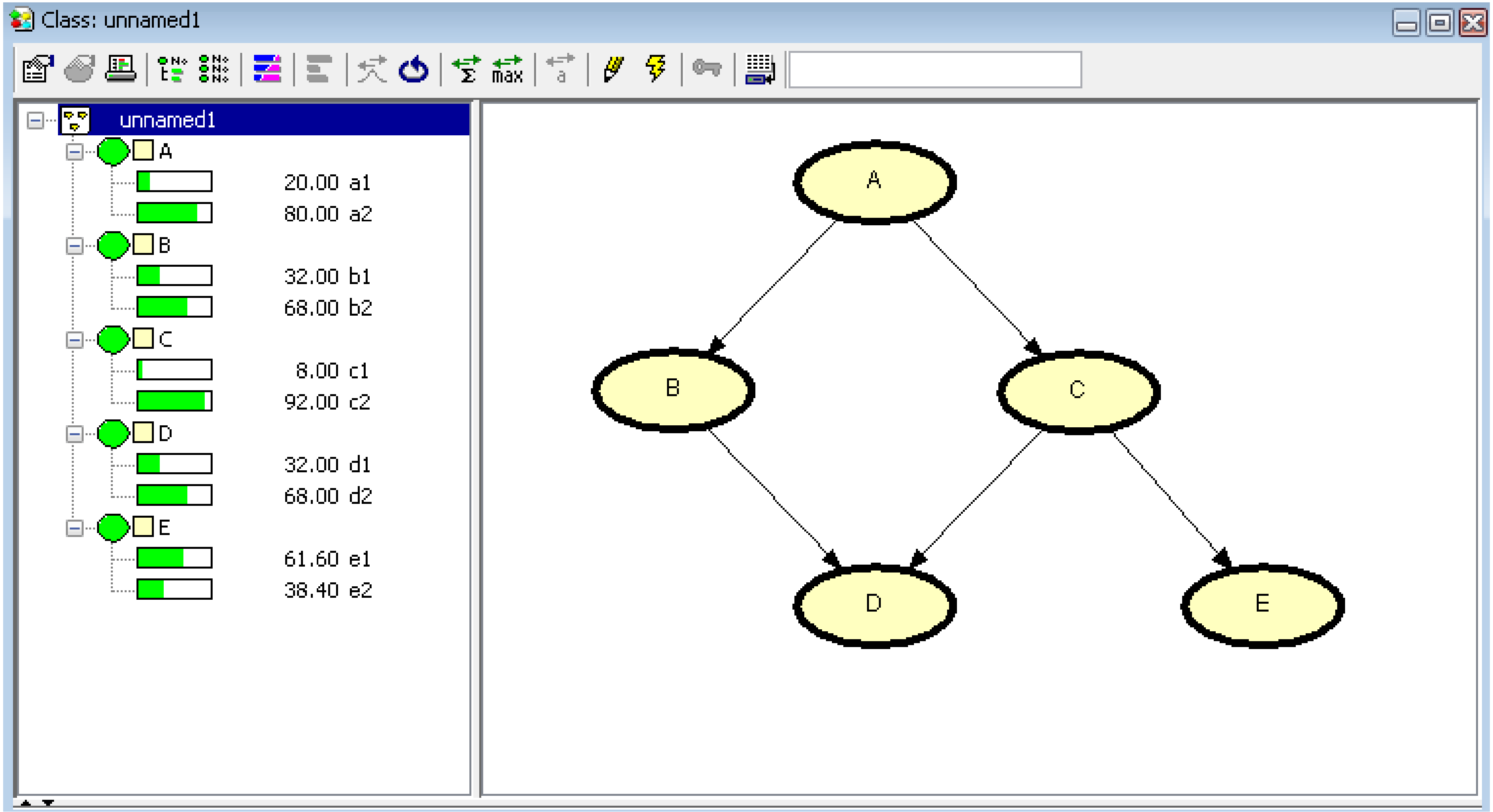
Propagation on Cliques (4)

Propagation:

$$P(d_1) = 0.32, \quad \text{evidence } E = e_1, \quad \text{desired:} \quad P^*(\ldots) = P(\cdot \mid \{e_1\})$$

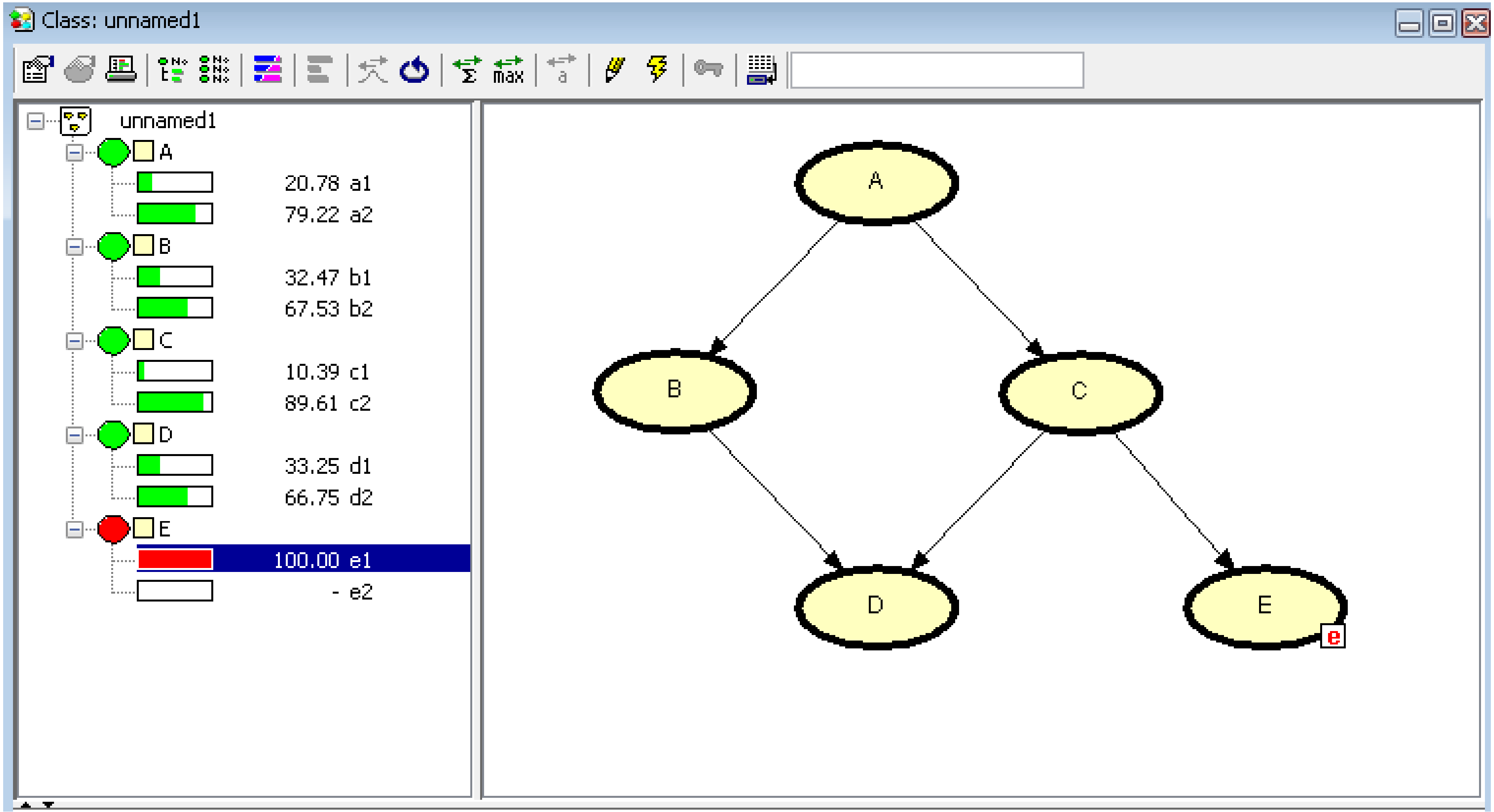
$P^*(c)$	$= P(c \mid e_1)$	conditional marginal distribution
$P^*(b,c,d)$	$= \dfrac{P(b,c,d)}{P(c)} P(c \mid e_1)$	multipl./division with separation prob.
$P(b,c,d),\ P^*(b,c)$		calculate marginal distributions
$P^*(a,b,c)$	$= \dfrac{P(a,b,c)}{P(b,c)} P(b,c \mid e_1)$	multipl./division with separation prob.
$P^*(d_1)$	$= P(d_1 \mid e_1) = 0.33$	

Propagation on Cliques (5)



Marginal distributions in the HUGIN tool.

Propagation on Cliques (6)



Conditional marginal distributions with evidence $E = e_1$

Potential Representation

Let $V = \{X_j\}$ be a set of random variables $X_j : \Omega \to \mathrm{dom}(X_j)$ and P the joint distribution over V. Further, let

$$\{W_i \mid W_i \subseteq V, 1 \leq i \leq p\}$$

a family of subsets of V with associated functions

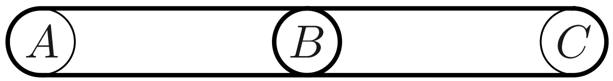
$$\psi_i : \bigtimes_{X_j \in W_i} \mathrm{dom}(X_j) \to \mathbb{R}$$

It is said that $P(V)$ *factorizes* according $\big(\{W_1, \ldots, W_p\}, \{\psi_1, \ldots, \psi_p\}\big)$ if $P(V)$ can be written as:

$$P(v) = k \cdot \prod_{i=1}^{p} \psi_i(w_i)$$

where $k \in \mathbb{R}$, w_i is a realization of W_i that meets the values of v.

Example



$$V = \{A, B, C\},\ W_1 = \{A, B\},\ W_2 = \{B, C\}$$

$$\text{dom}(A) = \{a_1, a_2\}$$
$$\text{dom}(B) = \{b_1, b_2\}$$
$$\text{dom}(C) = \{c_1, c_2\}$$

$$P(a, b, c) = \frac{1}{8}$$

$$\psi_1 : \{a_1, a_2\} \times \{b_1, b_2\} \to \mathbb{R}$$
$$\psi_2 : \{b_1, b_2\} \times \{c_1, c_2\} \to \mathbb{R}$$

$$\psi_1(a, b) = \frac{1}{4}$$
$$\psi_2(b, c) = \frac{1}{2}$$

$(\{W_1, W_2\}, \{\psi_1, \psi_2\})$ is a potential representation of P.

Factorization of a Belief Network

Let (V, E, P) be an belief network and $\{C_1, \dots, C_p\}$ the cliques of the join tree. For every node $v \in V$ choose a clique C such that v and all of its parents are contained in C, i. e. $\{v\} \cup c(v) \subseteq C$. The chosen clique is designated as $f(v)$.

To arrive at a factorization $(\{C_1, \dots, C_p\}, \{\psi_1, \dots, \psi_p\})$ of P the factor potentials are:

$$\psi_i(c_i) = \prod_{v : f(v) = C_i} P(v \mid c(v))$$

Separator Sets and Residual Sets

Let $\{C_1, \dots, C_p\}$ be a set of cliques w. r. t. V. The sets

$$S_i = C_i \cap (C_1 \cup \cdots \cup C_{i-1}), \qquad i = 1, \dots, p, \qquad S_1 = \emptyset$$

are called *separator sets* with their corresponding *residual sets*

$$R_i = C_i \backslash S_i$$

Decomposition w. r. t. a Join-Tree

Given a clique ordering $\{C_1, \ldots, C_p\}$ that satisfies the RIP, we can easily conclude the following separation statements:

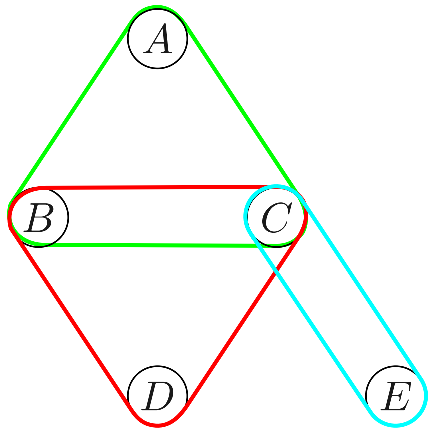
$$R_i \perp\!\!\!\perp (C_1 \cup \cdots \cup C_{i-1}) \backslash S_i \mid S_i \qquad \text{for } i > 1$$

Hence, we can formulate the following factorization:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{p} P(R_i \mid S_i),$$

which also gives us a representation in terms of conditional probabilities (as for directed graphs before).

Example

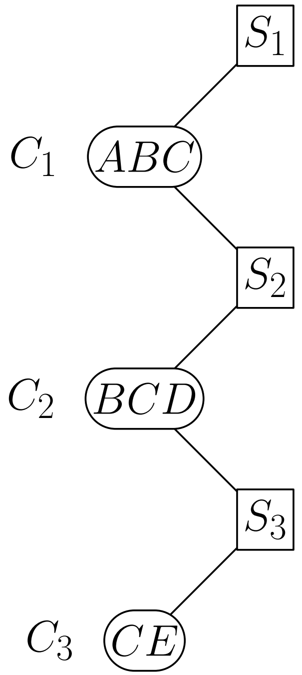


$S_1 = \emptyset$ $\quad R_1 = \{A, B, C\}$

$S_2 = \{B, C\}$ $\quad R_2 = \{D\}$

$S_3 = \{C\}$ $\quad R_3 = \{E\}$

$$f(A) = C_1$$
$$f(B) = C_1$$
$$f(C) = C_1$$
$$f(D) = C_2$$
$$f(E) = C_3$$



$$\psi_1(C_1) = P(A, B, C \mid \emptyset) = P(A) \cdot P(C \mid A) \cdot P(B|A)$$
$$\psi_2(C_2) = P(D \mid B, C)$$
$$\psi_3(C_3) = P(E \mid C)$$

Propagation is accomplished by sending messages across the cliques in the tree. The emerging potentials are maintained by each clique.

Main Idea

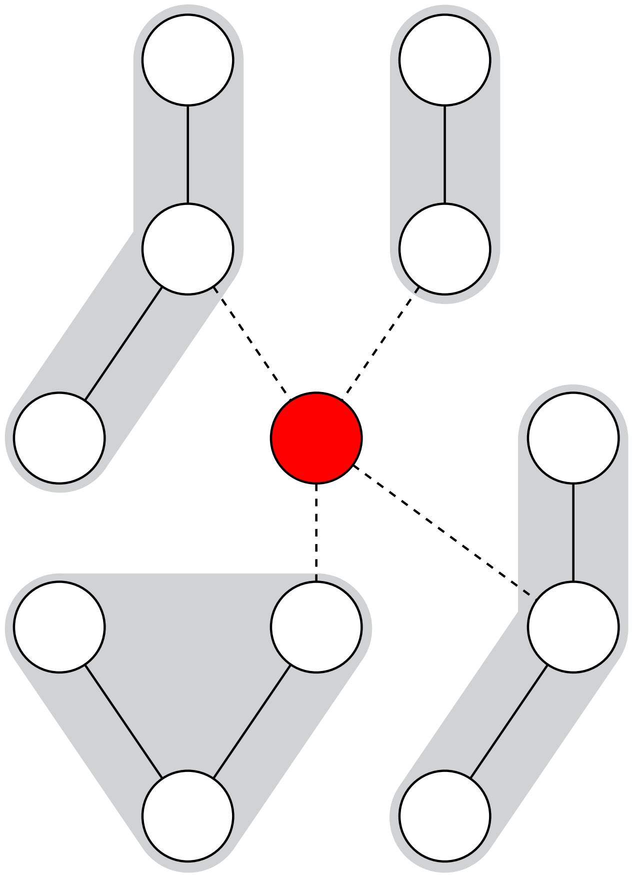
Incorporate evidence into the clique potentials.

Since we are dealing with a tree structure, exploit the fact that a clique "separates" all its neighboring cliques (and their respective subtrees) from each other.

Apply a message passing scheme to inform neighboring cliques about evidence.

Since we do not have edge directions, we will only need one type of message.

After having updated all cliques' potentials, we marginalize (and normalize) to get the probabilities of single attributes.



Incorporating Evidence

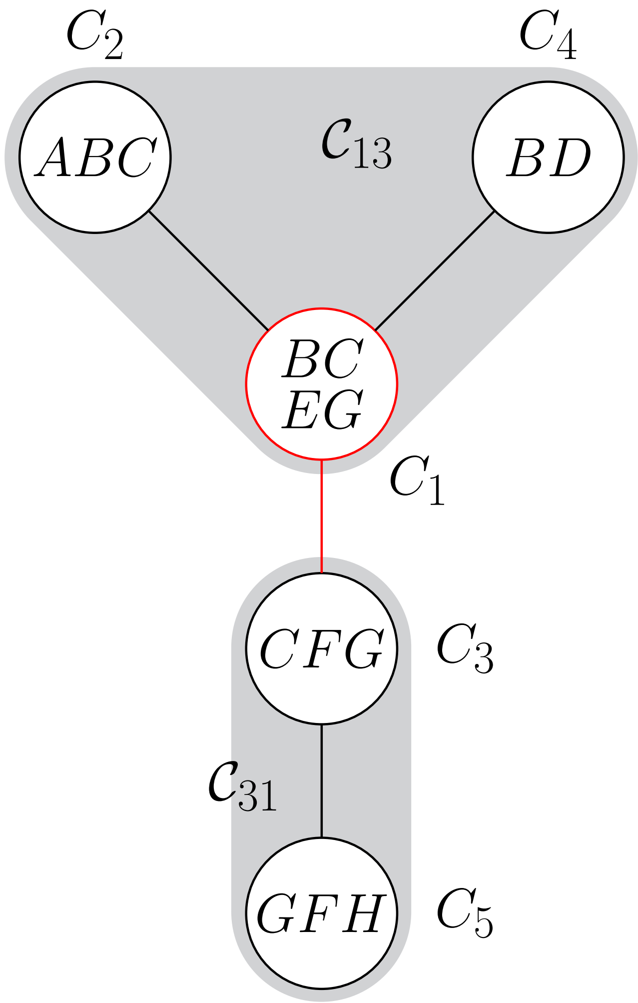
Every clique C_i maintains a potential function ψ_i.

If for an attribute E some evidence e becomes known, we alter all potential functions of cliques containing E as follows:

$$\psi_i^*(c_i) = \begin{cases} 0, & \text{if a value in } c_i \text{ is inconsistent with } e \\ \psi_i(c_i), & \text{otherwise} \end{cases}$$

All other potential functions are unchanged.

Notation and Nomenclature



In general:

Clique C_i has q neighboring cliques $B_1, \ldots, B_q$.

$\mathcal{C}_{ij}$ is the set of cliques in the subtree containing C_i after dropping the link to B_j.

X_{ij} is the set of attributes in the cliques of $\mathcal{C}_{ij}$.

$V = X_{ij} \cup X_{ji}$ (complementary sets)

$S_{ij} = S_{ji} = C_i \cap C_j$ (not shown here)

$R_{ij} = X_{ij} \setminus S_{ij}$ (not shown here)

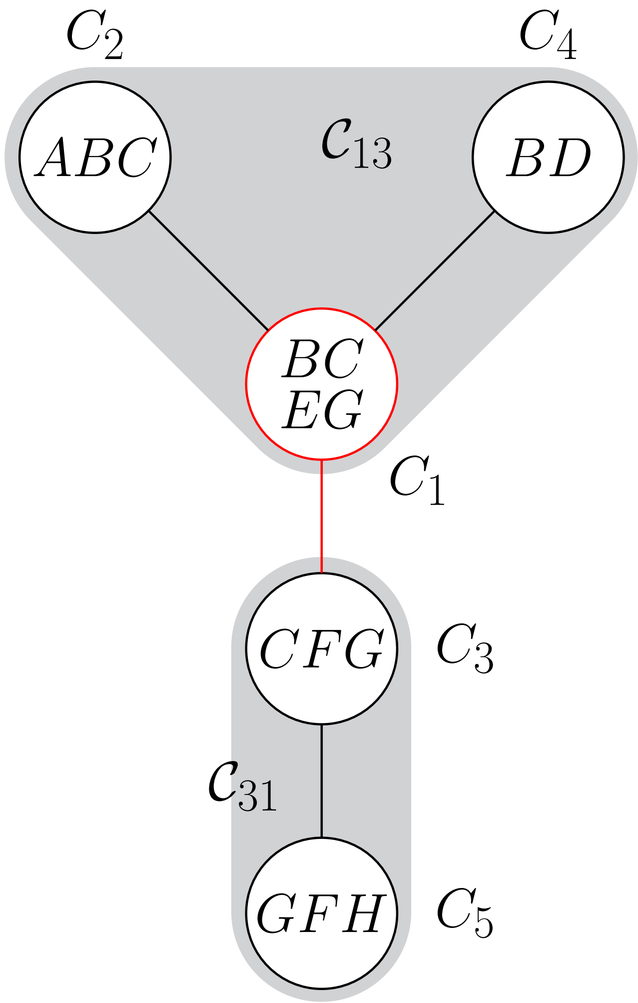
Here:

Neighbors of C_1: $\{C_2, C_4, C_3\}$, $\mathcal{C}_{13} = \{C_1, C_2, C_4\}$

$X_{13} = \{A, B, C, D, E, G\}$, $S_{13} = \{C, G\}$

$V = X_{13} \cup X_{31} = \{A, B, C, D, E, F, G, H\}$

$R_{13} = \{A, B, D, E\}$, $R_{31} = \{F, H\}$

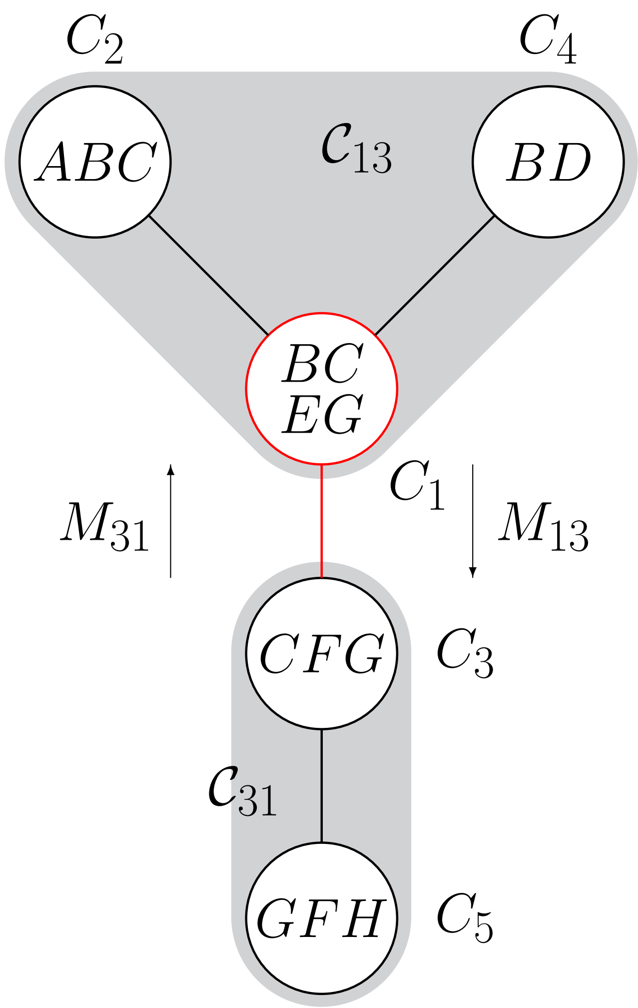


Task: Calculate $P(s_{ij})$:

$$\begin{aligned} V \setminus S_{ij} &= (X_{ij} \cup X_{ji}) \setminus S_{ij} \\ &= (X_{ij} \setminus S_{ij}) \;\cup\; (X_{ji} \setminus S_{ij}) \\ &= R_{ij} \;\cup\; R_{ji} \end{aligned}$$

$$\begin{aligned} V \setminus S_{13} &= (X_{13} \cup X_{31}) \setminus S_{13} \\ &= R_{13} \;\cup\; R_{31} \\ V \setminus \{C, G\} &= \{A, B, D, E\} \;\cup\; \{F, H\} \\ &= \{A, B, D, E, F, H\} \end{aligned}$$

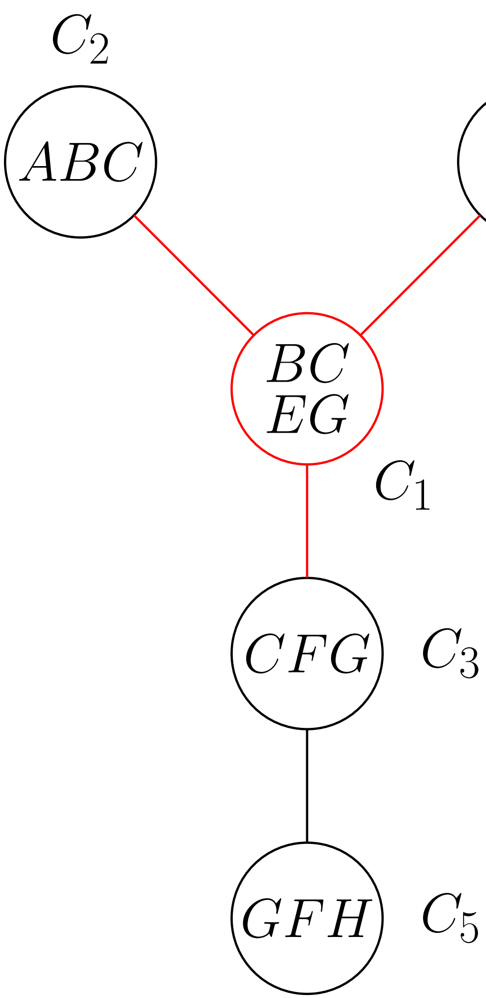
Note: R_{ij} is the set of attributes that are in C_i's subtree but not in B_j's. Therefore, R_{ij} and R_{ji} are always **disjoint**.



Task: Calculate $P(s_{ij})$:

$$
\begin{aligned}
P(s_{ij}) \quad &= \quad \sum_{v \setminus s_{ij}} \prod_{k=1}^{m} \psi_k(c_k) \\
&\overset{\text{last slide}}{=} \quad \sum_{r_{ij} \cup r_{ji}} \prod_{k=1}^{m} \psi_k(c_k) \\
&\overset{\text{sum rule}}{=} \quad \left(\sum_{r_{ij}} \prod_{c_k \in \mathcal{C}_{ij}} \psi_k(c_k) \right) \cdot \left(\sum_{r_{ji}} \prod_{c_k \in \mathcal{C}_{ji}} \psi_k(c_k) \right) \\
&= \quad M_{ij}(s_{ij}) \cdot M_{ji}(s_{ij})
\end{aligned}
$$

M_{ij} is the message sent from C_i to neighbor B_j and vice versa.

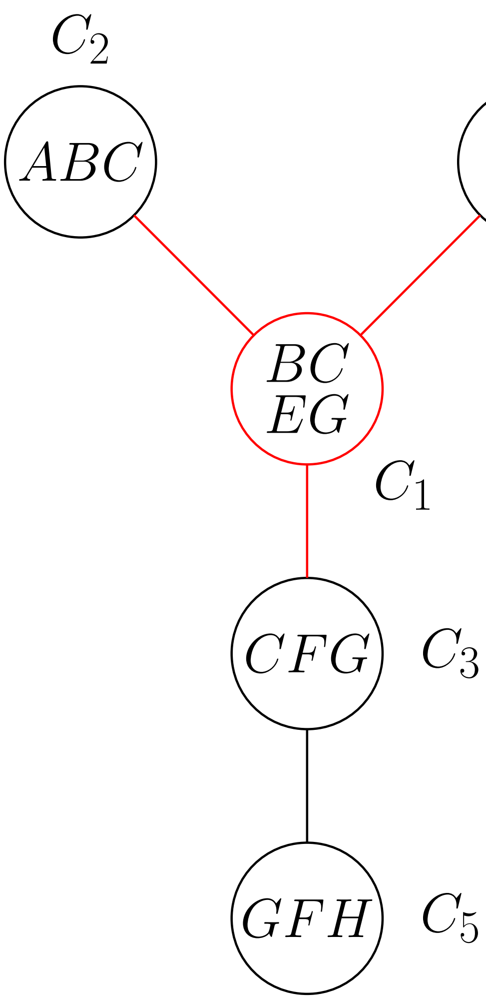


Task: Calculate $P(c_i)$:

$$\begin{aligned} V \setminus C_i &= \left(\bigcup_{k=1}^{q} X_{ki} \right) \setminus C_i \\ &= \bigcup_{k=1}^{q} \left(X_{ki} \setminus C_i \right) \\ &= \bigcup_{k=1}^{q} R_{ki} \end{aligned}$$

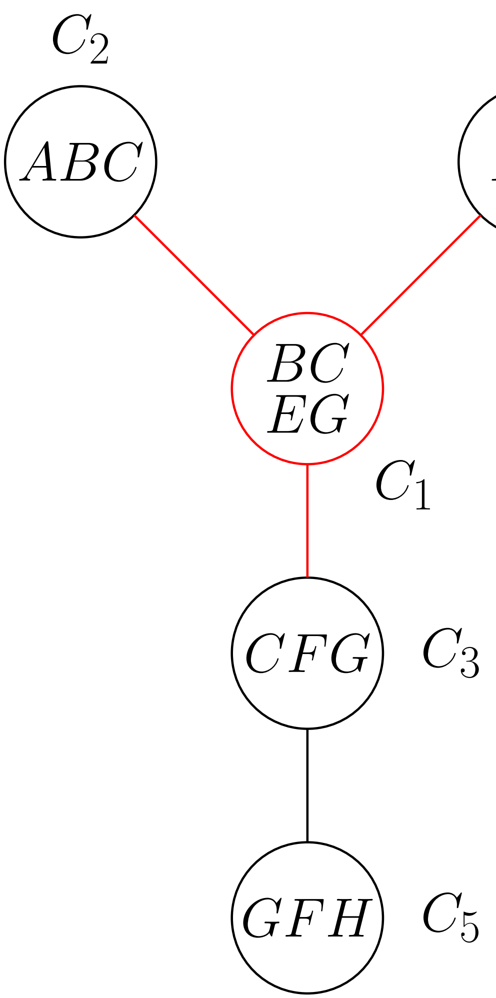
Example:

$$\begin{aligned} V \setminus C_1 &= R_{21} \cup R_{41} \cup R_{31} \\ \{A, D, F, H\} &= \{A\} \cup \{D\} \cup \{F, H\} \end{aligned}$$



Task: Calculate $P(c_i)$:

$$\begin{aligned}
P(c_i) &= \underbrace{\sum_{v \setminus c_i}}_{\text{Marginalization}} \underbrace{\prod_{j=1}^{m} \psi_j(c_j)}_{\text{Decomposition}} \\
&= \psi_i(c_i) \sum_{v \setminus c_i} \prod_{i \neq j} \psi_j(c_j) \\
&= \psi_i(c_i) \sum_{r_{1i} \cup \cdots \cup r_{qi}} \prod_{i \neq j} \psi_j(c_j) \\
&= \psi_i(c_i) \underbrace{\left(\sum_{r_{1i}} \prod_{c_k \in \mathcal{C}_{1i}} \psi_k(c_k) \right)}_{M_{1i}(s_{ij})} \cdots \underbrace{\left(\sum_{r_{qi}} \prod_{c_k \in \mathcal{C}_{qi}} \psi_k(c_k) \right)}_{M_{qi}(s_{ij})} \\
&= \psi_i(c_i) \prod_{j=1}^{q} M_{ji}(s_{ij})
\end{aligned}$$



Example: $P(c_1)$:

$$P(c_1) = \psi_1(c_1) M_{21}(s_{12}) M_{41}(s_{14}) M_{31}(s_{13})$$

$M_{ij}(s_{ij})$ can be simplified further (without proof):

$$\begin{aligned} M_{ij}(s_{ij}) &= \sum_{r_{ij}} \prod_{c_k \in \mathcal{C}_{ij}} \psi_k(c_k) \\ &= \sum_{c_i \setminus s_{ij}} \psi_i(c_i) \prod_{k \neq j} M_{ki}(s_{ki}) \end{aligned}$$

Final Algorithm

Input: Join tree $(\mathcal{C}, \Psi)$ over set of variables V and evidence $E = e$.

Output: The a-posteriori probability $P(x_i \,|\, e)$ for every non-evidential X_i.

Initialization: Incorporate evidence $E = e$ into potential functions.

Iterations:

1. For every clique C_i do: For every neighbor B_j of C_i do: If C_i has received all messages from the *other* neighbors, calculate and send $M_{ij}(s_{ij})$ to B_j.
2. Repeat step 1 until no message is calculated.
3. Calculate the joint probability distribution for every clique:

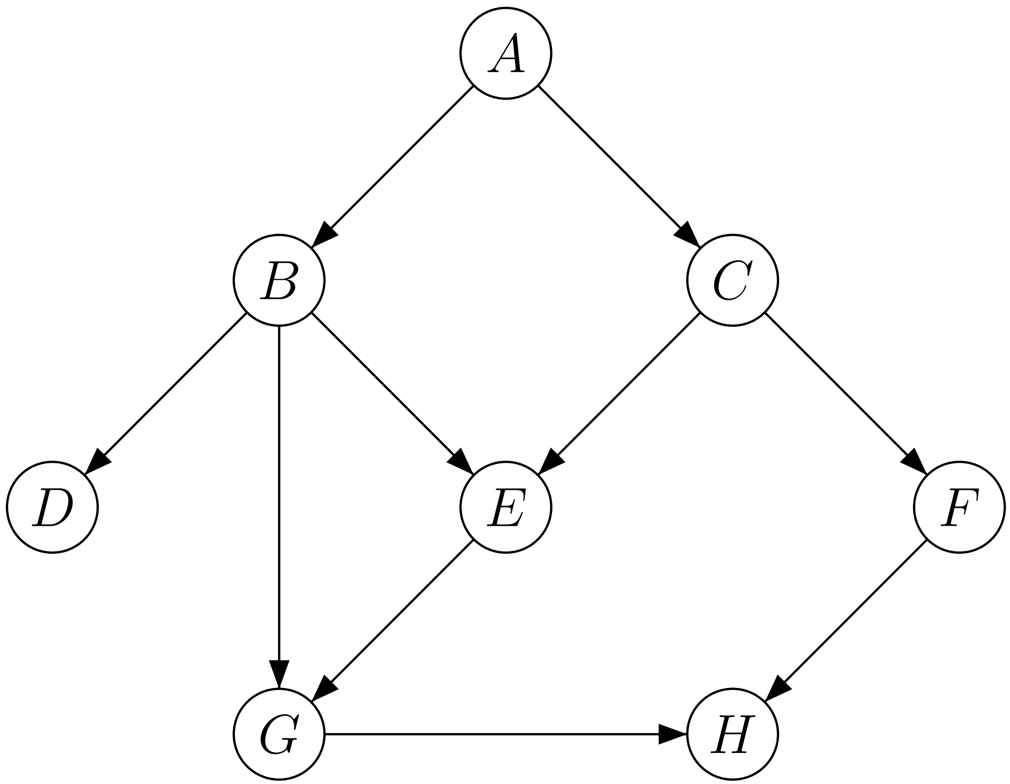
$$P(c_i) \;\propto\; \psi_i(c_i) \prod_{j=1}^{q} M_{ji}(s_{ij})$$

4. For every $X \in V$ calculate the a-posteriori probability:

$$P(x_i \mid e) \;=\; \sum_{c_k \backslash x_i} P(c_k)$$

where C_k is the smallest clique containing X_i.

Example: Putting it together

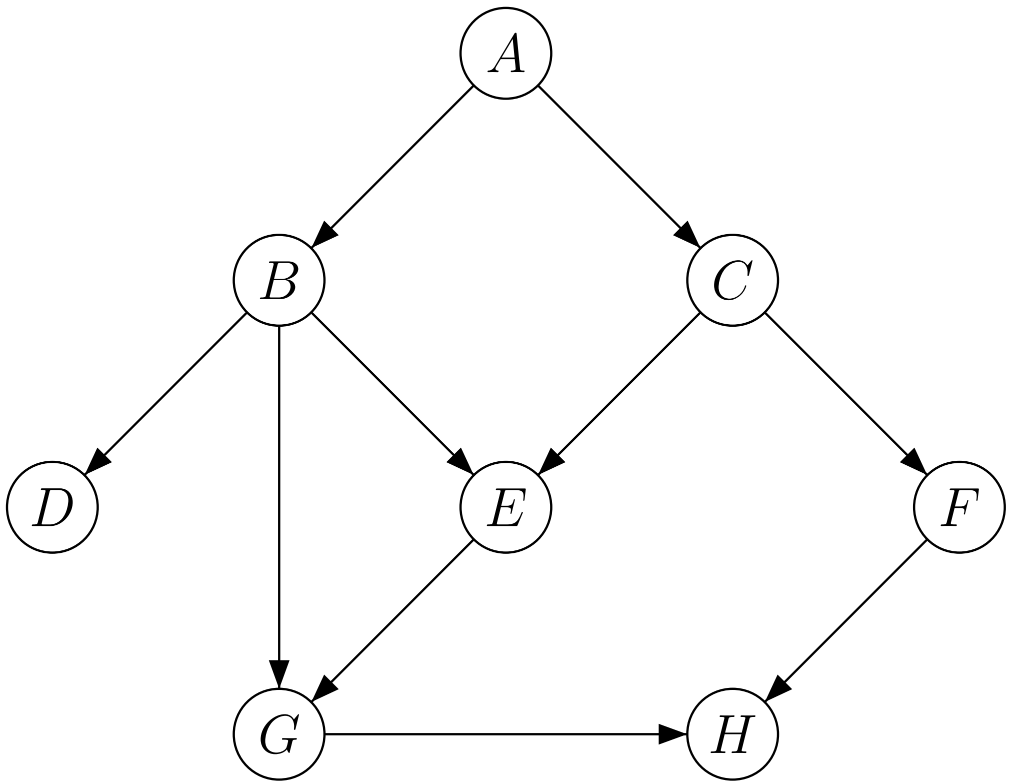


Goals: Find the marginal distributions and update them when evidence $H = h_1$ becomes known.

Steps:

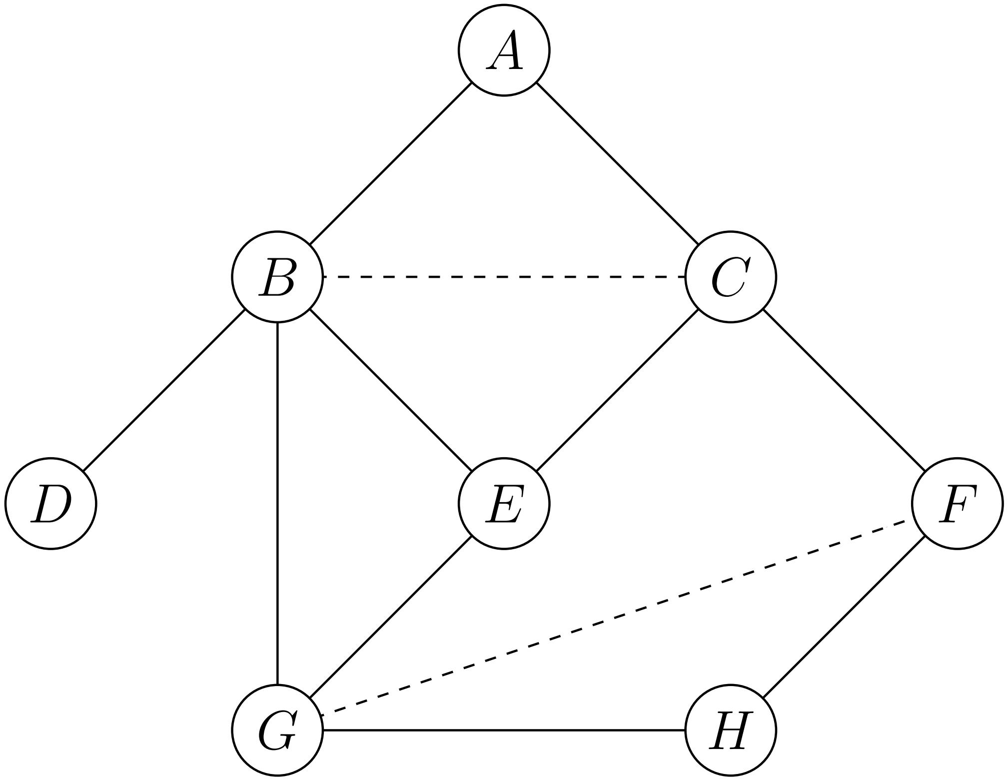
1. Transform network into join-tree.
2. Specify factor potentials.
3. Propagate "zero" evidence to obtain the marginals before evidence is present.
4. Update factor potentials w. r. t. the evidence and do another propagation run.

Example: Step 1: Find a Join-Tree



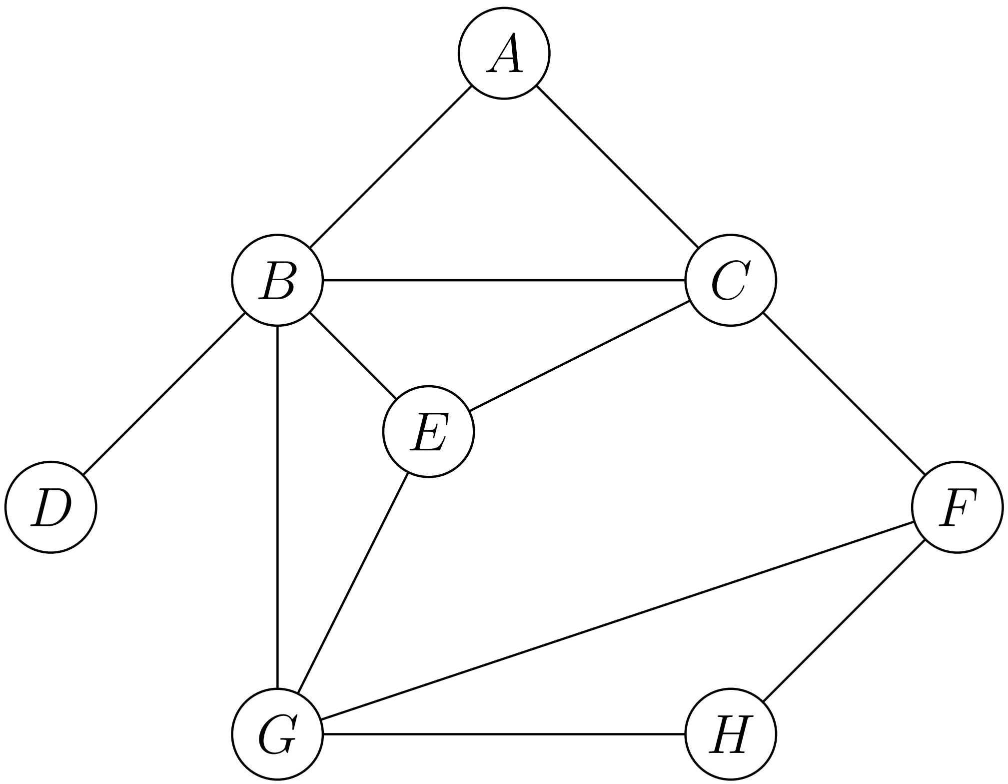
Join-Tree creation:

Example: Step 1: Find a Join-Tree



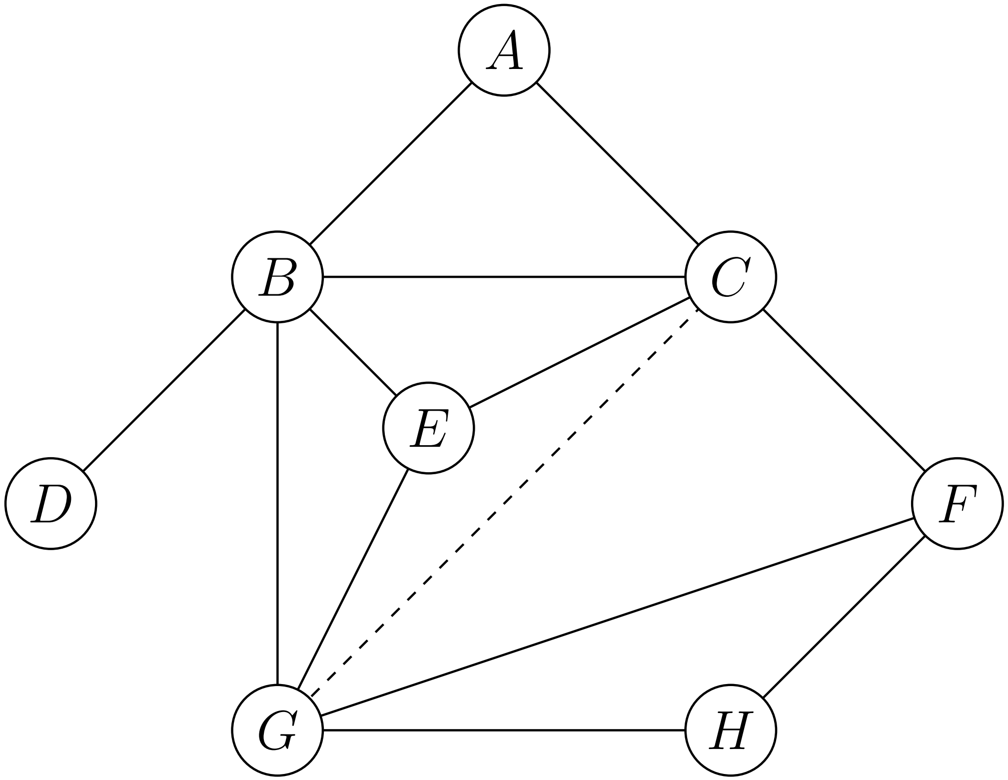
Join-Tree creation:

1. Moralize the graph.



Join-Tree creation:

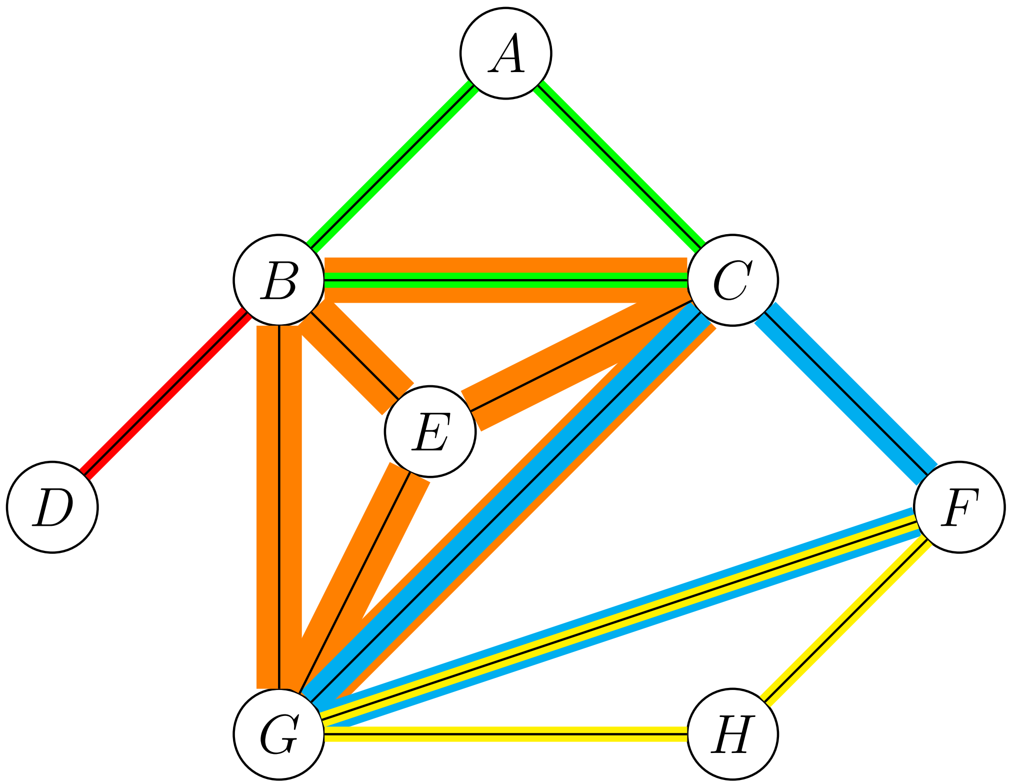
1. Moralize the graph.
2. Not yet triangulated.



Join-Tree creation:

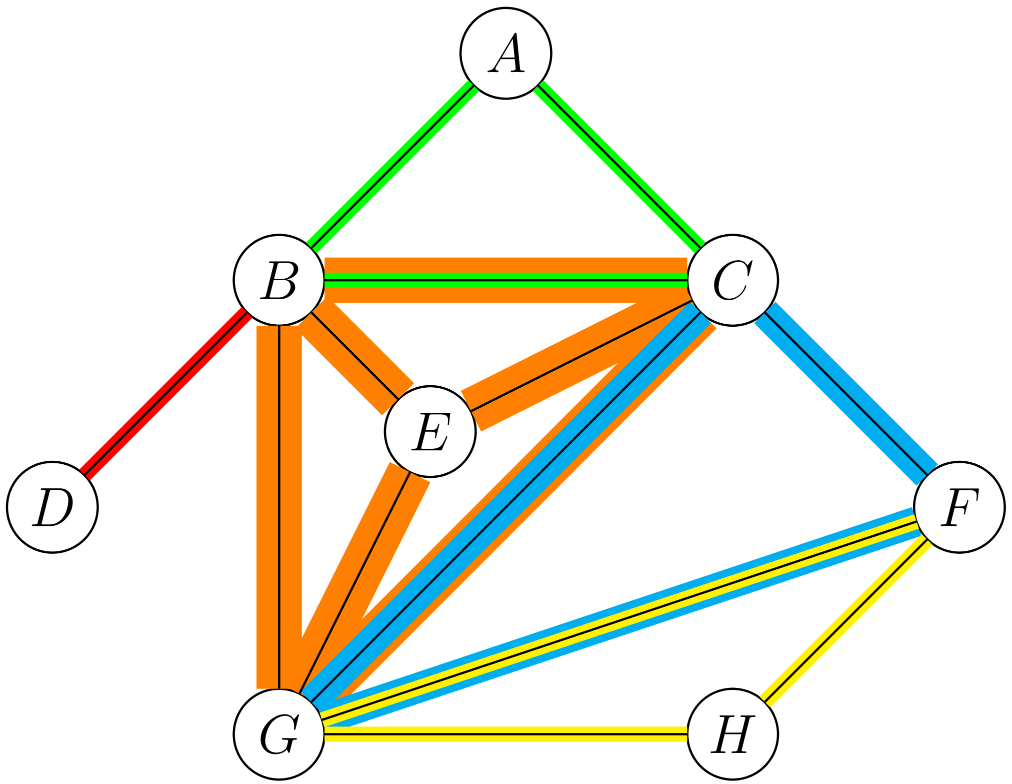
1. Moralize the graph.
2. Triangulate the graph.

Example: Step 1: Find a Join-Tree



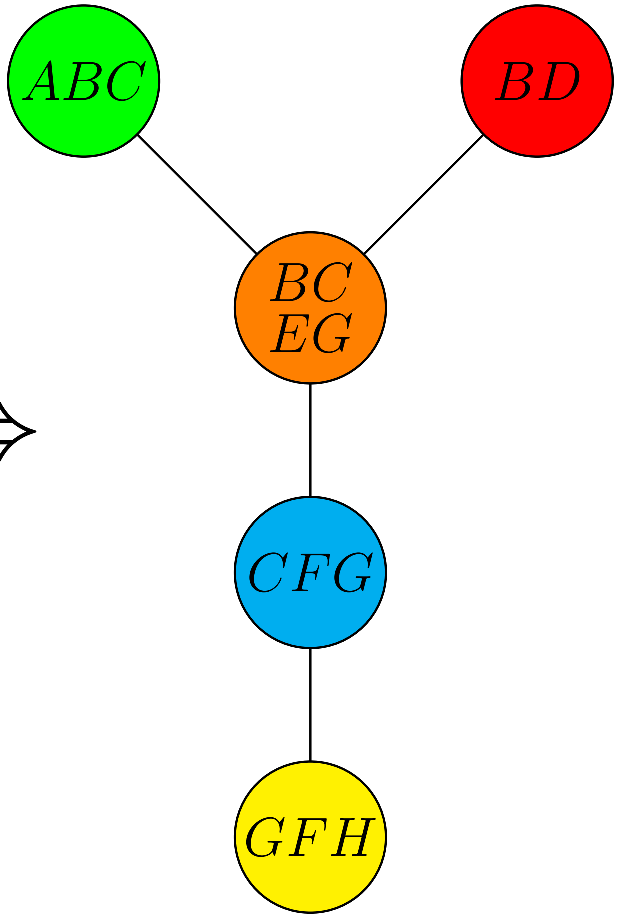
Join-Tree creation:

1. Moralize the graph.
2. Triangulate the graph.
3. Identify the maximal cliques.

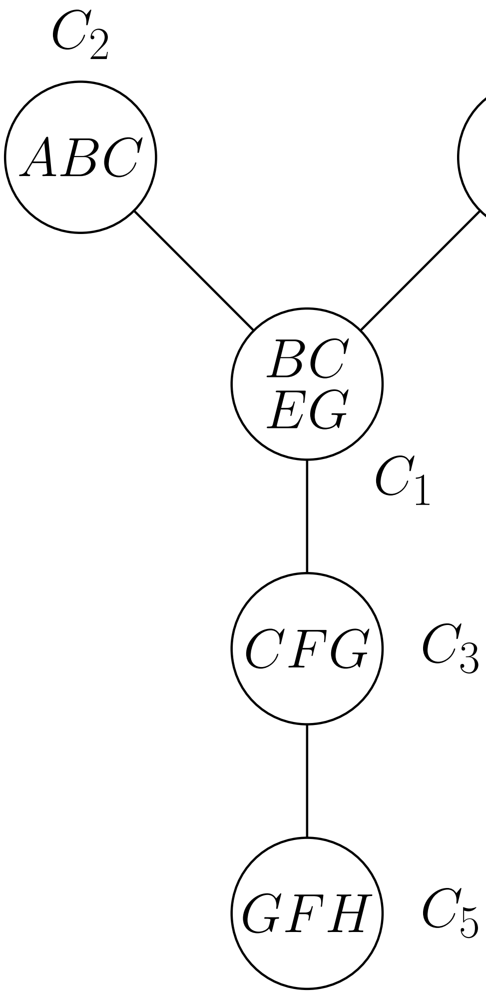


Example Bayesian network

$\Rightarrow$



One of the join trees



Decomposition of $P(A, B, C, D, E, F, G, H)$:

$$
\begin{aligned}
P(a, b, c, d, e, f, g, h) &= \prod_{i=1}^{5} \Psi_i(c_i) \\
&= \Psi_1(b, c, e, g) \cdot \Psi_2(a, b, c) \\
&\quad \cdot \Psi_3(c, f, g) \cdot \Psi_4(b, d) \\
&\quad\quad \cdot \Psi_5(g, f, h)
\end{aligned}
$$

Where to get the factor potentials from?

Example: Step 2: Specify the Factor Potentials

As long as the factor potentials multiply together as on the previous slide, we are free to choose them.

Option 1: A factor potential of clique C_i is the product of all conditional probabilities of all node families properly contained in C_i:

$$\Psi_i(c_i) = 1 \cdot \prod_{\substack{\{X_i\} \cup Y_i \subseteq C_i \wedge \\ \text{parents}(X_i) = Y_i}} P(x_i \mid y_i)$$

The 1 stresses that if no node family satisfies the product condition, we assign a constant 1 to the potential.

Option 2: Choose potentials from the decomposition formula:

$$P\left(\bigcup_{i=1}^{n} C_i\right) = \frac{\prod_{i=1}^{n} P(C_i)}{\prod_{j=1}^{m} P(S_j)}$$

Example: Step 2: Specify the Factor Potentials

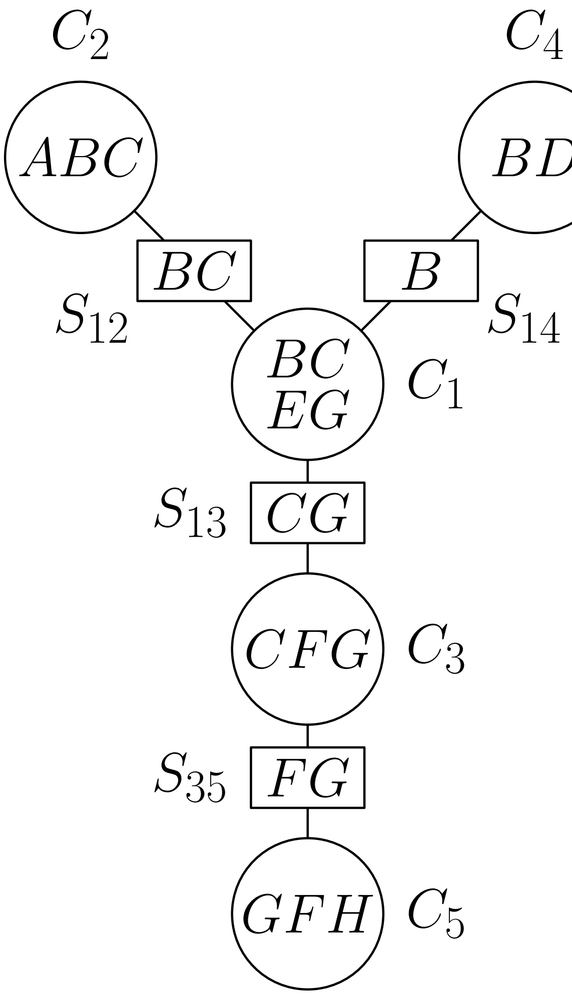
Option 1: Factor potentials according to the conditional distributions of the node families of the underlying Bayesian network:

$$\begin{aligned}
\Psi_1(b,c,e,g) &= P(e \mid b,c) \cdot P(g \mid e,b) \\
\Psi_2(a,b,c) &= P(b \mid a) \cdot P(c \mid a) \cdot P(a) \\
\Psi_3(c,f,g) &= P(f \mid c) \\
\Psi_4(b,d) &= P(d \mid b) \\
\Psi_5(g,f,h) &= P(h \mid g,f)
\end{aligned}$$

(This assignment of factor potentials is used in this example.)

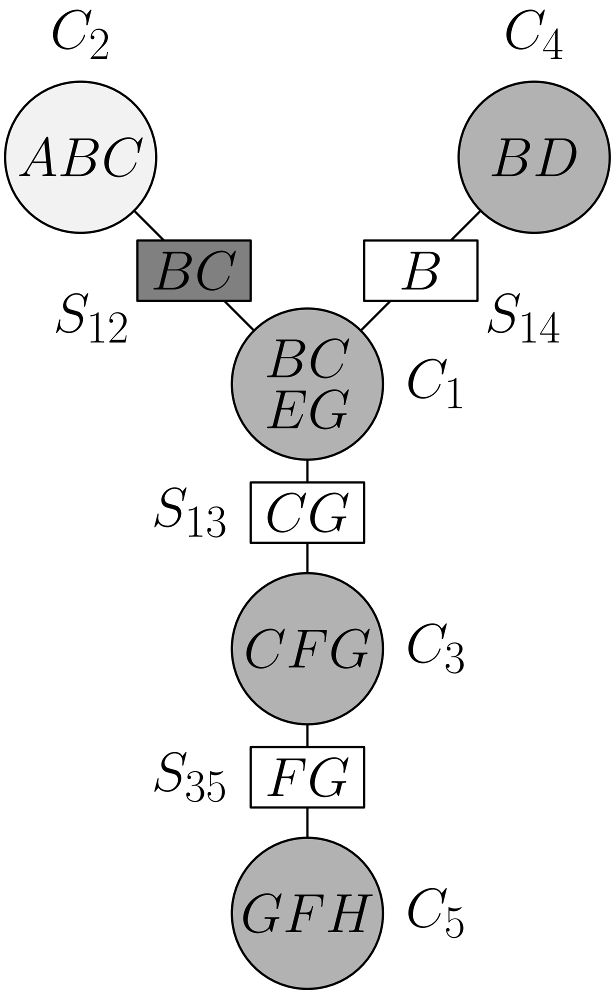
Option 2: Factor potentials chosen from the join-tree decomposition:

$$\begin{aligned}
\Psi_1(b,c,e,g) &= P(b,e \mid c,g) \\
\Psi_2(a,b,c) &= P(a \mid b,c) \\
\Psi_3(c,f,g) &= P(c \mid f,g) \\
\Psi_4(b,d) &= P(d \mid b) \\
\Psi_5(g,f,h) &= P(h,g,f)
\end{aligned}$$



Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

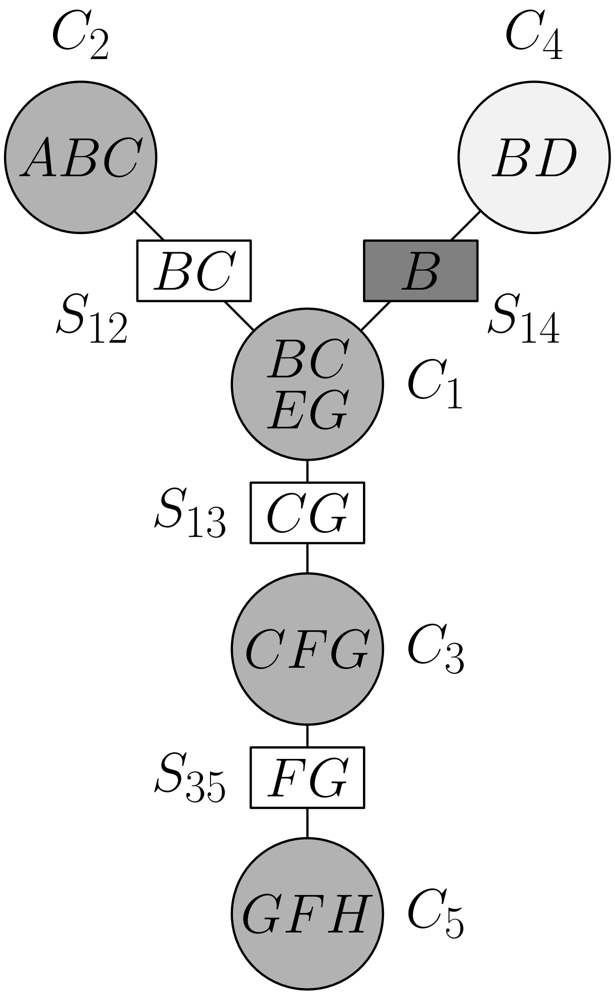


Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$A \perp\!\!\!\perp D, E, F, G, H \mid B, C$$

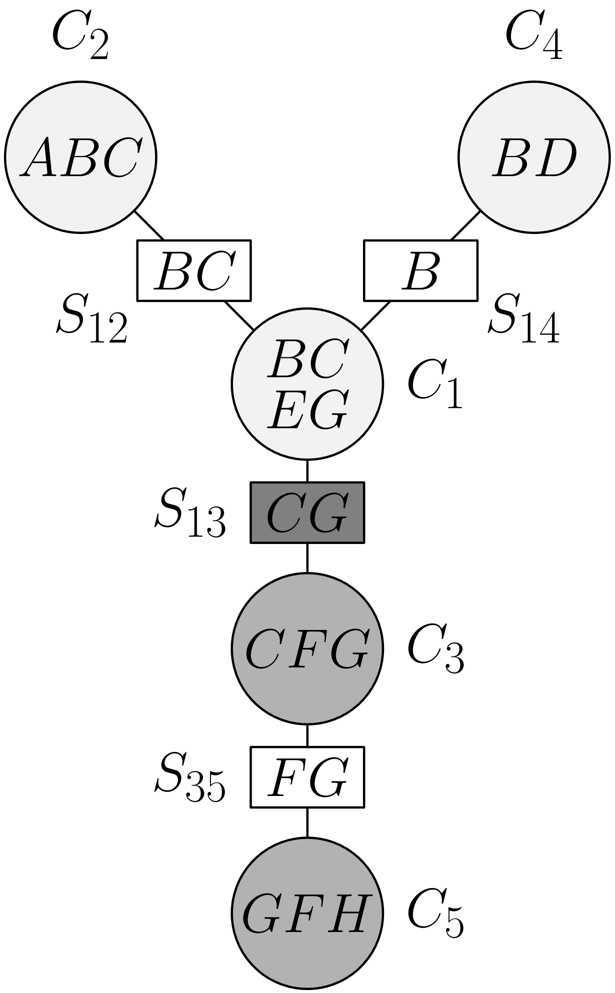
Example: Closer Look on Option 2: Separation in a Join-Tree



Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$A \perp\!\!\!\perp D, E, F, G, H \mid B, C$$
$$D \perp\!\!\!\perp A, C, E, F, G, H \mid B$$

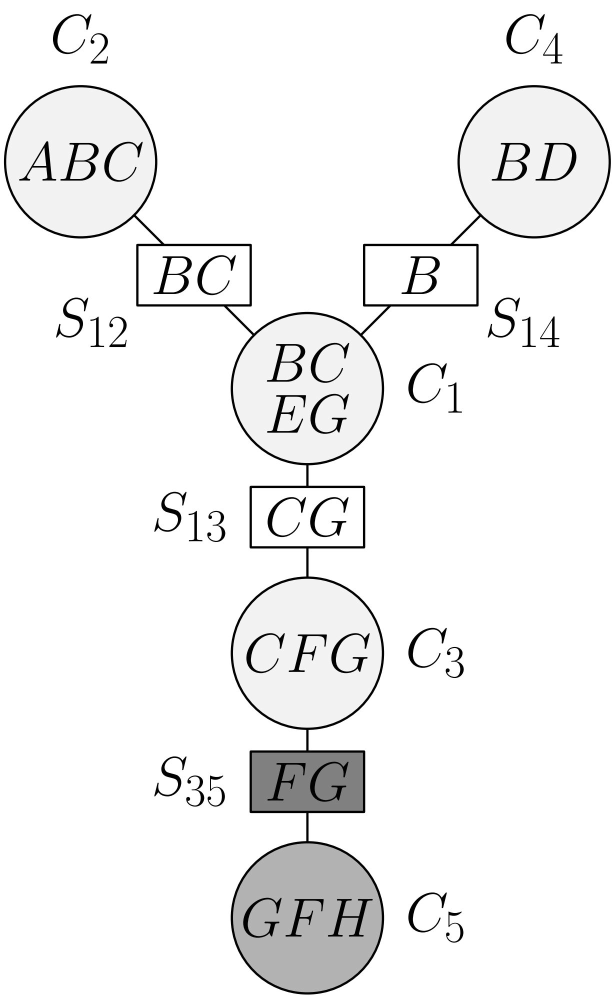


Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$A \perp\!\!\!\perp D, E, F, G, H \mid B, C$$
$$D \perp\!\!\!\perp A, C, E, F, G, H \mid B$$
$$A, B, E, D \perp\!\!\!\perp F, H \mid G, C$$

Example: Closer Look on Option 2: Separation in a Join-Tree



Encoded independence statements:

Given any separator, the variables in the cliques on one side become independent of the variables in the cliques on the other side.

$$A \perp\!\!\!\perp D, E, F, G, H \mid B, C$$
$$D \perp\!\!\!\perp A, C, E, F, G, H \mid B$$
$$A, B, E, D \perp\!\!\!\perp F, H \mid G, C$$
$$H \perp\!\!\!\perp A, B, C, D, E \mid F, G$$

Example: Closer Look on Option 2: Decomposition

The four separation statements translate into the following independence statements:

$$
\begin{aligned}
A \perp\!\!\!\perp D, E, F, G, H \mid B, C \quad &\Leftrightarrow P(A \mid B, C, D, E, F, G, H) &&= P(A \mid B, C) \\
D \perp\!\!\!\perp A, C, E, F, G, H \mid B \quad &\Rightarrow P(D \mid B, C, E, F, G, H) &&= P(D \mid B) \\
A, B, E, D \perp\!\!\!\perp F, H \mid G, C \quad &\Rightarrow P(B, E \mid G, C, F, H) &&= P(B, E \mid G, C) \\
H \perp\!\!\!\perp A, B, C, D, E \mid F, G \quad &\Rightarrow P(C \mid F, G, H) &&= P(C \mid F, G)
\end{aligned}
$$

According to the chain rule we always have the following relation:

$$
\begin{aligned}
P(A, B, C, D, E, F, G, H) = \; & P(A \mid B, C, D, E, F, G, H) \cdot \\
& P(D \mid B, C, E, F, G, H) \cdot \\
& P(B, E \mid C, F, G, H) \cdot \\
& P(C \mid F, G, H) \cdot \\
& P(F, G, H)
\end{aligned}
$$

Example: Closer Look on Option 2: Decomposition

The four separation statements translate into the following independence statements:

$$
\begin{array}{rlll}
A \perp\!\!\!\perp D, E, F, G, H \mid B, C & \Leftrightarrow P(A \mid B, C, D, E, F, G, H) & = P(A \mid B, C) \\
D \perp\!\!\!\perp A, C, E, F, G, H \mid B & \Rightarrow P(D \mid B, C, E, F, G, H) & = P(D \mid B) \\
A, B, E, D \perp\!\!\!\perp F, H \mid G, C & \Rightarrow P(B, E \mid G, C, F, H) & = P(B, E \mid G, C) \\
H \perp\!\!\!\perp A, B, C, D, E \mid F, G & \Rightarrow P(C \mid F, G, H) & = P(C \mid F, G)
\end{array}
$$

Exploiting the above independencies yields:

$$
\begin{aligned}
P(A, B, C, D, E, F, G, H) \; = \; & P(A \mid B, C) \cdot \\
& P(D \mid B) \cdot \\
& P(B, E \mid C, G) \cdot \\
& P(C \mid F, G) \cdot \\
& P(F, G, H)
\end{aligned}
$$

Example: Closer Look on Option 2: Decomposition

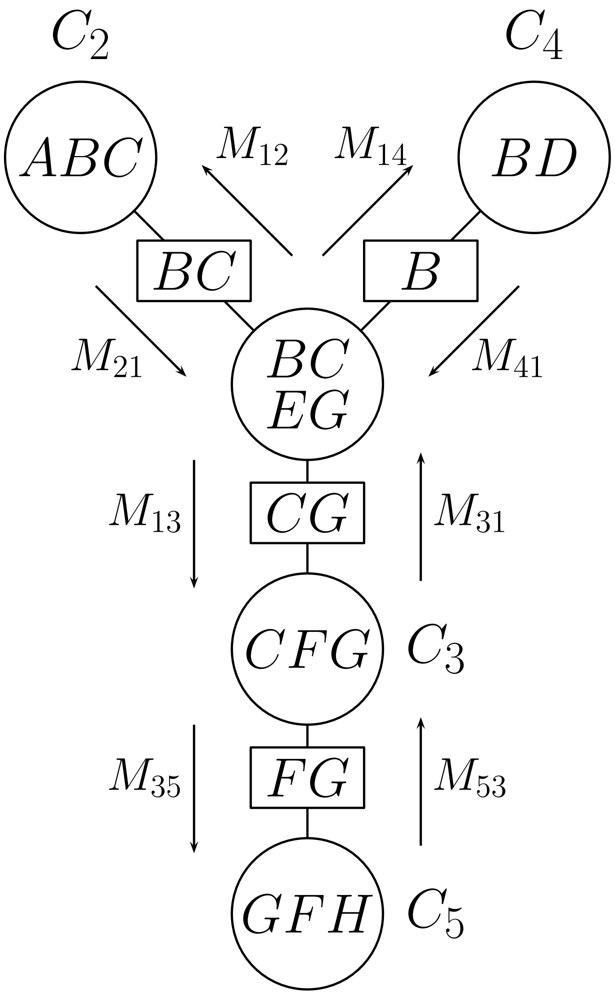
The four separation statements translate into the following independence statements:

$$\begin{array}{rll}
A \perp\!\!\!\perp D,E,F,G,H \mid B,C & \Leftrightarrow P(A \mid B,C,D,E,F,G,H) & = P(A \mid B,C) \\
D \perp\!\!\!\perp A,C,E,F,G,H \mid B & \Rightarrow P(D \mid B,C,E,F,G,H) & = P(D \mid B) \\
A,B,E,D \perp\!\!\!\perp F,H \mid G,C & \Rightarrow P(B,E \mid G,C,F,H) & = P(B,E \mid G,C) \\
H \perp\!\!\!\perp A,B,C,D,E \mid F,G & \Rightarrow P(C \mid F,G,H) & = P(C \mid F,G)
\end{array}$$

Getting rid of the conditions results in the final decomposition equation:

$$\begin{aligned}
P(A,B,C,D,E,F,G,H) &= P(A \,|\, B,C)P(D \,|\, B)P(B,E \,|\, C,G)P(C \,|\, F,G)P(F,G,H) \\
&= \frac{P(A,B,C)P(D,B)P(B,E,C,G)P(C,F,G)P(F,G,H)}{P(B,C)P(B)P(C,G)P(F,G)} \\
&= \frac{P(C_1)P(C_2)P(C_3)P(C_4)P(C_5)}{P(S_{12})P(S_{14})P(S_{13})P(S_{35})}
\end{aligned}$$

Example: Step 3: Messages to be sent for Propagation



According to the join-tree propagation algorithm, the probability distributions of all clique instantiations c_i is calculated as follows:

$$P(c_i) \;\propto\; \Psi_i(c_i) \prod_{j=1}^{q} M_{ji}(s_{ij})$$

Spelt out for our example, we get:

$$\begin{aligned}
P(c_1) &= P(b,c,e,g) = \Psi_1(b,c,e,g) \cdot M_{21}(b,c) \cdot M_{31}(c,g) \cdot M_{41}(b) \\
P(c_2) &= P(a,b,c) \propto \Psi_2(a,b,c) \cdot M_{12}(b,c) \\
P(c_3) &= P(c,f,g) \propto \Psi_3(c,f,g) \cdot M_{13}(c,g) \cdot M_{53}(f,g) \\
P(c_4) &= P(b,d) \propto \Psi_4(b,d) \cdot M_{14}(b) \\
P(c_5) &= P(f,g,h) \propto \Psi_5(f,g,h) \cdot M_{35}(f,g)
\end{aligned}$$

The $\propto$-symbol indicates that the right-hand side may not add up to one. In that case we just normalize.

Example: Step 3: Message Computation Order

The structure of the join-tree imposes a partial ordering according to which the messages need to be computed:

$$M_{41}(b) = \sum_{d} \Psi_4(b, d)$$

$$M_{53}(f, g) = \sum_{h} \Psi_5(f, g, h)$$

$$M_{21}(b, c) = \sum_{a} \Psi_2(a, b, c)$$

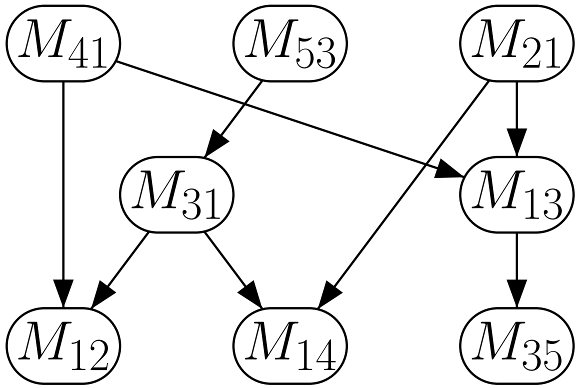
$$M_{31}(c, g) = \sum_{f} \Psi_3(c, f, g) M_{53}(f, g)$$

$$M_{13}(c, g) = \sum_{b,e} \Psi_1(b, c, e, g) M_{21}(b, c) M_{41}(b)$$

$$M_{12}(b, c) = \sum_{e,g} \Psi_2(b, c, e, g) M_{31}(c, g) M_{41}(b)$$

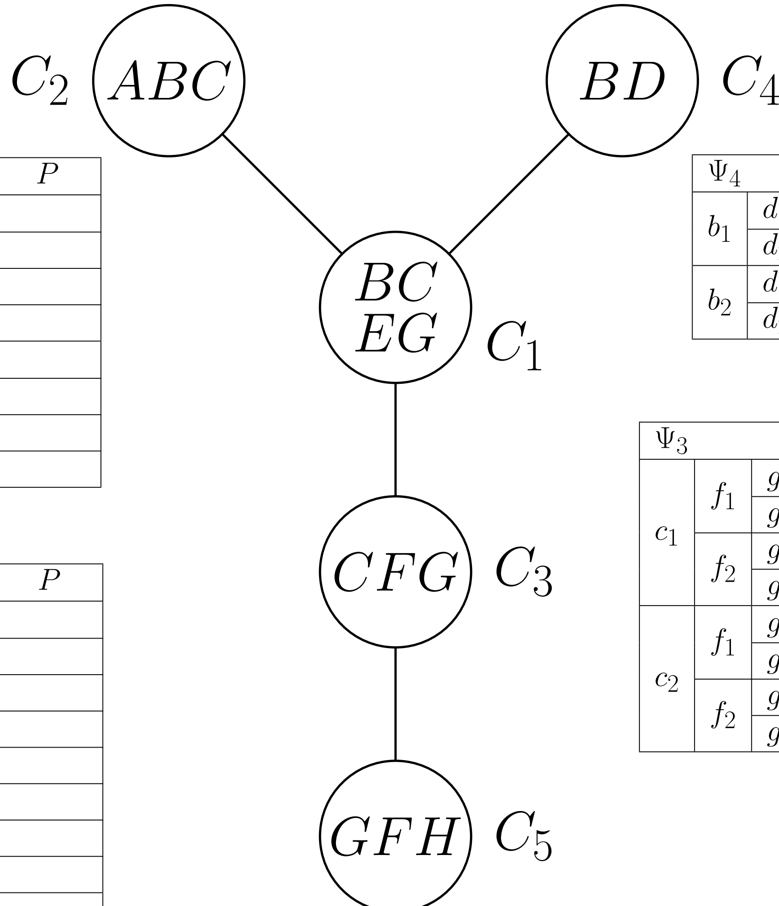
$$M_{14}(b) = \sum_{c,e,g} \Psi_1(b, c, e, g) M_{21}(b, c) M_{31}(c, g)$$

$$M_{35}(f, g) = \sum_{c} \Psi_3(c, f, g) M_{13}(c, g)$$



Arrows represent is-needed-for relations. Messages on the same level can be computed in any order. Messages are computed level-wise from top to bottom.

Example: Step 3: Initialization (Potential Layouts)



Ψ_2				P
a_1	b_1	c_1		
		c_2		
	b_2	c_1		
		c_2		
a_2	b_1	c_1		
		c_2		
	b_2	c_1		
		c_2		

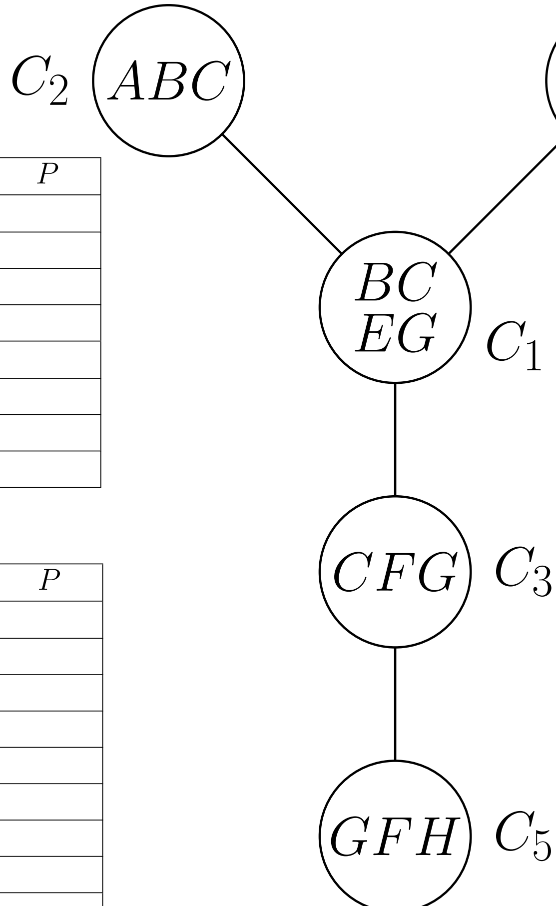
Ψ_4			P
b_1	d_1		
	d_2		
b_2	d_1		
	d_2		

Ψ_3				P
c_1	f_1	g_1		
		g_2		
	f_2	g_1		
		g_2		
c_2	f_1	g_1		
		g_2		
	f_2	g_1		
		g_2		

Ψ_1					P
b_1	c_1	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		
	c_2	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		
b_2	c_1	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		
	c_2	e_1	g_1		
			g_2		
		e_2	g_1		
			g_2		

Ψ_5				P
f_1	g_1	h_1		
		h_2		
	g_2	h_1		
		h_2		
f_2	g_1	h_1		
		h_2		
	g_2	h_1		
		h_2		

Example: Step 3: Initialization (Potential Values)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

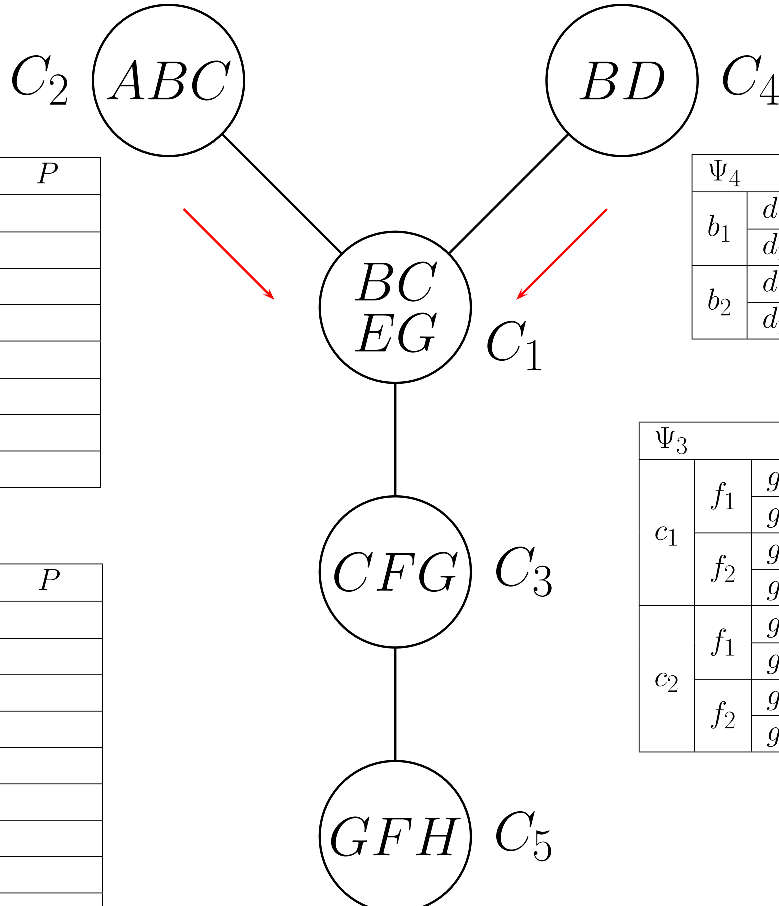
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

Example: Step 3: Initialization (Sending Messages)



$$M_{21} = \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array}$$
$$\qquad\;\; \left(0.06, 0.10, 0.40, 0.44\right)$$

$$M_{41} = \begin{array}{cc} b_1 & b_2 \end{array}$$
$$\qquad\;\; \left(1, 1\right)$$

Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

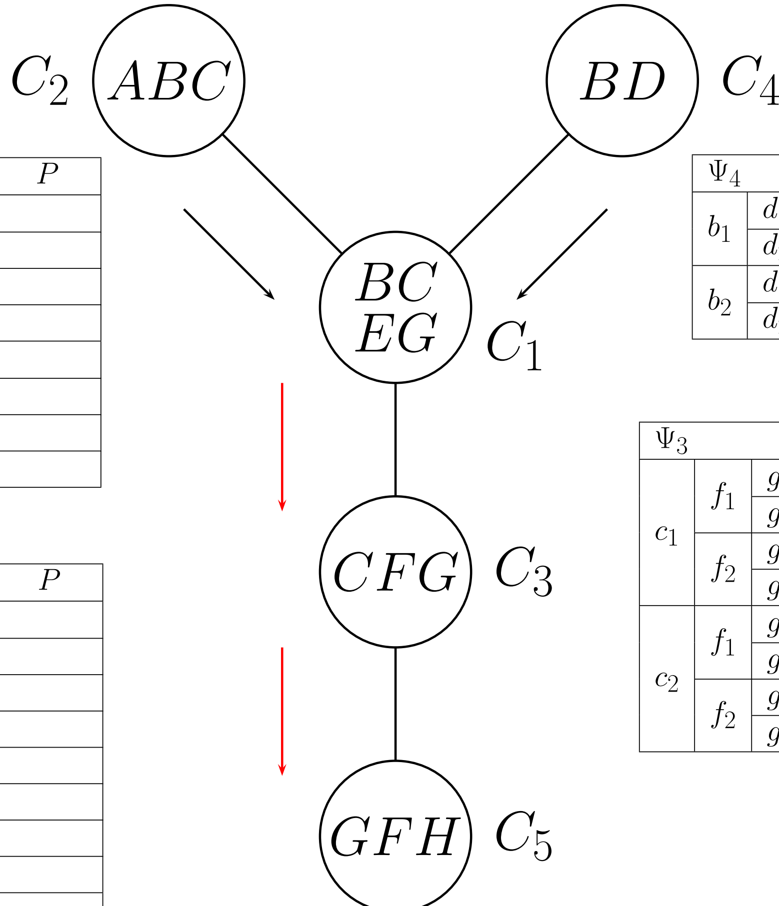
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

Example: Step 3: Initialization (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

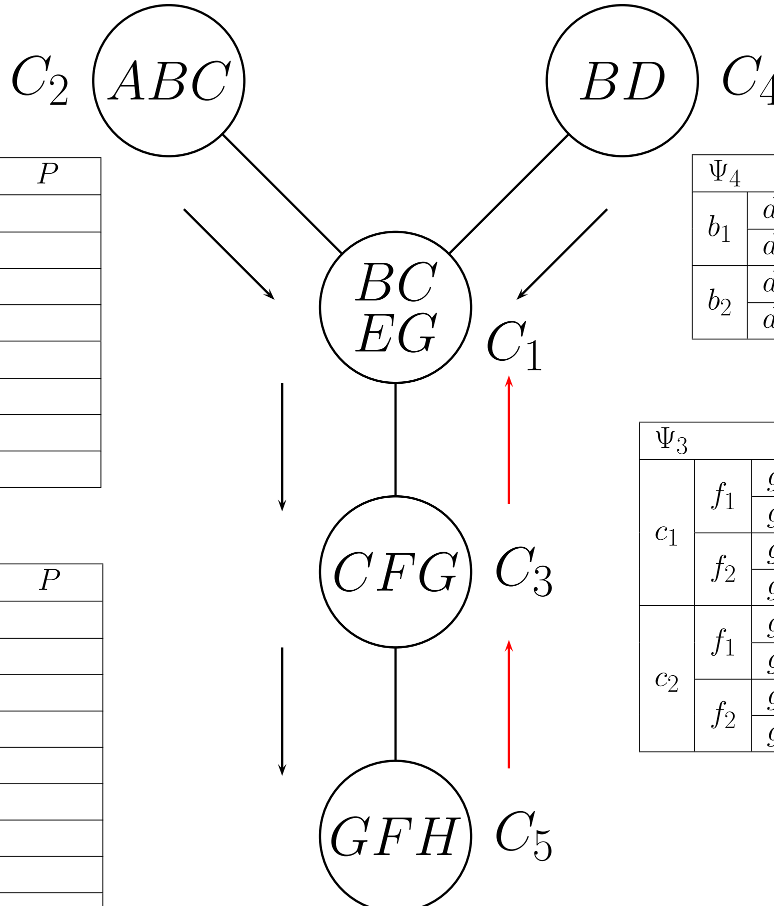
$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 1, & 1 \end{array}\right) \end{array}$$

$$M_{13} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.254, & 0.206, & 0.290, & 0.250 \end{array}\right) \end{array}$$

$$M_{35} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.14, & 0.12, & 0.40, & 0.33 \end{array}\right) \end{array}$$

Example: Step 3: Initialization (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

$$M_{21} = \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \\ \end{array} \left(0.06,\ 0.10,\ 0.40,\ 0.44 \right)$$

$$M_{41} = \begin{array}{cc} b_1 & b_2 \end{array} \left(1,\ 1 \right)$$

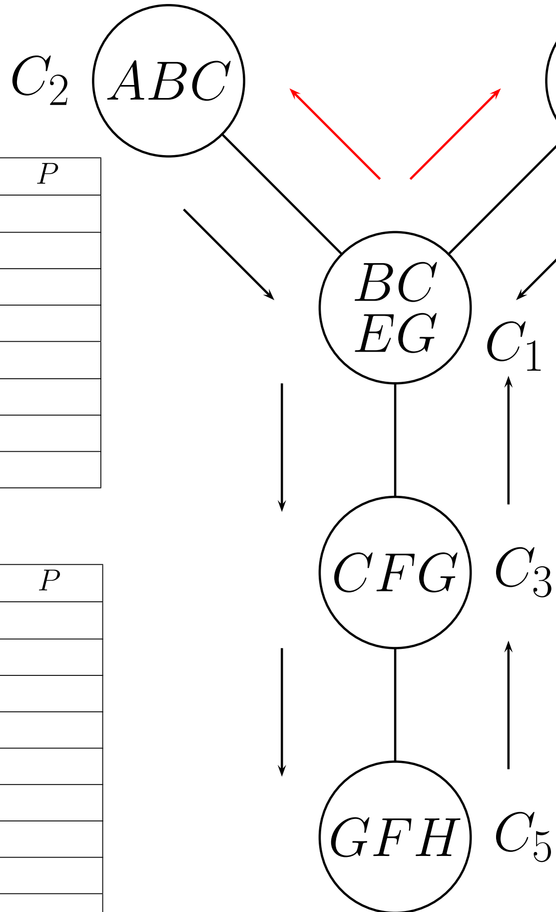
$$M_{13} = \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \left(0.254,\ 0.206,\ 0.290,\ 0.250 \right)$$

$$M_{35} = \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \left(0.14,\ 0.12,\ 0.40,\ 0.33 \right)$$

$$M_{53} = \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \left(1,\ 1,\ 1,\ 1 \right)$$

$$M_{31} = \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \left(1,\ 1,\ 1,\ 1 \right)$$

Example: Step 3: Initialization (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0.8	
	g_2	h_1	0.5	
		h_2	0.5	
f_2	g_1	h_1	0.4	
		h_2	0.6	
	g_2	h_1	0.7	
		h_2	0.3	

$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 1, & 1 \end{array}\right) \end{array}$$

$$M_{13} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.254, & 0.206, & 0.290, & 0.250 \end{array}\right) \end{array}$$

$$M_{35} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.14, & 0.12, & 0.40, & 0.33 \end{array}\right) \end{array}$$

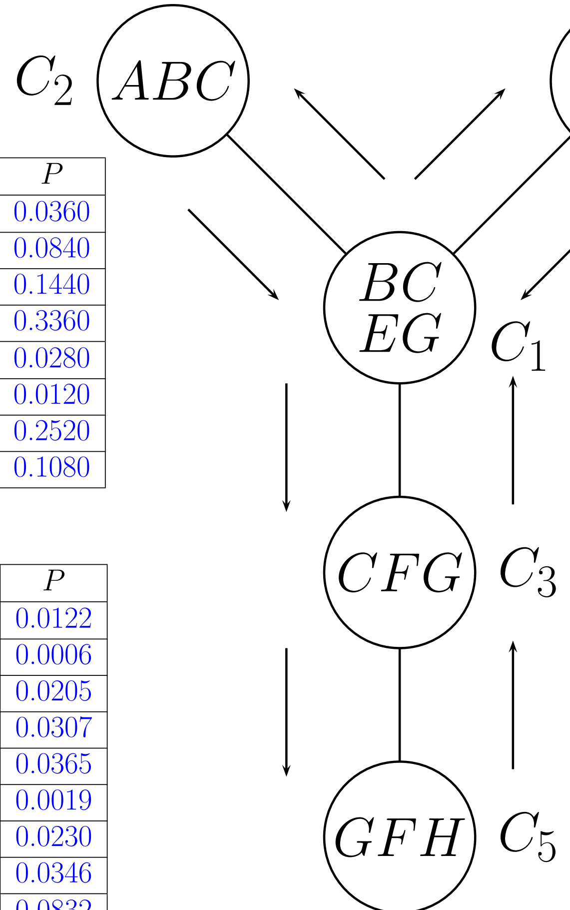
$$M_{53} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 1, & 1, & 1, & 1 \end{array}\right) \end{array}$$

$$M_{31} = \begin{array}{c} \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 1, & 1, & 1, & 1 \end{array}\right) \end{array}$$

$$M_{12} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 1, & 1, & 1, & 1 \end{array}\right) \end{array}$$

$$M_{14} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 0.16, & 0.84 \end{array}\right) \end{array}$$

Example: Step 3: Initialization Complete



Ψ_2				P
a_1	b_1	c_1	0.036	0.0360
		c_2	0.084	0.0840
	b_2	c_1	0.144	0.1440
		c_2	0.336	0.3360
a_2	b_1	c_1	0.028	0.0280
		c_2	0.012	0.0120
	b_2	c_1	0.252	0.2520
		c_2	0.108	0.1080

Ψ_4			P
b_1	d_1	0.4	0.0640
	d_2	0.6	0.0960
b_2	d_1	0.7	0.5880
	d_2	0.3	0.2520

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	0.0122
			g_2	0.010	0.0006
		e_2	g_1	0.320	0.0205
			g_2	0.480	0.0307
	c_2	e_1	g_1	0.380	0.0365
			g_2	0.020	0.0019
		e_2	g_1	0.240	0.0230
			g_2	0.360	0.0346
b_2	c_1	e_1	g_1	0.210	0.0832
			g_2	0.090	0.0356
		e_2	g_1	0.350	0.1386
			g_2	0.350	0.1386
	c_2	e_1	g_1	0.070	0.0311
			g_2	0.030	0.0133
		e_2	g_1	0.450	0.1998
			g_2	0.450	0.1998

Ψ_3				P
c_1	f_1	g_1	0.1	0.0254
		g_2	0.1	0.0206
	f_2	g_1	0.9	0.2290
		g_2	0.9	0.1850
c_2	f_1	g_1	0.4	0.1162
		g_2	0.4	0.0998
	f_2	g_1	0.6	0.1742
		g_2	0.6	0.1498

Ψ_5				P
f_1	g_1	h_1	0.2	0.0283
		h_2	0.8	0.1133
	g_2	h_1	0.5	0.0602
		h_2	0.5	0.0602
f_2	g_1	h_1	0.4	0.1613
		h_2	0.6	0.2419
	g_2	h_1	0.7	0.2344
		h_2	0.3	0.1004

$$M_{21} = \begin{array}{c} \begin{matrix} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{matrix} \\ (0.06,\ 0.10,\ 0.40,\ 0.44) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{matrix} b_1 & b_2 \end{matrix} \\ (1,\ 1) \end{array}$$

$$M_{13} = \begin{array}{c} \begin{matrix} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{matrix} \\ (0.254,\ 0.206,\ 0.290,\ 0.250) \end{array}$$

$$M_{35} = \begin{array}{c} \begin{matrix} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{matrix} \\ (0.14,\ 0.12,\ 0.40,\ 0.33) \end{array}$$

$$M_{53} = \begin{array}{c} \begin{matrix} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{matrix} \\ (1,\ 1,\ 1,\ 1) \end{array}$$

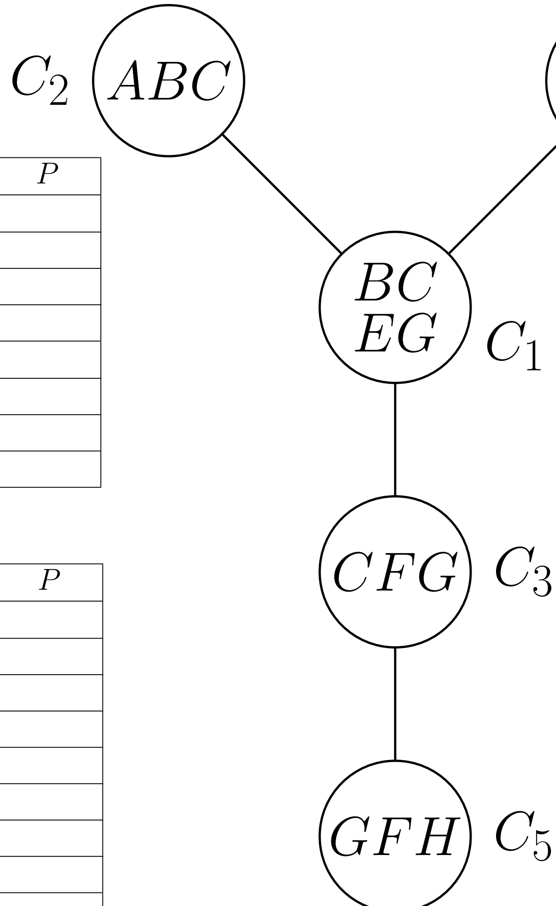
$$M_{31} = \begin{array}{c} \begin{matrix} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \end{matrix} \\ (1,\ 1,\ 1,\ 1) \end{array}$$

$$M_{12} = \begin{array}{c} \begin{matrix} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{matrix} \\ (1,\ 1,\ 1,\ 1) \end{array}$$

$$M_{14} = \begin{array}{c} \begin{matrix} b_1 & b_2 \end{matrix} \\ (0.16,\ 0.84) \end{array}$$

P	A	B	C	D	E	F	G	H
$\cdot_1$	**0.6000**	**0.1600**	**0.4600**	**0.6520**	**0.2144**	**0.2620**	**0.5448**	**0.4842**
$\cdot_2$	**0.4000**	**0.8400**	**0.4500**	**0.3480**	**0.7856**	**0.7380**	**0.4552**	**0.5158**

Example: Step 4: Evidence $H = h_1$ (Altering Potentials)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

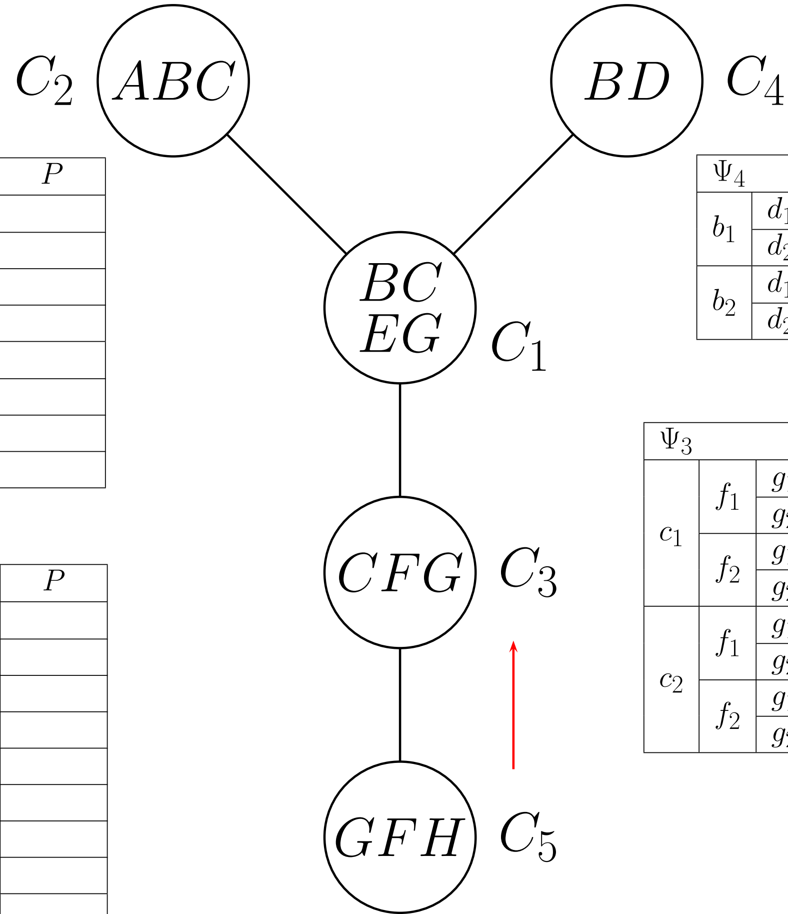
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	**0**	
	g_2	h_1	0.5	
		h_2	**0**	
f_2	g_1	h_1	0.4	
		h_2	**0**	
	g_2	h_1	0.7	
		h_2	**0**	

Example: Step 4: Evidence $H = h_1$ (Sending Messages)



$$M_{53} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.2, & 0.5, & 0.4, & 0.7 \end{array} \right) \end{array}$$

Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

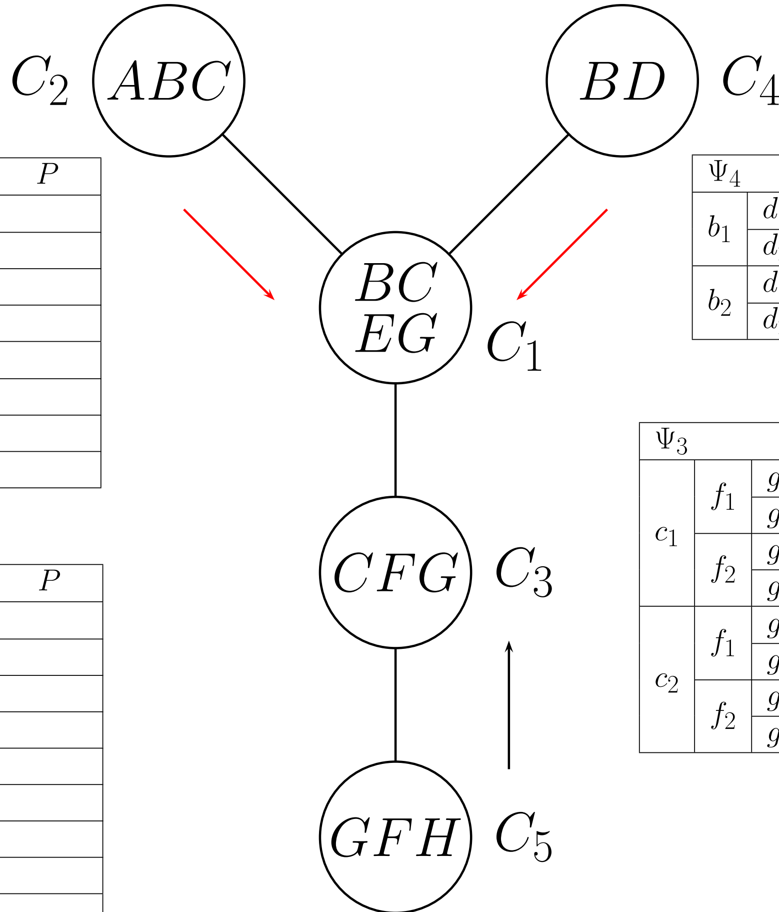
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

Example: Step 4: Evidence $H = h_1$ (Sending Messages)



$$M_{53} = \begin{array}{c} \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \end{array} \\ \left(\begin{array}{cccc} 0.2\,, & 0.5\,, & 0.4\,, & 0.7 \end{array}\right) \end{array}$$

$$M_{21} = \begin{array}{c} \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \end{array} \\ \left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right) \end{array}$$

$$M_{41} = \begin{array}{c} \begin{array}{cc} b_1 & b_2 \end{array} \\ \left(\begin{array}{cc} 1\,, & 1 \end{array}\right) \end{array}$$

Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

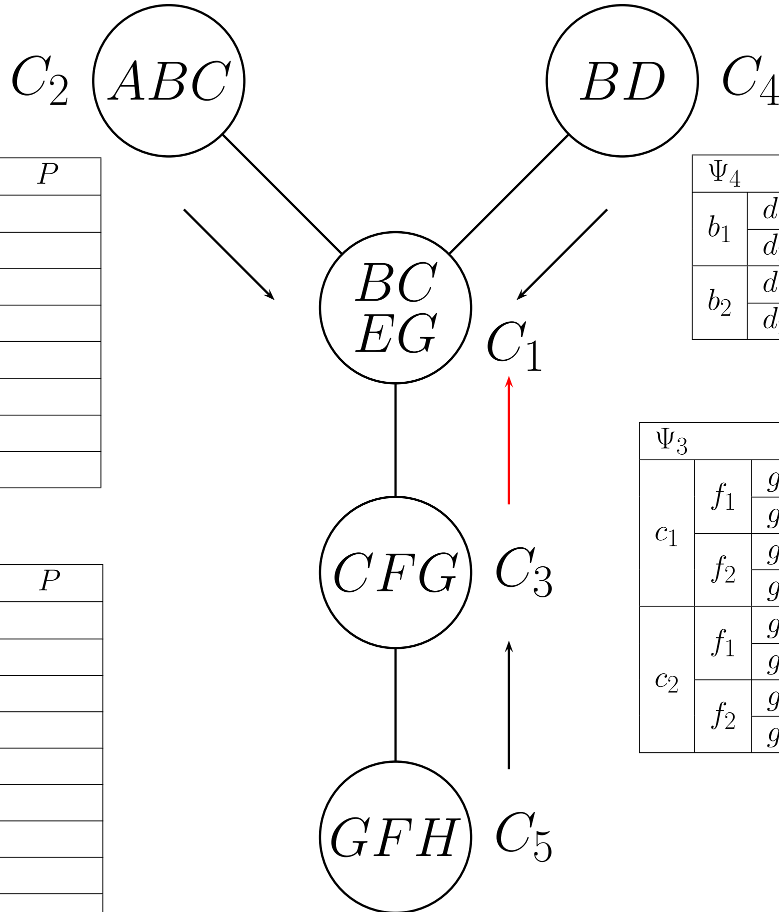
Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

Example: Step 4: Evidence $H = h_1$ (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

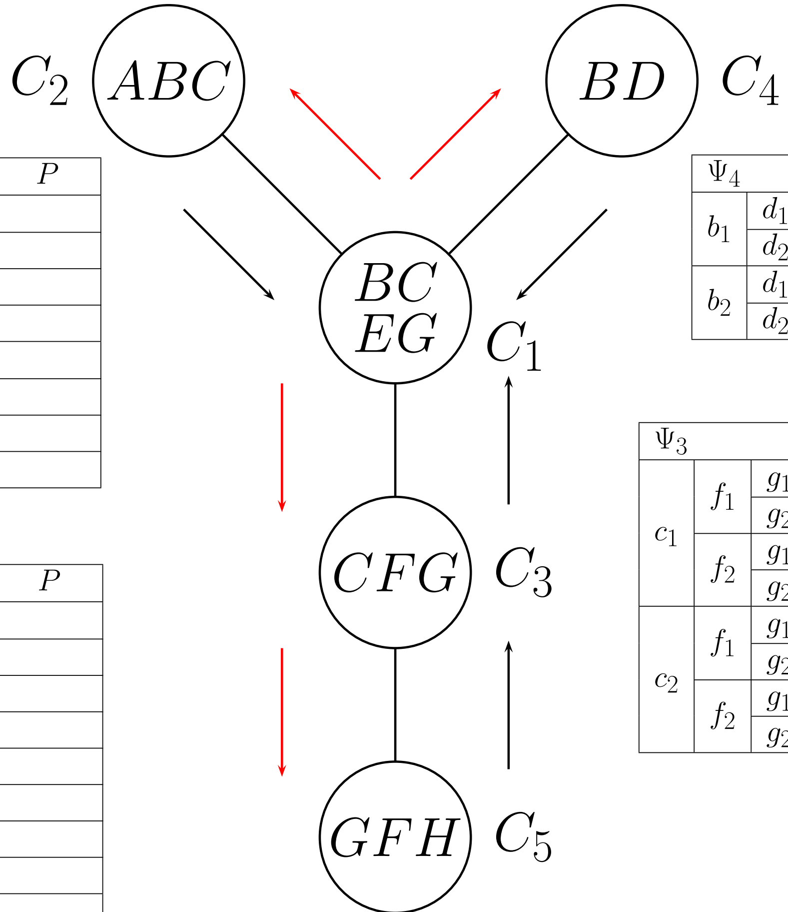
$$M_{53} = \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \\ \multicolumn{4}{c}{\left(\begin{array}{cccc} 0.2\,, & 0.5\,, & 0.4\,, & 0.7 \end{array}\right)} \end{array}$$

$$M_{21} = \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \\ \multicolumn{4}{c}{\left(\begin{array}{cccc} 0.06, & 0.10, & 0.40, & 0.44 \end{array}\right)} \end{array}$$

$$M_{41} = \begin{array}{cc} b_1 & b_2 \\ \multicolumn{2}{c}{\left(\begin{array}{cc} 1, & 1 \end{array}\right)} \end{array}$$

$$M_{31} = \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \\ \multicolumn{4}{c}{\left(\begin{array}{cccc} 0.38, & 0.68, & 0.32, & 0.62 \end{array}\right)} \end{array}$$

Example: Step 4: Evidence $H = h_1$ (Sending Messages)



Ψ_2				P
a_1	b_1	c_1	0.036	
		c_2	0.084	
	b_2	c_1	0.144	
		c_2	0.336	
a_2	b_1	c_1	0.028	
		c_2	0.012	
	b_2	c_1	0.252	
		c_2	0.108	

Ψ_4			P
b_1	d_1	0.4	
	d_2	0.6	
b_2	d_1	0.7	
	d_2	0.3	

Ψ_3				P
c_1	f_1	g_1	0.1	
		g_2	0.1	
	f_2	g_1	0.9	
		g_2	0.9	
c_2	f_1	g_1	0.4	
		g_2	0.4	
	f_2	g_1	0.6	
		g_2	0.6	

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	
			g_2	0.010	
		e_2	g_1	0.320	
			g_2	0.480	
	c_2	e_1	g_1	0.380	
			g_2	0.020	
		e_2	g_1	0.240	
			g_2	0.360	
b_2	c_1	e_1	g_1	0.210	
			g_2	0.090	
		e_2	g_1	0.350	
			g_2	0.350	
	c_2	e_1	g_1	0.070	
			g_2	0.030	
		e_2	g_1	0.450	
			g_2	0.450	

Ψ_5				P
f_1	g_1	h_1	0.2	
		h_2	0	
	g_2	h_1	0.5	
		h_2	0	
f_2	g_1	h_1	0.4	
		h_2	0	
	g_2	h_1	0.7	
		h_2	0	

$$M_{53} = \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \\ \end{array} \left(0.2\ ,\ 0.5\ ,\ 0.4\ ,\ 0.7 \right)$$

$$M_{21} = \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \\ \end{array} \left(0.06, 0.10, 0.40, 0.44 \right)$$

$$M_{41} = \begin{array}{cc} b_1 & b_2 \\ \end{array} \left(1, 1 \right)$$

$$M_{31} = \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \\ \end{array} \left(0.38, 0.68, 0.32, 0.62 \right)$$

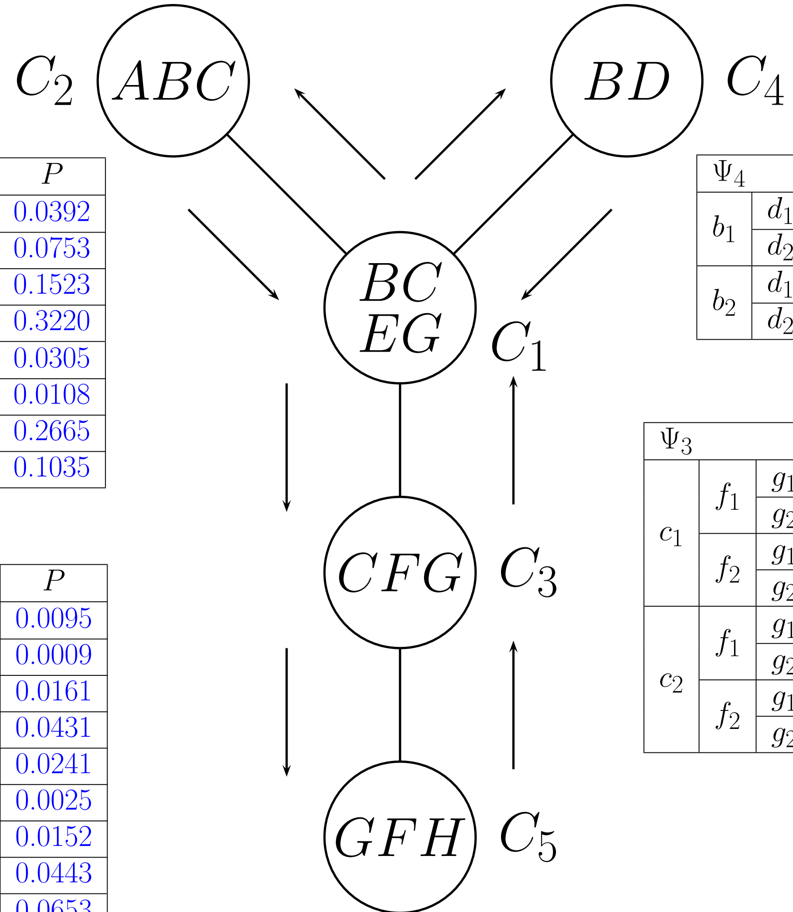
$$M_{12} = \begin{array}{cccc} b_1,c_1 & b_1,c_2 & b_2,c_1 & b_2,c_2 \\ \end{array} \left(0.527, 0.434, 0.512, 0.464 \right)$$

$$M_{14} = \begin{array}{cc} b_1 & b_2 \\ \end{array} \left(0.075, 0.409 \right)$$

$$M_{13} = \begin{array}{cccc} c_1,g_1 & c_1,g_2 & c_2,g_1 & c_2,g_2 \\ \end{array} \left(0.254, 0.206, 0.290, 0.250 \right)$$

$$M_{35} = \begin{array}{cccc} f_1,g_1 & f_1,g_2 & f_2,g_1 & f_2,g_2 \\ \end{array} \left(0.14, 0.12, 0.40, 0.33 \right)$$

Example: Step 4: Evidence $H = h_1$ Incorporated



Ψ_2				P
a_1	b_1	c_1	0.036	0.0392
		c_2	0.084	0.0753
	b_2	c_1	0.144	0.1523
		c_2	0.336	0.3220
a_2	b_1	c_1	0.028	0.0305
		c_2	0.012	0.0108
	b_2	c_1	0.252	0.2665
		c_2	0.108	0.1035

Ψ_4			P
b_1	d_1	0.4	0.0623
	d_2	0.6	0.0934
b_2	d_1	0.7	0.5910
	d_2	0.3	0.2533

Ψ_3				P
c_1	f_1	g_1	0.1	0.0105
		g_2	0.1	0.0212
	f_2	g_1	0.9	0.1892
		g_2	0.9	0.2675
c_2	f_1	g_1	0.4	0.0480
		g_2	0.4	0.1031
	f_2	g_1	0.6	0.1440
		g_2	0.6	0.2165

Ψ_1					P
b_1	c_1	e_1	g_1	0.190	0.0095
			g_2	0.010	0.0009
		e_2	g_1	0.320	0.0161
			g_2	0.480	0.0431
	c_2	e_1	g_1	0.380	0.0241
			g_2	0.020	0.0025
		e_2	g_1	0.240	0.0152
			g_2	0.360	0.0443
b_2	c_1	e_1	g_1	0.210	0.0653
			g_2	0.090	0.0501
		e_2	g_1	0.350	0.1088
			g_2	0.350	0.1947
	c_2	e_1	g_1	0.070	0.0205
			g_2	0.030	0.0171
		e_2	g_1	0.450	0.1321
			g_2	0.450	0.2559

Ψ_5				P
f_1	g_1	h_1	0.2	0.0585
		h_2	0	0
	g_2	h_1	0.5	0.1243
		h_2	0	0
f_2	g_1	h_1	0.4	0.3331
		h_2	0	0
	g_2	h_1	0.7	0.4841
		h_2	0	0

$$M_{53} = \left(\overset{f_1,g_1}{0.2}, \overset{f_1,g_2}{0.5}, \overset{f_2,g_1}{0.4}, \overset{f_2,g_2}{0.7}\right)$$

$$M_{21} = \left(\overset{b_1,c_1}{0.06}, \overset{b_1,c_2}{0.10}, \overset{b_2,c_1}{0.40}, \overset{b_2,c_2}{0.44}\right)$$

$$M_{41} = \left(\overset{b_1}{1}, \overset{b_2}{1}\right)$$

$$M_{31} = \left(\overset{c_1,g_1}{0.38}, \overset{c_1,g_2}{0.68}, \overset{c_2,g_1}{0.32}, \overset{c_2,g_2}{0.62}\right)$$

$$M_{12} = \left(\overset{b_1,c_1}{0.527}, \overset{b_1,c_2}{0.434}, \overset{b_2,c_1}{0.512}, \overset{b_2,c_2}{0.464}\right)$$

$$M_{14} = \left(\overset{b_1}{0.075}, \overset{b_2}{0.409}\right)$$

$$M_{13} = \left(\overset{c_1,g_1}{0.254}, \overset{c_1,g_2}{0.206}, \overset{c_2,g_1}{0.290}, \overset{c_2,g_2}{0.250}\right)$$

$$M_{35} = \left(\overset{f_1,g_1}{0.14}, \overset{f_1,g_2}{0.12}, \overset{f_2,g_1}{0.40}, \overset{f_2,g_2}{0.33}\right)$$

P	A	B	C	D	E	F	G	H
$\cdot_1$	**0.5888**	**0.1557**	**0.4884**	**0.6533**	**0.1899**	**0.1828**	**0.3916**	**1.0000**
$\cdot_2$	**0.4112**	**0.8443**	**0.5116**	**0.3467**	**0.8101**	**0.8172**	**0.6084**	**0.0000**

Summary

There are several exact inference methods such as variable elimination, clique tree propagation or recursive conditioning. These algorithms have complexity that is exponential with networks tree width. Exact inference is NP-hard.

In very large applications it is necessary to introduce topological structural constraints or restrictions on conditional probabilities, i.e. bounded variance algorithms.

Manual Building of Bayes Networks

Building Graphical Models: Causal Modeling

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

causal model of given domain

↓ **heuristics!**

conditional independence graph

↓ formally provable

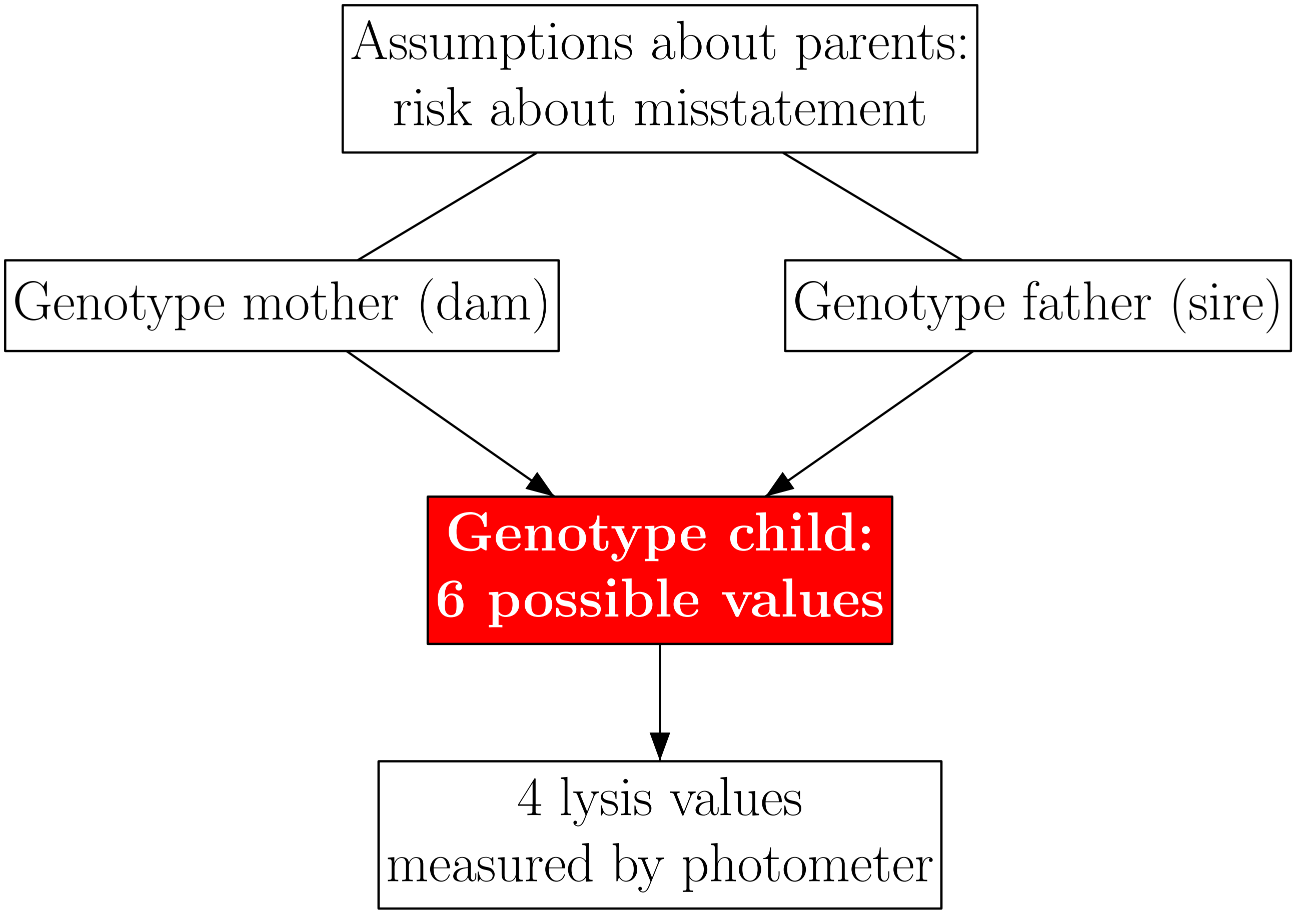
decomposition of the distribution

↓ formally provable

evidence propagation scheme

Problem: strong assumptions about the statistical effects of causal relations.

Nevertheless this approach often yields usable graphical models.



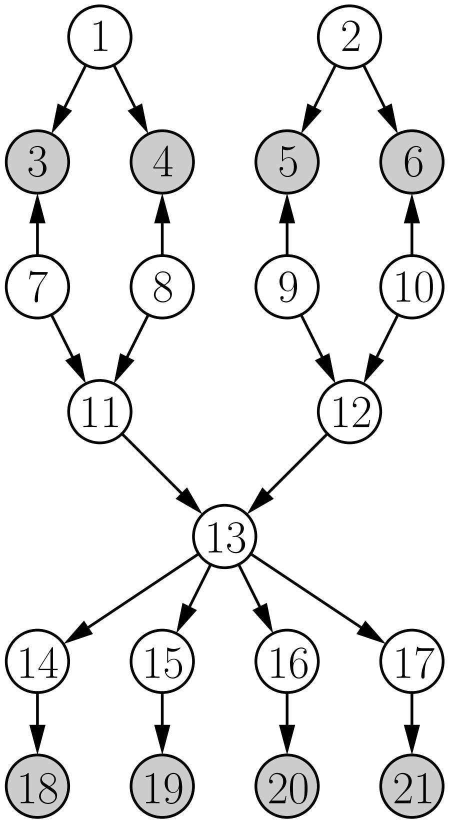
Reliability of databases

Inheritance rules

Blood group determination

See paper on our website.

Danish Jersey Cattle Blood Type Determination



21 attributes:

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

Example 1: Genotype Determination of Danish Jersey Cattle

Full 21-dimensional domain has $2^6 \cdot 3^{10} \cdot 6 \cdot 8^4 = 92\,876\,046\,336$ possible states.

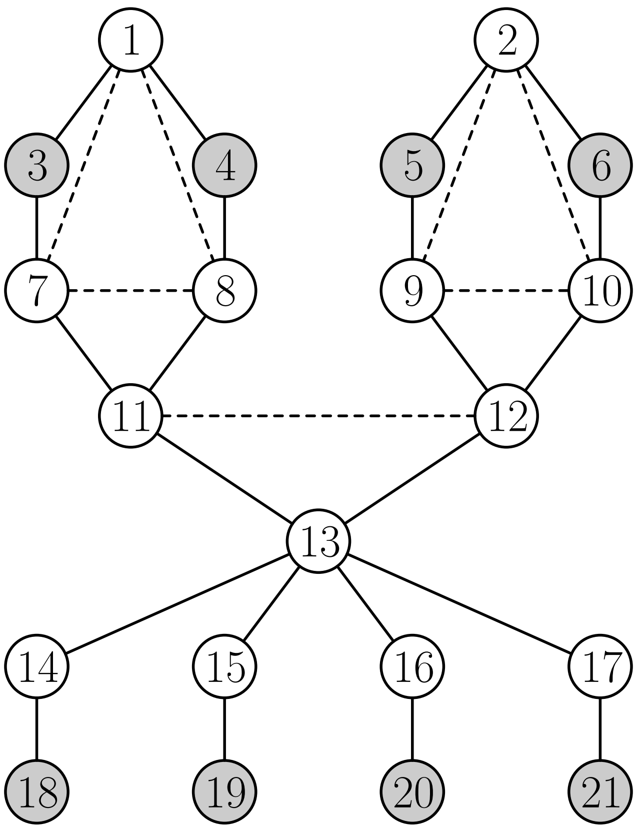
Bayesian network requires only 306 conditional probabilities.

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

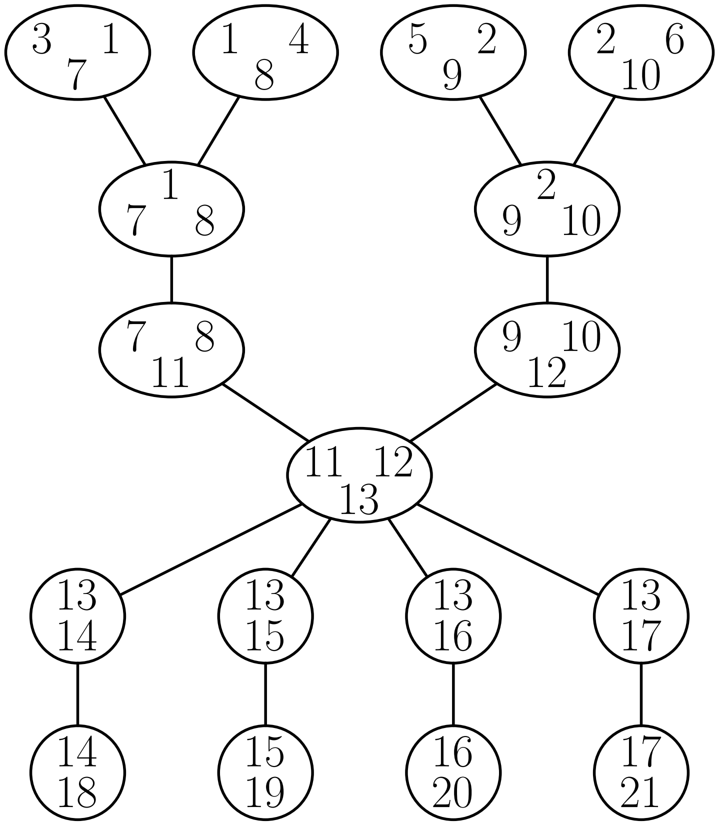
The probabilities are acquired from human domain experts or estimated from historical data.

Example 1: Genotype Determination of Danish Jersey Cattle



moral graph

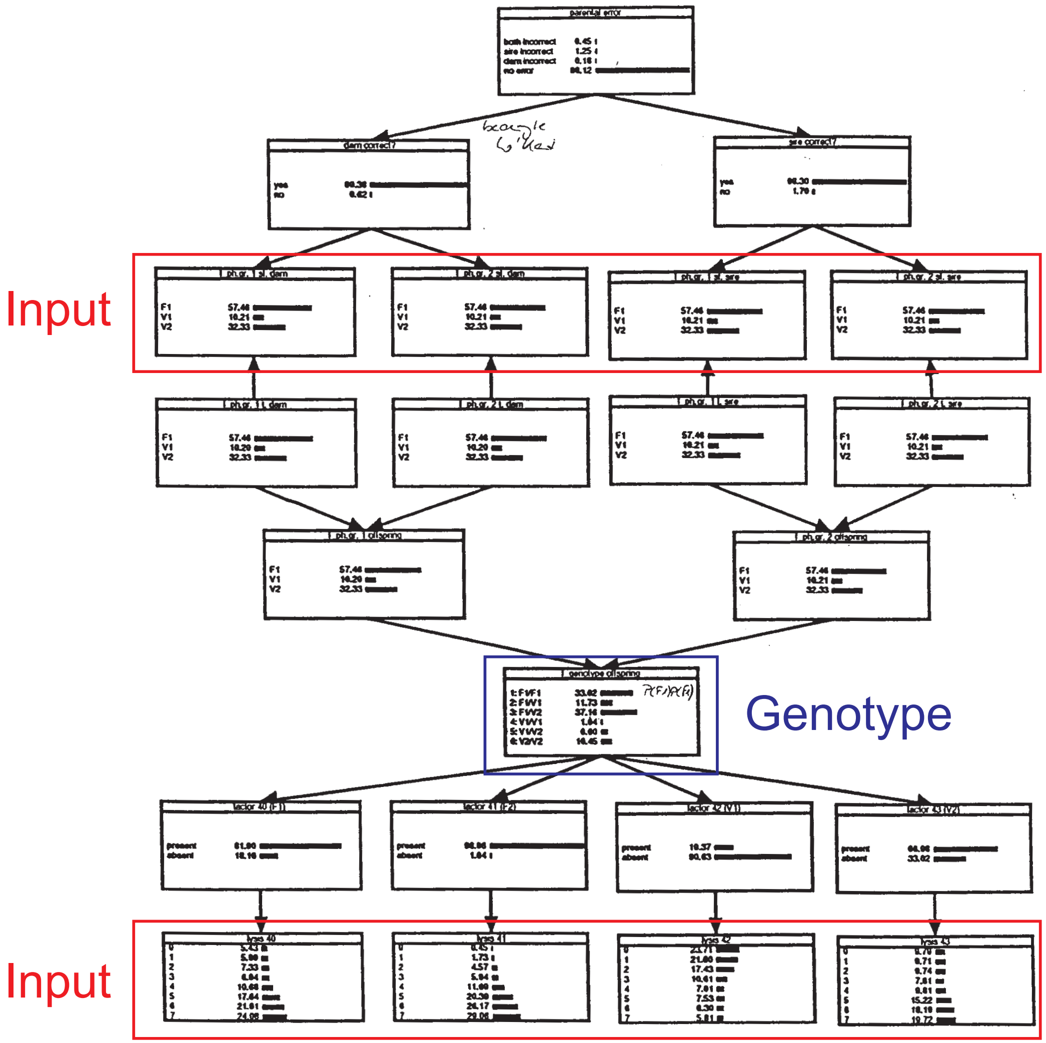
(already triangulated)



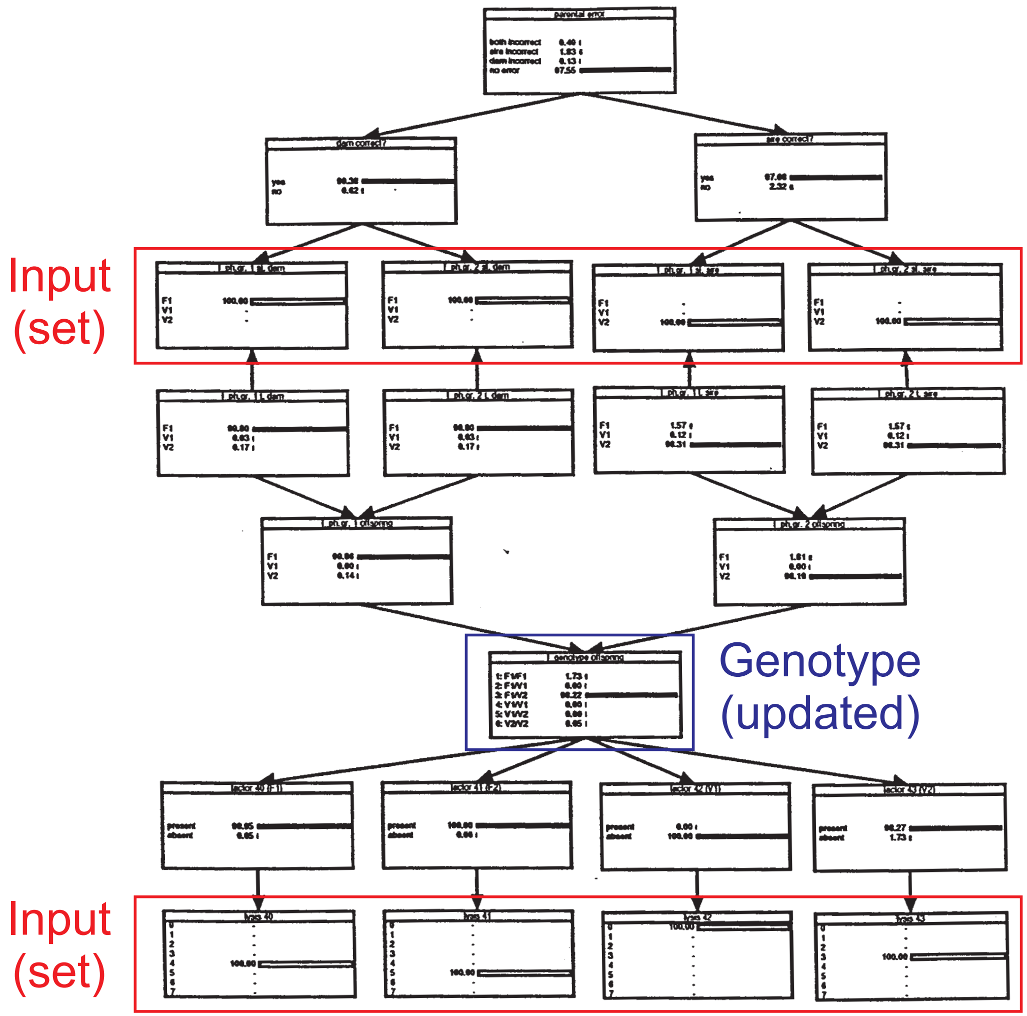
join tree

Example 1: Genotype Determination of Danish Jersey Cattle

Marginal distributions before setting evidence:



Conditional distributions given evidence in the input variables:



Example 2: Item Planning at Volkswagen

Strategy of the VW Group

Marketing strategy	Vehicle specification by clients	Bestsellers defined by manufacturer
Complexity	Huge number of variants	Small number of variants

Vehicle specification

Equipment	fastback	2,8 l, 150 kW	Type Alpha	4	leather	...
Group	car body type	engine	radio	doors	seat cover	...

Example 2: Model "Golf"

Approx. 200 equipment groups

2 to 50 items per group

Therefore more than 2^{200} possible vehicle specifications

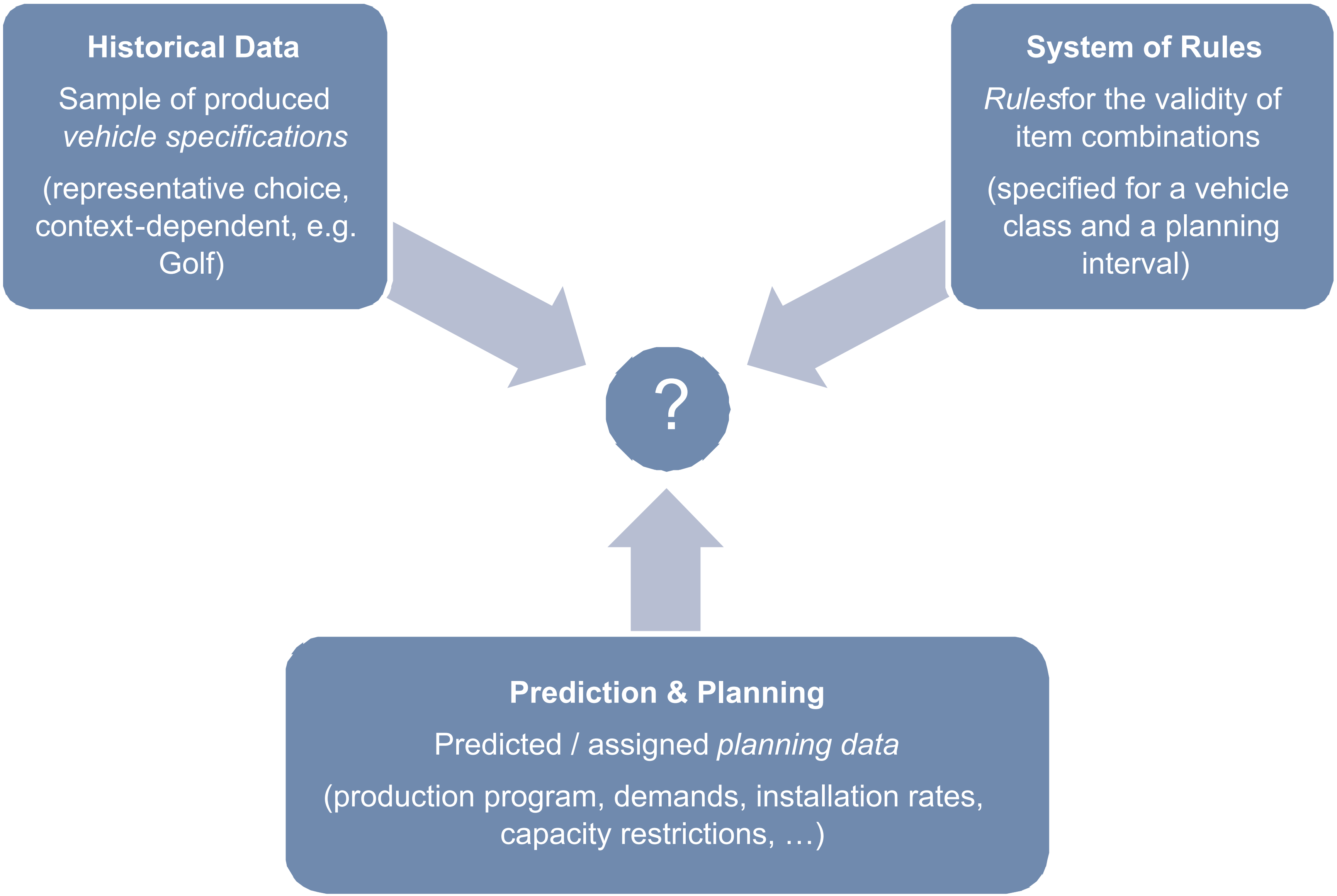
Choice of valid specifications is constrained by a rule system
(10000 technical rules, plus marketing and production rules)

Example of technical rules:

If Engine=e_1 **then** Transmission=t_3

If Engine=e_4 and Heating=h_2 **then** Generator $\in \{g_3, g_4, g_5\}$

Problem Representation



Complexity of the Planning Problem

Equipment table

	Engine	Transmission	Heating	Generator	$\cdots$
1	e_1	t_3	h_1	g_1	$\cdots$
2	e_2	t_4	h_3	g_5	$\cdots$
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
100000	e_7	t_1	h_3	g_2	$\cdots$

Installation rates

Engine	Transmission	Heating	Generator	$\cdots$	Rate
e_1	t_1	h_1	g_1	$\cdots$	0.0000012
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

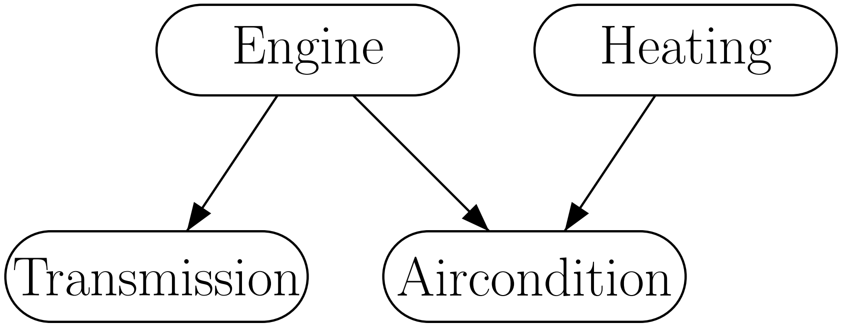
Result is a 200-dimensional, finite probability space

$$P(\text{Engine} = e_1, \text{Transmission} = t_3) = ?$$

$$P(\text{Heating} = h_1 \mid \text{Generator} = g_3) = ?$$

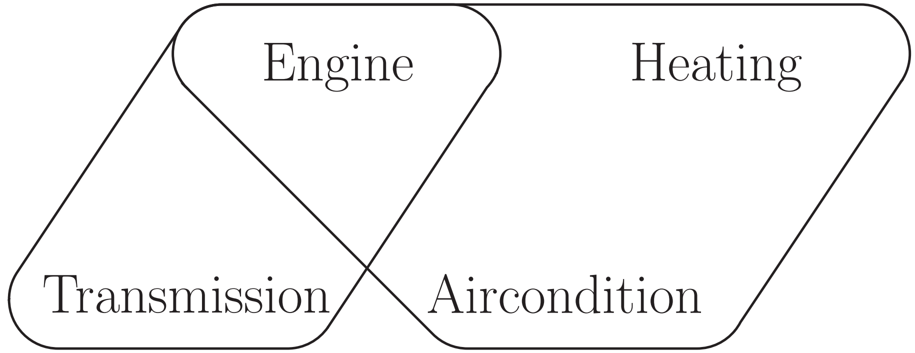
Problem of complexity!

Solution: Decomposition into Subspaces



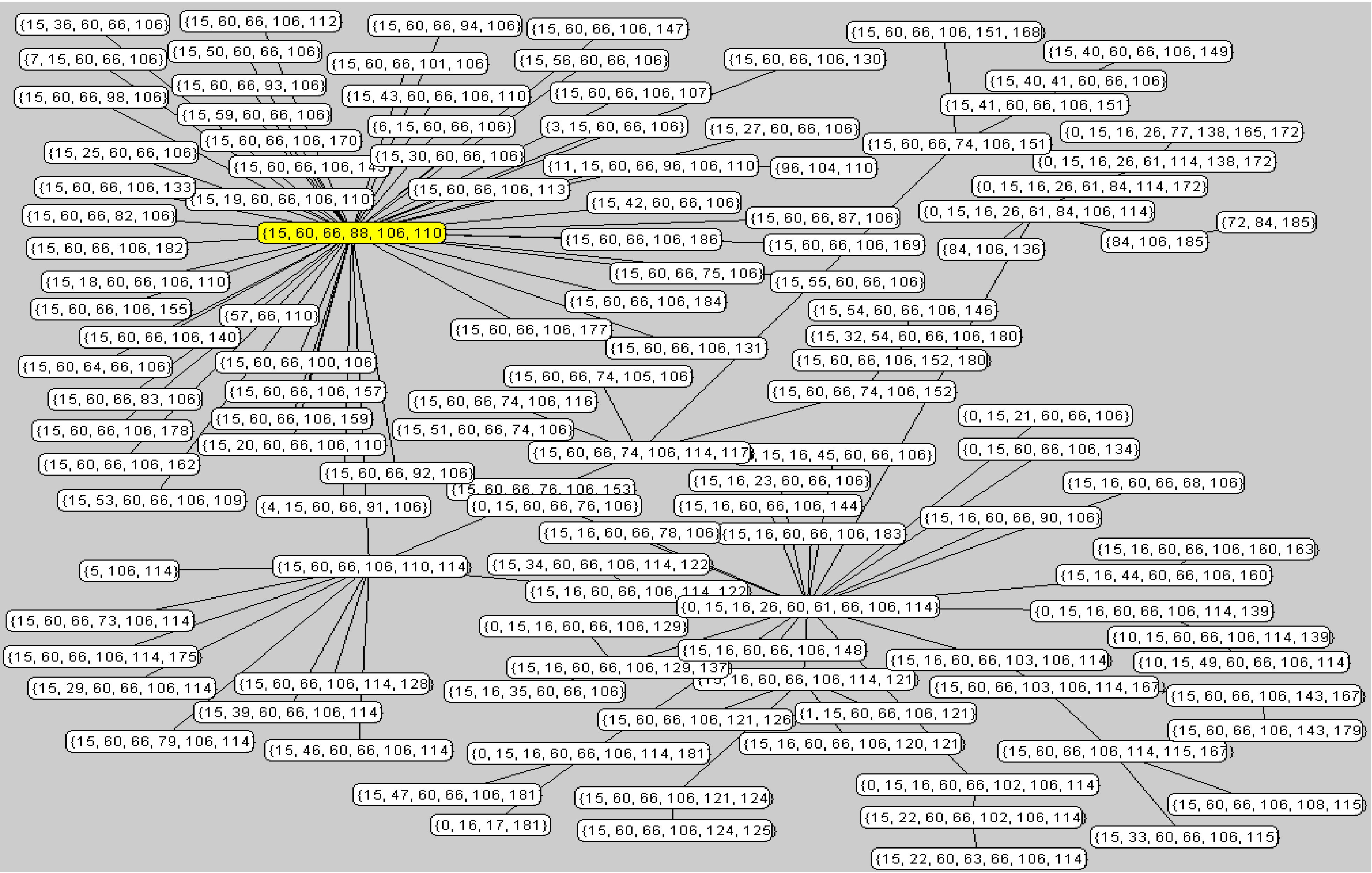
Bayesian Network

$$P(E,H,T,A) = P(A \mid E,H,T) \cdot P(T \mid E,H) \cdot P(E \mid H) \cdot P(H)$$
$$\overset{\text{here}}{=} P(A \mid E,H) \cdot P(T \mid E) \cdot P(E) \cdot P(H)$$



Hypergraph Decomposition

Clique Tree of the VW Bora



Typical Planning Operation: Focusing

Application:

- **Compute item demand**
  Calculation of installation rates of equipment combinations
- **Simulation**
  Analyze customer requirements (e. g. of persons having ordered a navigation system for a VW Polo)

Input: Equipment combinations

Operation: Compute

- the conditional network distribution and
- the probabilities of the specified equipment combinations.

Implementation and Deployment

Project leader: Intelligent System Consulting (Gebhardt)

Client server system

Server on 6–8 maschines

Quadcore platform

Terabyte hard drive

Java, Linux, Oracle

WebSphere application server

Software used daily worldwide

20 developers

5000 Bayesian networks are currently used

Building Bayes Networks:

Parameter Learning

Learning Naive Bayes Classifier

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

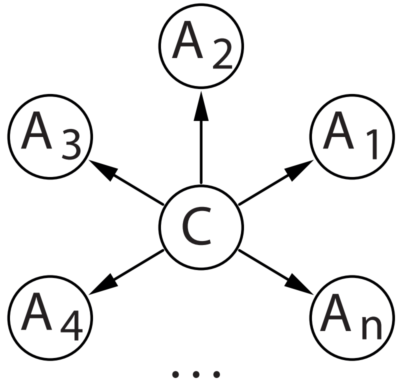
Desired: Good values for the numeric parameters of the model.

Example: Naive Bayes Classifiers

A naive Bayes classifier is a Bayesian network with star-like structure.

The class attribute is the only unconditional 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 $\mathrm{dom}(C) = \{c_1, \ldots, c_{n_C}\}$, 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

Consequence: An optimal classifier should assign the **most probable class**.

This argument does not change if we take descriptive attributes into account.

- Let $U = \{A_1, \ldots, A_m\}$ be a set of descriptive attributes with domains $\mathrm{dom}(A_k)$, $1 \leq k \leq m$.
- Let $A_1 = a_1, \ldots, A_m = a_m$ be an instantiation of the descriptive attributes.
- An optimal classifier should assign the class c_i for which

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

Problem: We cannot store a class (or the class probabilities) for every possible instantiation $A_1 = a_1, \ldots, A_m = a_m$ 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' classifiers: Compute the class probabilities as

$$P(C = c_i \mid A_1 = a_1, \ldots, A_m = a_m) =$$

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

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:

$$P(C = c_i \mid \omega) = \frac{P(A_1 = a_1, \ldots, A_m = a_m \mid C = c_i) \cdot P(C = c_i)}{P(A_1 = a_1, \ldots, A_m = a_m)}$$

$\leftarrow p_0$
abbrev. for the
normalizing constant

Chain Rule of Probability:

$$P(C = c_i \mid \omega) = \frac{P(C = c_i)}{p_0} \cdot \prod_{k=1}^{m} P(A_k = a_k \mid A_1 = a_1, \ldots, A_{k-1} = a_{k-1}, C = c_i)$$

Conditional Independence Assumption:

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

Naive Bayes Classifiers (continued)

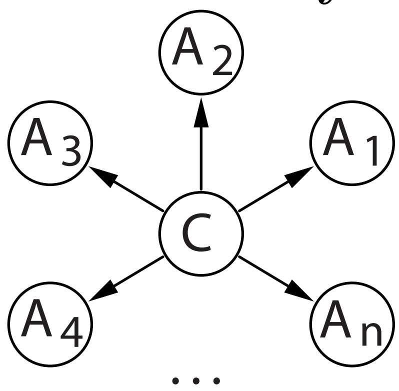
Consequence: Manageable amount of data to store.
Store distributions $P(C = c_i)$ and $\forall 1 \leq k \leq m : P(A_k = a_k \mid C = c_i)$.

Classification: Compute for all classes c_i

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

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

Relation to Bayesian Networks:



Decomposition formula:

$$\begin{aligned} & P(C = c_i, A1 = a_1, \ldots, A_n = a_n) \\ &= P(C = c_i) \cdot \prod_{j=1}^{n} P(A_j = a_j | C = c_i) \end{aligned}$$

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

- γ 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.

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

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

$\xi = \#(C = c_i)$: 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
μ	36.3	47.8
σ^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

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

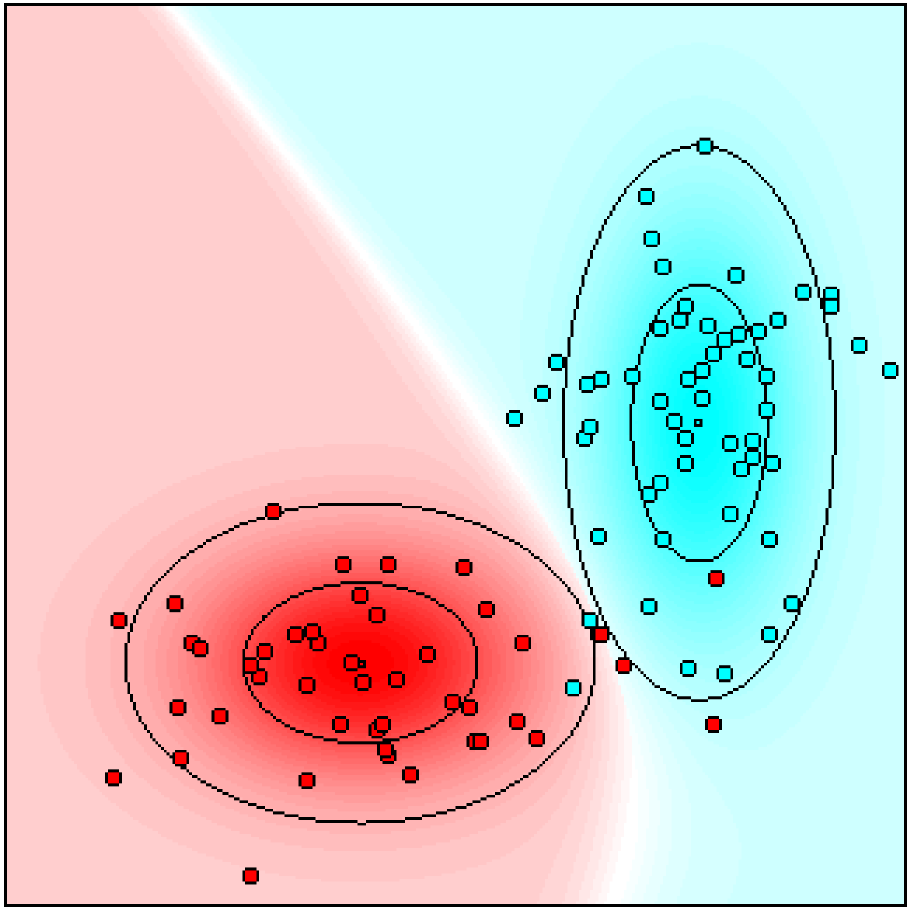
$$\begin{aligned}
&P(\text{Drug } \mathbf{B} \mid \text{male, 61, normal}) \\
&\quad = \;\; c_1 \cdot P(\text{Drug B}) \cdot P(\text{male} \mid \text{Drug B}) \cdot P(61 \mid \text{Drug B}) \cdot P(\text{normal} \mid \text{Drug B}) \\
&\quad \approx \;\; c_1 \cdot 0.5 \cdot 0.5 \cdot 0.017120 \cdot 0.5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

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

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

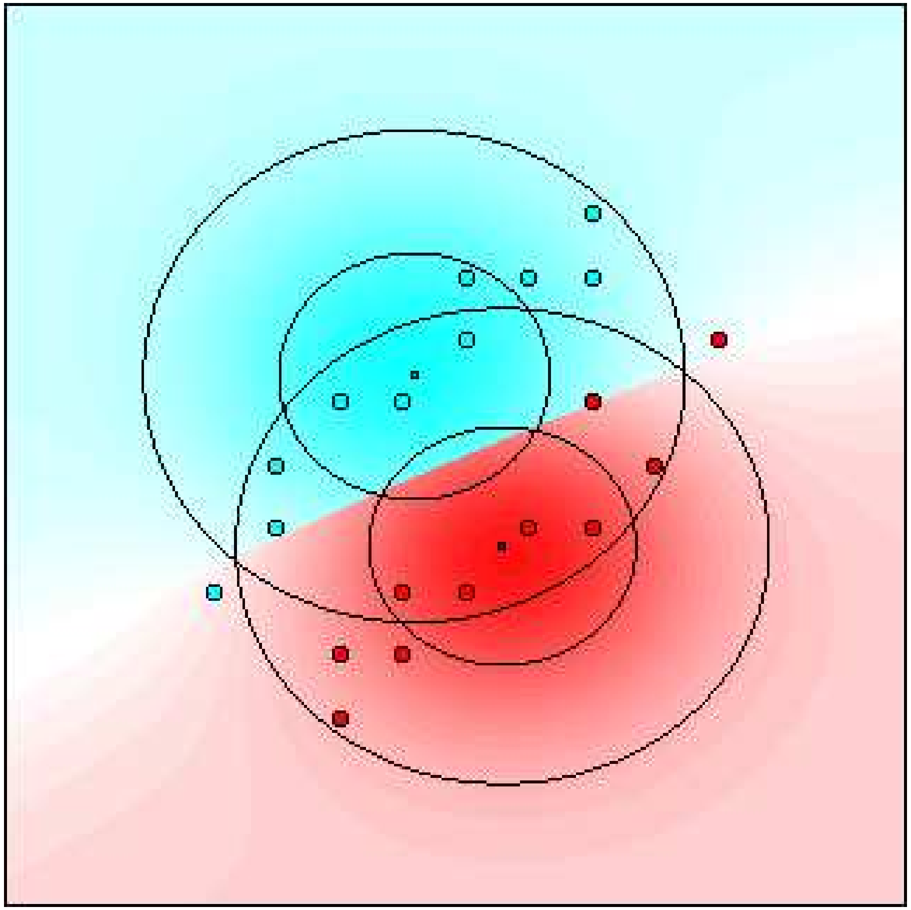
Naive Bayes Classifiers: Simple Example 2

- 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

- 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.

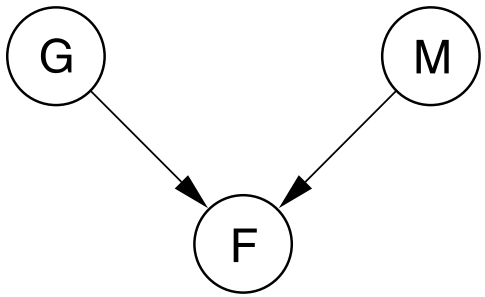


Naive Bayes Classifier

Learning the parameters of a Graphical Model

$A_1 = \mathsf{G}$	$Q_{11} = \phi$
$a_{11} = \mathsf{g}$	
$a_{12} = \overline{\mathsf{g}}$	

$A_2 = \mathsf{M}$	$Q_{21} = \phi$
$a_{12} = \mathsf{m}$	
$a_{22} = \overline{\mathsf{m}}$	



$A_3 = \mathsf{F}$	$Q_{31} = (\mathsf{g}, \mathsf{m})$	$Q_{32} = (\mathsf{g}, \overline{\mathsf{m}})$	$Q_{33} = (\overline{\mathsf{g}}, \mathsf{m})$	$Q_{34} = (\overline{\mathsf{g}}, \overline{\mathsf{m}})$
$a_{31} = \mathsf{f}$				
$a_{32} = \overline{\mathsf{f}}$				

$$V = \{\mathsf{G}, \mathsf{M}, \mathsf{F}\}$$
$$\mathrm{dom}(\mathsf{G}) = \{\mathsf{g}, \overline{\mathsf{g}}\}$$
$$\mathrm{dom}(\mathsf{M}) = \{\mathsf{m}, \overline{\mathsf{m}}\}$$
$$\mathrm{dom}(\mathsf{F}) = \{\mathsf{f}, \overline{\mathsf{f}}\}$$

The potential tables' layout is determined by the graph structure.

The parameters (i. e. the table entries) can be easily estimated from the database, e. g.:

$$\hat{P}(\mathsf{f} \mid \mathsf{g}, \mathsf{m}) = \frac{\hat{P}(\mathsf{f}, \mathsf{g}, \mathsf{m})}{\hat{P}(\mathsf{g}, \mathsf{m})} = \frac{\frac{\#(\mathsf{g},\mathsf{m},\mathsf{f})}{|D|}}{\frac{\#(\mathsf{g},\mathsf{m})}{|D|}} = \frac{\#(\mathsf{g}, \mathsf{m}, \mathsf{f})}{\#(\mathsf{g}, \mathsf{m})}$$

Learning the parameters of a Graphical Model

Flu G Malaria M Fever F	$\overline{\mathsf{g}}$ $\overline{\mathsf{m}}$ $\overline{\mathsf{f}}$	$\overline{\mathsf{g}}$ $\overline{\mathsf{m}}$ f	$\overline{\mathsf{g}}$ m $\overline{\mathsf{f}}$	$\overline{\mathsf{g}}$ m f	g $\overline{\mathsf{m}}$ $\overline{\mathsf{f}}$	g $\overline{\mathsf{m}}$ f	g m $\overline{\mathsf{f}}$	g m f
#	34	6	2	8	16	24	0	10

Database D with 100 entries for 3 attributes.

As the structure given by the graph of the previous slide suggests, the probability of $P(g, m, f)$ can be computed by:

$$P(\mathsf{g}, \mathsf{m}, \mathsf{f}) = P(\mathsf{g})P(\mathsf{m})P(\mathsf{f} \mid \mathsf{g}, \mathsf{m})$$

Estimates for these probabilities can be calculated, e.g. using the database

$$\hat{P}(\mathsf{f} \mid \mathsf{g}, \mathsf{m}) = \frac{\hat{P}(\mathsf{f}, \mathsf{g}, \mathsf{m})}{\hat{P}(\mathsf{g}, \mathsf{m})} = \frac{\frac{\#(\mathsf{g},\mathsf{m},\mathsf{f})}{|D|}}{\frac{\#(\mathsf{g},\mathsf{m})}{|D|}} = \frac{\#(\mathsf{g}, \mathsf{m}, \mathsf{f})}{\#(\mathsf{g}, \mathsf{m})} = \frac{10}{10} = 1.00$$

$$\hat{P}(\mathsf{f} \mid \overline{\mathsf{g}}, \overline{\mathsf{m}}) = \frac{\hat{P}(\mathsf{f}, \overline{\mathsf{g}}, \overline{\mathsf{m}})}{\hat{P}(\overline{\mathsf{g}}, \overline{\mathsf{m}})} = \frac{\frac{\#(\overline{\mathsf{g}},\overline{\mathsf{m}},\mathsf{f})}{|D|}}{\frac{\#(\overline{\mathsf{g}},\overline{\mathsf{m}})}{|D|}} = \frac{\#(\overline{\mathsf{g}}, \overline{\mathsf{m}}, \mathsf{f})}{\#(\overline{\mathsf{g}}, \overline{\mathsf{m}})} = \frac{6}{40} = 0.15$$

Likelihood of a Database

Let B_P be the description of the parameters, B_S be the given structure and D the data.
The likelihood of the calculated probabilities $P(D \mid B_S, B_P)$ can be computed under presence of three assumptions:

1. The data generation process can be described exactly by a Bayesian network (B_S, B_P)
2. The single tuples of the dataset are independent of each other.
3. All tuples are complete, therefore no missing values hinder the probability inference

The first assumption legitimates the search of an appropriate bayesian network.

The second assumption is required for an unbiased observation of dataset tuples.

Assumption three ensures the inference of B_P using D and B_S as shown on the previous slides.

Likelihood of a Database

Flu G	$\overline{\mathsf{g}}$	$\overline{\mathsf{g}}$	$\overline{\mathsf{g}}$	$\overline{\mathsf{g}}$	g	g	g	g
Malaria M	$\overline{\mathsf{m}}$	$\overline{\mathsf{m}}$	m	m	$\overline{\mathsf{m}}$	$\overline{\mathsf{m}}$	m	m
Fever F	$\overline{\mathsf{f}}$	f	$\overline{\mathsf{f}}$	f	$\overline{\mathsf{f}}$	f	$\overline{\mathsf{f}}$	f
#	34	6	2	8	16	24	0	10

Database D with 100 entries for 3 attributes.

$$P(D \mid B_S, B_P) = \prod_{h=1}^{100} P(c_h \mid B_S, B_P)$$

$$= \underbrace{\overbrace{P(\mathsf{g},\mathsf{m},\mathsf{f})}^{\text{Case 1}} \cdot \ldots \cdot \overbrace{P(\mathsf{g},\mathsf{m},\mathsf{f})}^{\text{Case 10}}}_{10\text{ times}} \quad \cdots \quad \underbrace{\overbrace{P(\overline{\mathsf{g}},\mathsf{m},\mathsf{f})}^{\text{Case 51}} \cdot \ldots \cdot \overbrace{P(\overline{\mathsf{g}},\mathsf{m},\mathsf{f})}^{\text{Case 58}}}_{8\text{ times}} \quad \cdots \quad \underbrace{\overbrace{P(\overline{\mathsf{g}},\overline{\mathsf{m}},\overline{\mathsf{f}})}^{\text{Case 67}} \cdot \ldots \cdot \overbrace{P(\overline{\mathsf{g}},\overline{\mathsf{m}},\overline{\mathsf{f}})}^{\text{Case 100}}}_{34\text{ times}}$$

$$= \underbrace{P(\mathsf{g},\mathsf{m},\mathsf{f})^{10}} \quad \cdots \quad \underbrace{P(\overline{\mathsf{g}},\mathsf{m},\mathsf{f})^{8}} \quad \cdots \quad \underbrace{P(\overline{\mathsf{g}},\overline{\mathsf{m}},\overline{\mathsf{f}})^{34}}$$

$$= \overbrace{P(\mathsf{f} \mid \mathsf{g},\mathsf{m})^{10} P(\mathsf{g})^{10} P(\mathsf{m})^{10}} \quad \cdots \quad \overbrace{P(\mathsf{f} \mid \overline{\mathsf{g}},\mathsf{m})^{8} P(\overline{\mathsf{g}})^{8} P(\mathsf{m})^{8}} \quad \cdots \quad \overbrace{P(\overline{\mathsf{f}} \mid \overline{\mathsf{g}},\overline{\mathsf{m}})^{34} P(\overline{\mathsf{g}})^{34} P(\overline{\mathsf{m}})^{34}}$$

Likelihood of a Database

$$P(D \mid B_S, B_P) \quad = \quad \prod_{h=1}^{100} P(c_h \mid B_S, B_P)$$

$$= P(\mathsf{f} \mid \mathsf{g}, \mathsf{m})^{10} P(\bar{\mathsf{f}} \mid \mathsf{g}, \mathsf{m})^{0} P(\mathsf{f} \mid \mathsf{g}, \overline{\mathsf{m}})^{24} P(\bar{\mathsf{f}} \mid \mathsf{g}, \overline{\mathsf{m}})^{16}$$

$$\cdot\, P(\mathsf{f} \mid \overline{\mathsf{g}}, \mathsf{m})^{8} P(\bar{\mathsf{f}} \mid \overline{\mathsf{g}}, \mathsf{m})^{2} P(\mathsf{f} \mid \overline{\mathsf{g}}, \overline{\mathsf{m}})^{6} P(\bar{\mathsf{f}} \mid \overline{\mathsf{g}}, \overline{\mathsf{m}})^{34}$$

$$\cdot\, P(\mathsf{g})^{50} P(\overline{\mathsf{g}})^{50} P(\mathsf{m})^{20} P(\overline{\mathsf{m}})^{80}$$

The last equation shows the principle of reordering the factors:

First, we sort by attributes (here: F, G then M).

Within the same attributes, factors are grouped by the parent attributes' values combinations (here: for F: (g, m), $(\mathsf{g}, \overline{\mathsf{m}})$, $(\overline{\mathsf{g}}, \mathsf{m})$ and $(\overline{\mathsf{g}}, \overline{\mathsf{m}})$).

Finally, it is sorted by attribute values (here: for F: first f, then $\bar{\mathsf{f}}$).

Bayes Theorem gives the likelihood $P(B_P \mid D, B_S)$.

Maximum likelihood approach gives a good estimate for $\hat{B}_P$.

Likelihood of a Database

General likelihood of a database D given a known bayesian network structure B_S and the parameters B_P:

$$P(D \mid B_S, B_P) \quad = \quad \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}}$$

General potential table:

A_i	Q_{i1}	$\cdots$	Q_{ij}	$\cdots$	Q_{iq_i}
a_{i1}	θ_{i11}	$\cdots$	θ_{ij1}	$\cdots$	θ_{iq_i1}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{ik}	θ_{i1k}	$\cdots$	θ_{ijk}	$\cdots$	θ_{iq_ik}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{ir_i}	θ_{i1r_i}	$\cdots$	θ_{ijr_i}	$\cdots$	$\theta_{iq_ir_i}$

$$P(A_i = a_{ik} \mid \text{parents}(A_i) = Q_{ij}) = \theta_{ijk}$$

$$\sum_{k=1}^{r_i} \theta_{ijk} = 1$$

Building Bayes Networks:

Structure Learning

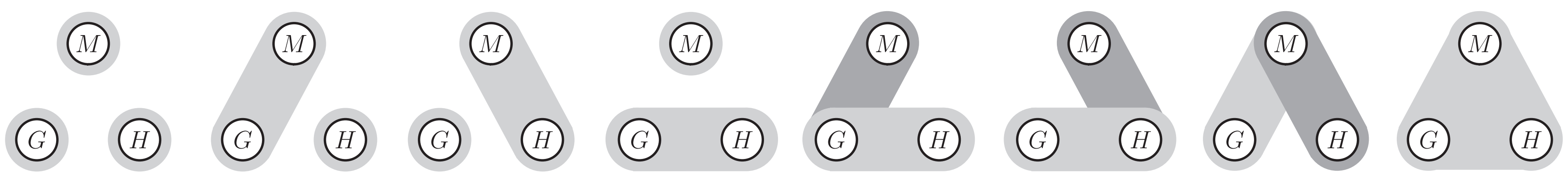
Evaluation Measures and Search Methods

All learning algorithms for graphical models consist of
an **evaluation measure** or **scoring function**
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.



8 possible undirected graphs with 3 nodes

Evaluation Measures / Scoring Functions

Relational Networks

Hartley Information Gain

Conditional Hartley Information Gain

Probabilistic Networks

χ^2-Measure

Mutual Information / Cross Entropy / Information Gain

(Symmetric) Information Gain Ratio

(Symmetric/Modified) Gini Index

Bayesian Measures (K2 metric, BDeu metric)

Measures based on the Minimum Description Length Principle

Other measures that are known from Decision Tree Induction

(A) 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.

(B) 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.

(C) 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.

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 for any family $\mathcal{M}$ of subsets of U it is always:

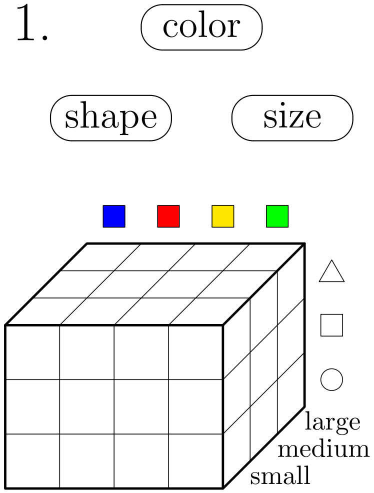
$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$r_U\left(\bigwedge_{A_i \in U} A_i = a_i\right) \leq \min_{M \in \mathcal{M}} \left\{ r_M\left(\bigwedge_{A_i \in M} A_i = a_i\right)\right\}.$$

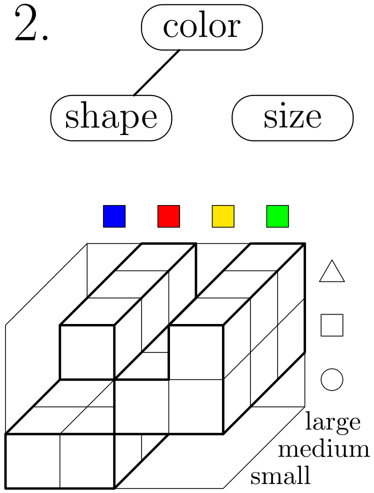
Therefore: Measure the quality of a family $\mathcal{M}$ as:

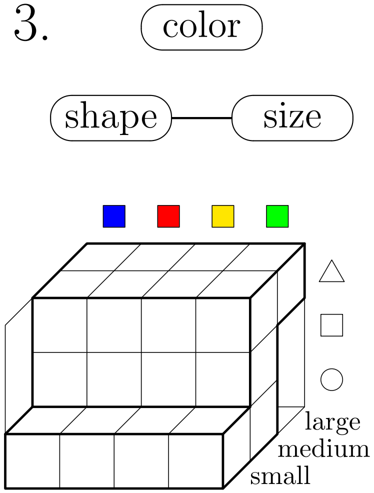
$$\sum_{a_1 \in \mathrm{dom}(A_1)} \cdots \sum_{a_n \in \mathrm{dom}(A_n)} \left(\min_{M \in \mathcal{M}} \left\{ r_M\left(\bigwedge_{A_i \in M} A_i = a_i\right)\right\} - r_U\left(\bigwedge_{A_i \in U} A_i = a_i\right)\right)$$

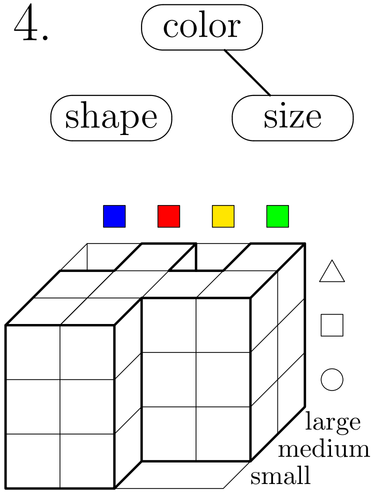
Intuitively: **Count the number of additional tuples.**

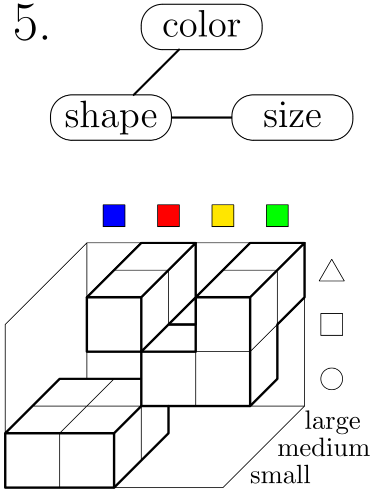
Direct Test for Decomposability: Relational

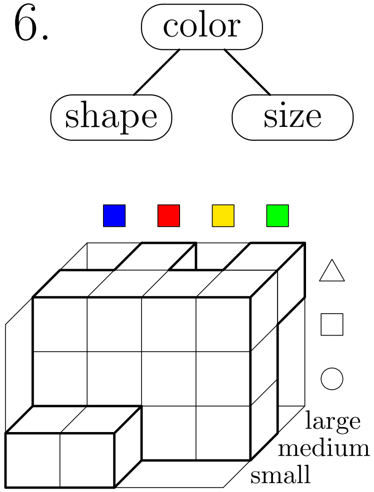
1.


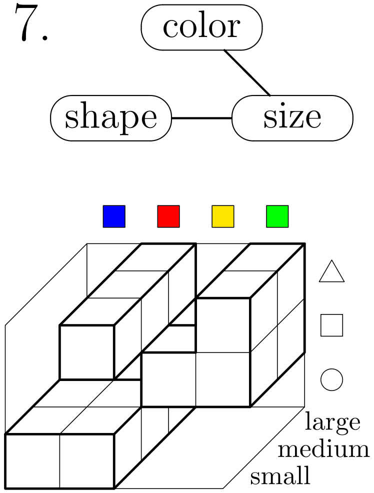
2.


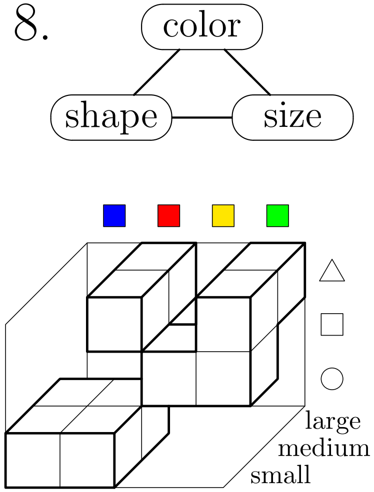
3.


4.


5.


6.


7.


8.


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

$$I_{\mathrm{KLdiv}}(P_1, P_2) = \sum_{F \in \mathcal{E}} P_1(F) \log_2 \frac{P_1(F)}{P_2(F)}$$

is called the **Kullback-Leibler information divergence** of P_1 and P_2.

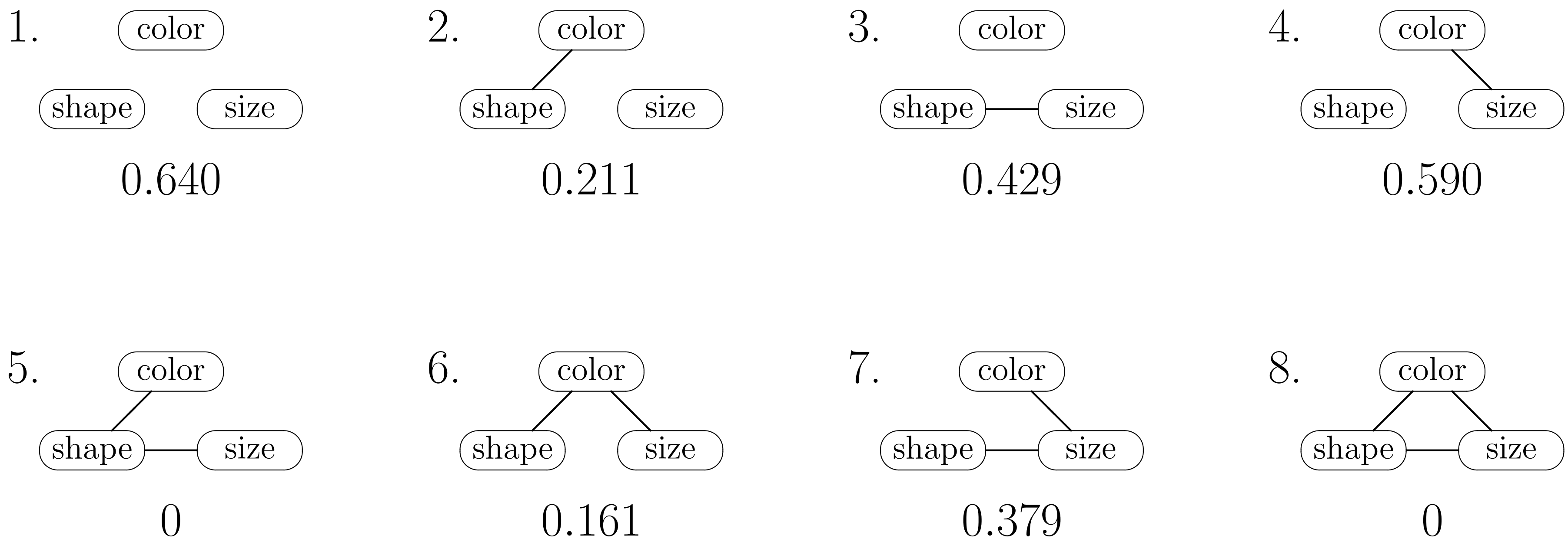
The Kullback-Leibler information divergence is non-negative.

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



Numbers below graphs: The Kullback-Leibler information divergence of the original distribution and its approximation.

Excursus: Shannon Entropy

Let X be a random variable with domain $\mathrm{dom}(X) = \{x_1, \ldots, x_n\}$. Then,

$$H^{(\mathrm{Shannon})}(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

is called the **Shannon entropy** of (the probability distribution of) X, where $0 \cdot \log_2 0 = 0$ is assumed.

Intuitively: **Expected number of yes/no questions that have to be asked in order to determine the obtaining value of X.**

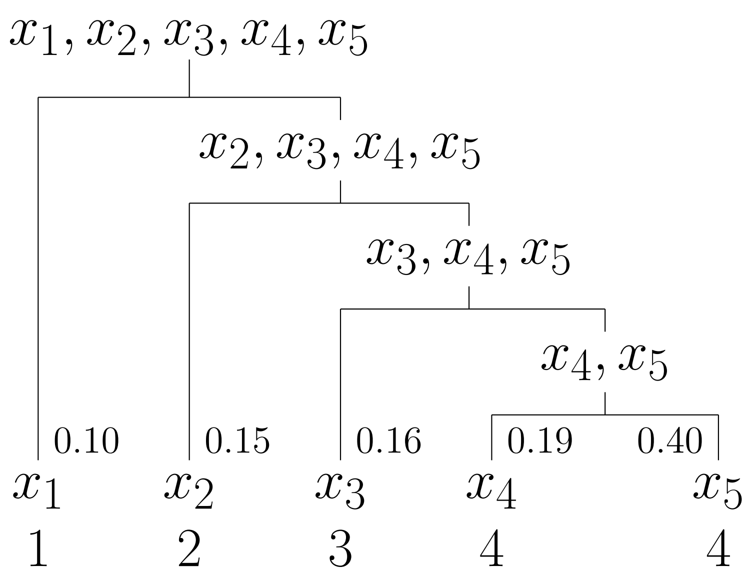
- Suppose there is an oracle, which knows the obtaining value, but responds only if the question can be answered with "yes" or "no".
- A better question scheme than asking for one alternative after the other can easily be found: Divide the set into two subsets of about equal size.
- Ask for containment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

Question/Coding Schemes

$$P(x_1) = 0.10, \quad P(x_2) = 0.15, \quad P(x_3) = 0.16, \quad P(x_4) = 0.19, \quad P(x_5) = 0.40$$

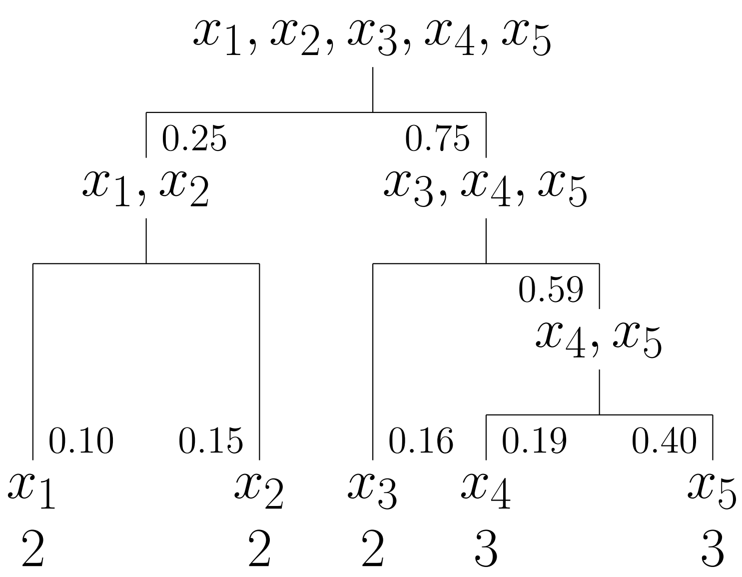
Shannon entropy: $\quad -\sum_i P(x_i) \log_2 P(x_i) = 2.15$ bit/symbol

Linear Traversal



Code length: 3.24 bit/symbol
Code efficiency: 0.664

Equal Size Subsets



Code length: 2.59 bit/symbol
Code efficiency: 0.830

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)

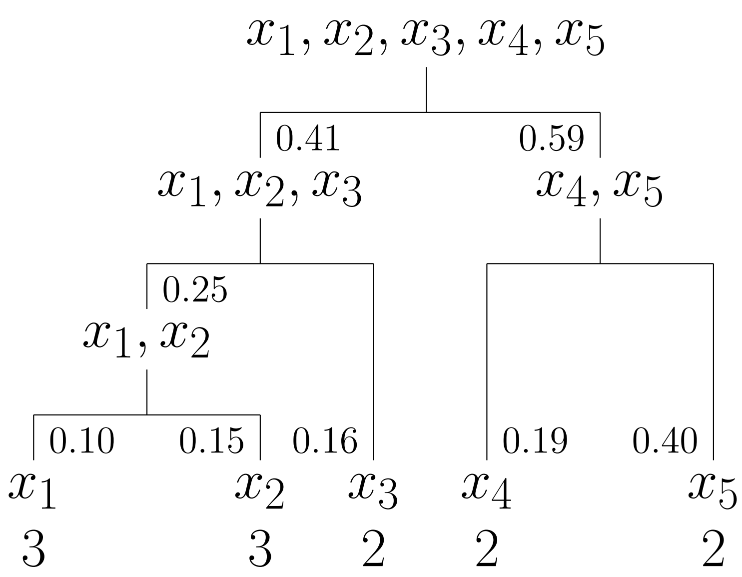
- Build the question/coding scheme bottom-up.
- Start with one element sets.
- Always combine those two sets that have the smallest probabilities.

Question/Coding Schemes

$$P(x_1) = 0.10, \quad P(x_2) = 0.15, \quad P(x_3) = 0.16, \quad P(x_4) = 0.19, \quad P(x_5) = 0.40$$

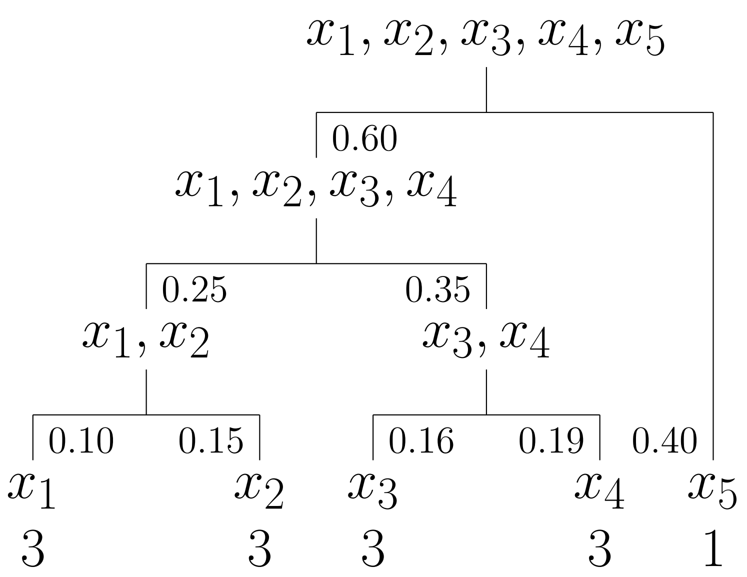
Shannon entropy: $\quad -\sum_i P(x_i) \log_2 P(x_i) = 2.15$ bit/symbol

Shannon–Fano Coding (1948)



Code length: 2.25 bit/symbol
Code efficiency: 0.955

Huffman Coding (1952)



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

$$P(x_1) = \tfrac{1}{2}, \quad P(x_2) = \tfrac{1}{4}, \quad P(x_3) = \tfrac{1}{8}, \quad P(x_4) = \tfrac{1}{16}, \quad P(x_5) = \tfrac{1}{16}$$

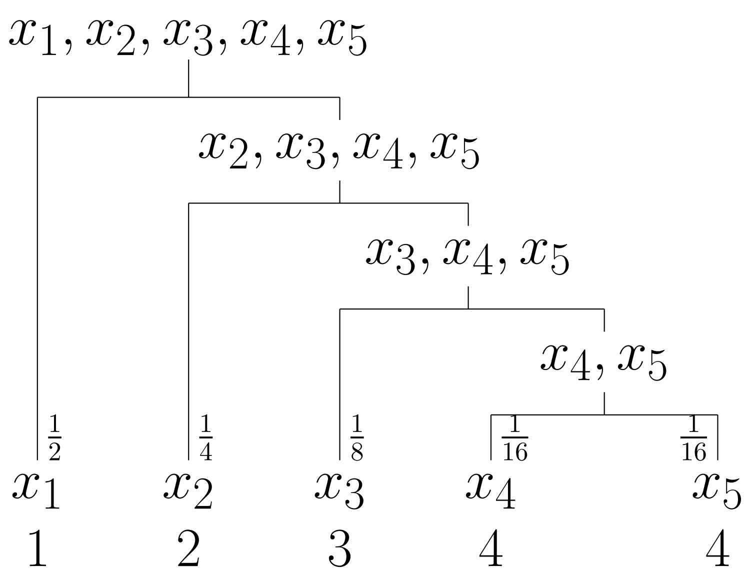
Shannon entropy: $\quad -\sum_i P(x_i) \log_2 P(x_i) = 1.875$ bit/symbol

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

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

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

Reference to Kullback-Leibler Information Divergence

Information Content

The information content of an event $F \in \mathcal{E}$ that occurs with probability $P(F)$ is defined as

$$\mathrm{Inf}_P(F) = -\log_2 P(F).$$

Intention:

Neglect all subjective references to F and let the information content be determined by $P(F)$ only.

The information of a certain message $(P(\Omega) = 1)$ is zero.

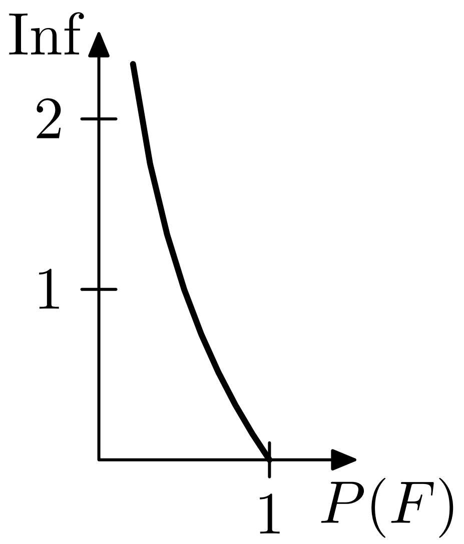
The less frequent a message occurs (i. e., the less probable it is), the more interesting is the fact of its occurrence:

$$P(F_1) < P(F_2) \quad \Rightarrow \quad \mathrm{Inf}_P(F_1) > \mathrm{Inf}_P(F_2)$$

We only use one bit to encode the occurrence of a message with probability $\frac{1}{2}$.

Excursus: Information Content

The function Inf fulfills all these requirements:



The expected value (w. r. t. to a probability distribution P_1) of Inf_{P_2} can be written as follows:

$$E_{P_1}(\mathrm{Inf}_{P_2}) = -\sum_{F \in \mathcal{E}} P_1(F) \cdot \log_2 P_2(F)$$

$H^{(\mathrm{Shannon})}(P)$ is the expected value (in bits) of the information content that is related to the occurrence of the events $F \in \mathcal{E}$:

$$H(P) = E_P(\mathrm{Inf}_P)$$

$$H^{(\mathrm{Shannon})}(P) = \sum_{F \in \mathcal{E}} \underbrace{P(F)}_{\text{Probability of } F} \cdot \underbrace{(-\log_2 P(F))}_{\text{Information content of } F}$$

Excursus: Approximation Measure

Let P^* be a hypothetical probability distribution and P a (given or known) probability distribution that acts as a reference.

We can compare both P^* and P by computing the **difference of the expected information contents**:

$$
\begin{aligned}
E_P(\mathrm{Inf}_{P^*}) - E_P(\mathrm{Inf}_P) &= -\sum_{F \in \mathcal{E}} P(F) \log_2 P^*(F) + \sum_{F \in \mathcal{E}} P(F) \log_2 P(F) \\
&= \sum_{F \in \mathcal{E}} \Big(P(F) \log_2 P(F) - P(F) \log_2 P^*(F) \Big) \\
&= \sum_{F \in \mathcal{E}} P(F) \Big(\log_2 P(F) - \log_2 P^*(F) \Big) \\
I_{\mathrm{KLdiv}}(P, P^*) &= \sum_{F \in \mathcal{E}} P(F) \log_2 \frac{P(F)}{P^*(F)}
\end{aligned}
$$

(A) 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.

(B) 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.

(C) 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.

Strength of Marginal Dependences: Relational

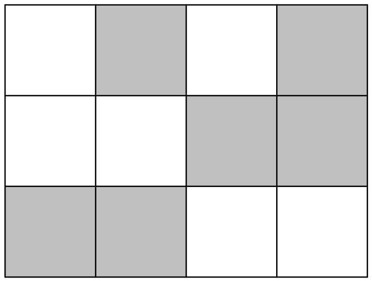
Learning a relational network consists in finding those subspace, for which the intersection of the cylindrical extensions of the projections to these subspaces approximates best the set of possible world states, i. e. contains as few additional tuples as possible.

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 × shape	shape × size	size × 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:	$\log_2 4 + \log_2 3 = \log_2 12$	≈ 3.58
coordinate pair:	$\log_2 6$	≈ 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 $\exists a \in \mathrm{dom}(A) : \exists b \in \mathrm{dom}(B) : R(A = a, B = b) = 1$. Then

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

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

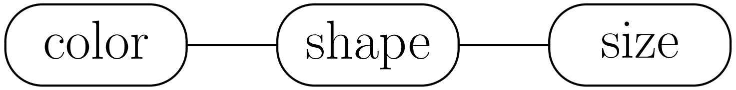
Strength of Marginal Dependences: Simple Example

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:

$$\begin{aligned}
I_{\text{gain}}^{(\text{Hartley})}(\text{color}, \text{shape}) &= 1.00 \text{ bit} \\
I_{\text{gain}}^{(\text{Hartley})}(\text{shape}, \text{size}) &\approx 0.86 \text{ bit} \\
I_{\text{gain}}^{(\text{Hartley})}(\text{color}, \text{size}) &\approx 0.58 \text{ bit}
\end{aligned}$$

Applying the Kruskal algorithm yields as a learning result:



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

Mutual Information / Cross Entropy / Information Gain

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

$$
\begin{aligned}
I_{\text{gain}}(A, B) &= \overbrace{-\sum_a P(a) \log_2 P(a)}^{H(A)} - \overbrace{\sum_b P(b) \left(-\sum_a P(a|b) \log_2 P(a|b) \right)}^{H(A \mid B)}
\end{aligned}
$$

$H(A)$	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*

Strength of Marginal Dependences: Probabilistic

$$
\begin{aligned}
I_{\text{gain}}(A,B) &= -\sum_a P(a)\log_2 P(a) - \sum_b P(b)\left(-\sum_a P(a|b)\log_2 P(a|b)\right) \\
&= -\sum_a \sum_b P(a,b)\log_2 P(a) + \sum_b \sum_a P(a|b)P(b)\log_2 P(a|b) \\
&= \quad \sum_a \sum_b P(a,b)\left(\log_2 \frac{P(a,b)}{P(b)} - \log_2 P(a)\right) \\
&= \quad \sum_a \sum_b P(a,b)\log_2 \frac{P(a,b)}{P(a)P(b)}
\end{aligned}
$$

The information gain equals the Kullback-Leibler information divergence between the actual distribution $P(A,B)$ and a hypothetical distribution P^* in which A and B are marginal independent:

$$P^*(A,B) = P(A) \cdot P(B)$$

$$I_{\text{gain}}(A,B) = I_{\text{KLdiv}}(P,P^*)$$

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

projection to subspace

	s	m	l
△	20	180	200
□	40	160	40
○	180	120	60

product of marginals

	s	m	l
△	96	184	120
□	58	110	72
○	86	166	108

information gain: 0.211 bit

projection to subspace

	🟦	🟥	🟨	🟩
large	50	115	35	100
medium	82	133	99	146
small	88	82	36	34

product of marginals

	🟦	🟥	🟨	🟩
large	66	99	51	84
medium	101	152	78	129
small	53	79	41	67

information gain: 0.050 bit

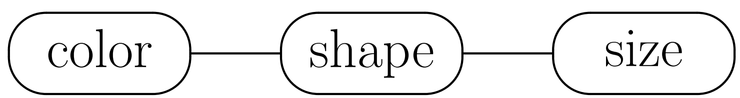
Strength of Marginal Dependences: Simple Example

Results for the simple example:

$$\begin{aligned} I_{\text{gain}}(\text{color}, \text{shape}) &= 0.429 \text{ bit} \\ I_{\text{gain}}(\text{shape}, \text{size}) &= 0.211 \text{ bit} \\ I_{\text{gain}}(\text{color}, \text{size}) &= 0.050 \text{ bit} \end{aligned}$$

The Kruskal algorithm is an greedy algorithm which can be used to determine the minimal spanning tree of an undirected graph.

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).

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.

K2 Algorithm

Idea: Compute the probability of a directed graph B_S given the database D (Bayesian approach by [Cooper and Herskovits 1992])

$$\begin{aligned}\hat{B}_S &= \arg\max_{B_S} P(B_S \mid D) = \arg\max_{B_S} \frac{P(B_S, D)}{P(D)} \\ &= \arg\max_{B_S} P(B_S, D)\end{aligned}$$

Find an equation for $P(B_S, D)$.

Model Averaging

We first consider $P(B_S, D)$ to be the marginalization of $P(B_S, B_P, D)$ over all possible parameters B_P.

$$
\begin{aligned}
P(B_S, D) &= \int_{B_P} P(B_S, B_P, D)\,\mathrm{d}B_P \\
&= \int_{B_P} P(D \mid B_S, B_P)\,P(B_S, B_P)\,\mathrm{d}B_P \\
&= \int_{B_P} P(D \mid B_S, B_P)\,f(B_P \mid B_S)P(B_S)\,\mathrm{d}B_P \\
&= \underbrace{P(B_S)}_{\text{A priori prob.}} \int_{B_P} \underbrace{P(D \mid B_S, B_P)}_{\text{Likelihood of } D}\ \underbrace{f(B_P \mid B_S)}_{\text{Parameter densities}}\ \mathrm{d}B_P
\end{aligned}
$$

K2 Algorithm

The a priori distribution $P(B_S)$ can be used to bias the evaluation measure towards user-specific network structures.

Substitute the likelihood $P(D \mid B_S, B_P)$ for its specific form:

$$P(B_S, D) \;=\; P(B_S) \int_{B_P} \underbrace{\left[\prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \right]}_{P(D|B_S,B_P)} f(B_P \mid B_S) \, \mathrm{d}B_P$$

See slide 355 for the derivation of the likelihood term.

K2 Algorithm

The parameter densities $f(B_P \mid B_S)$ describe the probabilities of the parameters given a network structure.

They are densities of second order (distribution over distributions)

For fixed i and j, a vector $(\theta_{ij1}, \ldots, \theta_{ijr_i})$ represents a probability distribution, namely the j-th column of the i-th potential table.

Assuming mutual independence between the potential tables, we arrive for $f(B_P \mid B_S)$ at the following:

$$f(B_P \mid B_S) \quad = \quad \prod_{i=1}^{n} \prod_{j=1}^{q_i} f(\theta_{ij1}, \ldots, \theta_{ijr_i})$$

K2 Algorithm

Thus, we can further concretize the equation for $P(B_S, D)$:

$$
\begin{aligned}
P(B_S, D) &= P(B_S) \underset{\theta_{ijk}}{\int \cdots \int} \left[\prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \right] \cdot \left[\prod_{i=1}^{n} \prod_{j=1}^{q_i} f(\theta_{ij1}, \ldots, \theta_{ijr_i}) \right] \mathrm{d}\theta_{111}, \ldots, \mathrm{d}\theta_{nq_nr_n} \\
&= P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \underset{\theta_{ijk}}{\int \cdots \int} \left[\prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \right] \cdot f(\theta_{ij1}, \ldots, \theta_{ijr_i}) \, \mathrm{d}\theta_{ij1}, \ldots, \mathrm{d}\theta_{ijr_i}
\end{aligned}
$$

A last assumption: For fixed i and j the density $f(\theta_{ij1}, \ldots, \theta_{ijr_i})$ is uniform:

$$f(\theta_{ij1}, \ldots, \theta_{ijr_i}) = (r_i - 1)!$$

It simplifies $P(B_S, D)$ further:

$$\begin{aligned} P(B_S, D) &= P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \int \cdots \int_{\theta_{ijk}} \left[\prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \right] \cdot (r_i - 1)! \, \mathrm{d}\theta_{ij1}, \ldots, \mathrm{d}\theta_{ijr_i} \\ &= P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} (r_i - 1)! \qquad \underbrace{\int \cdots \int_{\theta_{ijk}} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}} \, \mathrm{d}\theta_{ij1}, \ldots, \mathrm{d}\theta_{ijr_i}}_{\text{Dirichlet's integral} = \frac{\prod_{k=1}^{r_i} \alpha_{ijk}!}{(\sum_{k=1}^{r_i} \alpha_{ijk} + r_i - 1)!}} \end{aligned}$$

K2 Algorithm

We finally arrive at an expression for $P(B_S, D)$:

$$P(B_S, D) \quad = \quad \mathrm{K2}(B_S \mid D) \quad = \quad P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left[\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \right]$$

n number of attributes describing the domain under consideration

r_i number of values of the i-th attribute A_i, i. e., $r_i = |\mathrm{dom}(A_i)|$

q_i number of instantiations of the parents of the i-th attribute in $\vec{G}$, i. e., $q_i = \prod_{A_j \in \mathrm{parents}(A_i)} r_i = \prod_{A_j \in \mathrm{parents}(A_i)} |\mathrm{dom}(A_i)|$

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

$N_{ij} \quad = \sum_{k=1}^{r_i} \alpha_{ijk}$

Properties of the K2 Metric

Global — Refers to the outer product: The total value of the K2 metric is the product over all K2 values of attribute families.

Local — The likelihood equation assumes that given a parents instantiation, the probabilities for the respective child attribute values are mutual independent. This is reflected in the product over all q_i different parent attributes' value combinations of attribute A_i.

We exploit the global property to write the K2 metric as follows:

$$\mathrm{K2}(B_S \mid D) \quad = \quad P(B_S) \prod_{i=1}^{n} \mathrm{K2}_{\mathrm{local}}(A_i \mid D)$$

with

$$\mathrm{K2}_{\mathrm{local}}(A_i \mid D) \quad = \quad \prod_{j=1}^{q_i} \left[\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \right]$$

K2 Algorithm

Prerequisites:

Choose a topological order on the attributes $(A_1, \ldots, A_n)$

Start out with a network that consists of n isolated nodes.

Let ζ_i be the quality of the i-th attribute given the (tentative) set of parent attributes M:

$$\zeta_i(M) = \mathrm{K2}_{\mathrm{local}}(A_i \mid D) \quad \text{with} \quad \mathrm{parents}(A_i) = M$$

K2 Algorithm

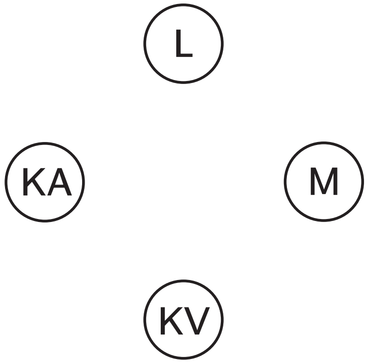
Execution:

1. Determine for the parentless node A_i the quality measure $\zeta_i(\emptyset)$
2. Evaluate for every predecessor $\{A_1, \ldots, A_{i-1}\}$ whether inserted as parent of A_i, the quality measure would increase. Let Y be the node that yields the highest quality (increase):
$$Y = \operatorname*{arg\,max}_{1 \leq l \leq i-1} \zeta_i(\{A_l\})$$
This best quality measure be $\zeta = \zeta_i(\{Y\})$.
3. If ζ is better than $\zeta_i(\emptyset)$, Y is inserted permanently as a parent node: $\text{parents}(A_i) = \text{parents}(A_i) \cup \{Y\}$
4. Repeat steps 2 and 3 to increase the parent set until no quality increase can be achieved or no nodes are left or a predefined maximum number of parent nodes per node is reached.

K2 Algorithm

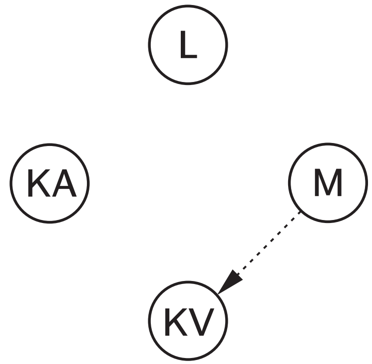
for $i \leftarrow 1 \ldots n$ **do** // Initialization
  parents$(A_i) \leftarrow \emptyset$
end for
for $i \leftarrow n, \ldots, 1$ **do** // Iteration
  repeat
    Select $Y \in \{A_1, \ldots, A_{i-1}\} \setminus$ parents(A_i),
      which maximizes $\zeta = \zeta_i(\text{parents}(A_i) \cup \{Y\})$
    $\delta \leftarrow \zeta - \zeta_i(\text{parents}(A_i))$
    if $\delta > 0$ **then**
      parents$(A_i) \leftarrow$ parents$(A_i) \cup \{Y\}$
    end if
  until $\delta \leq 0$ or parents$(A_i) = \{A_1, \ldots, A_{i-1}\}$ or $|\text{parents}(A_i)| = n_{\max}$
end for

Demo of K2 Algorithm



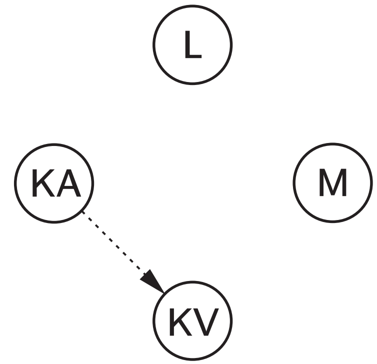
-43942.99

Step 1 – Edgeless graph



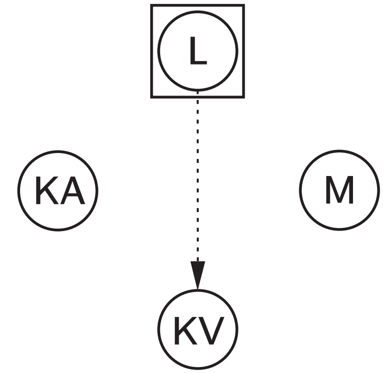
-43950.53

Step 2 – Insert M temporarily.



-43949.30

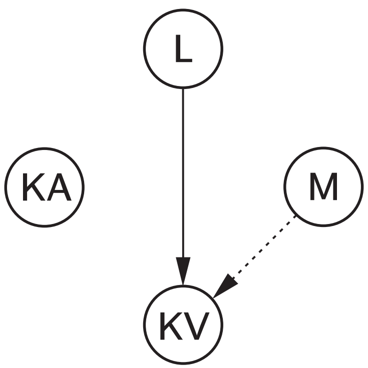
Step 3 – Insert KA temporarily.



-43942.48

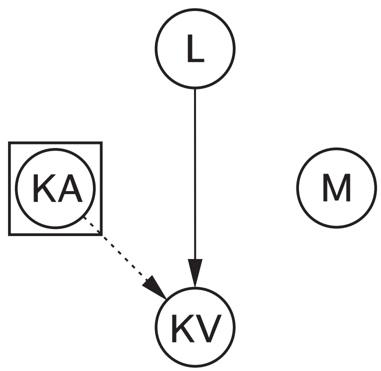
Step 4 – Node L maximizes K2 value and thus is added permantently.

Demo of K2 Algorithm



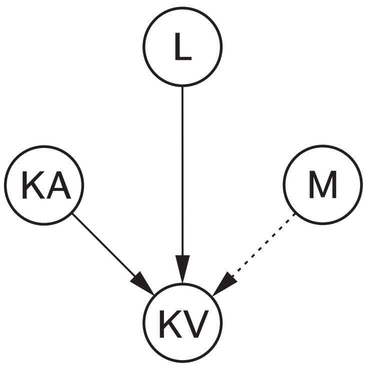
−43964.12

Step 5 – Insert M temporarily.



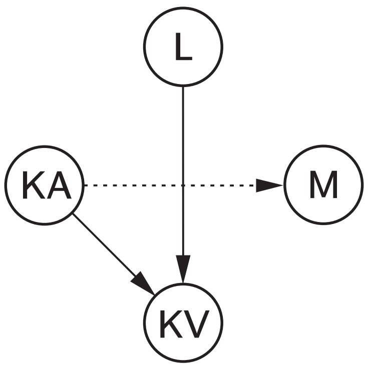
−43933.71

Step 6 – KA is added as second parent node of KV.



−43995.02

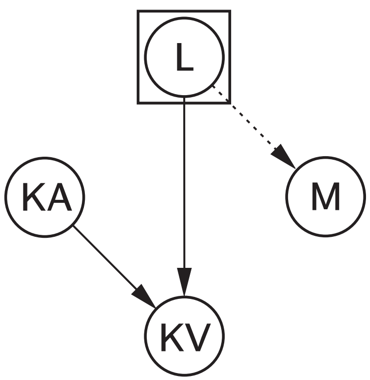
Step 7 – M does not increase the quality of the network if insertes as third parent node.



−43492.08

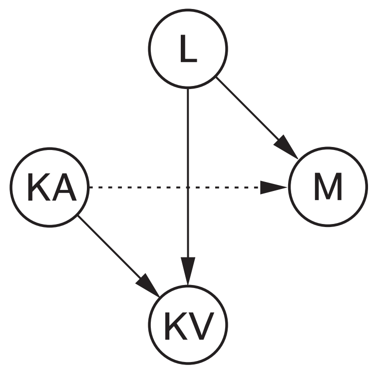
Step 8 – Insert KA temporarily.

Demo of K2 Algorithm



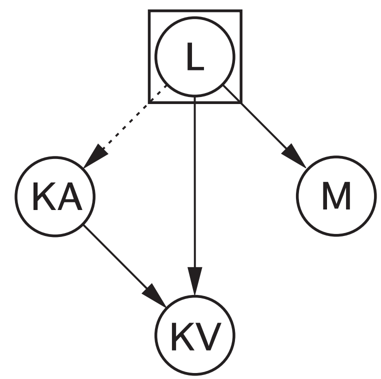
-40823.76

Step 9 – Node L becomes perent node of M.



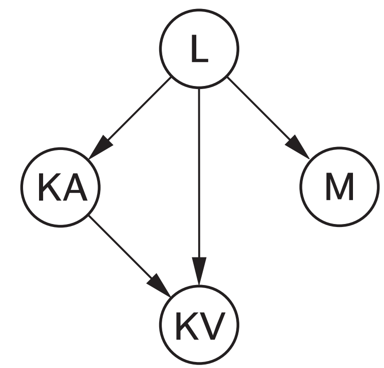
-40903.39

Step 10 – Adding KA does not increase overall network quaility.



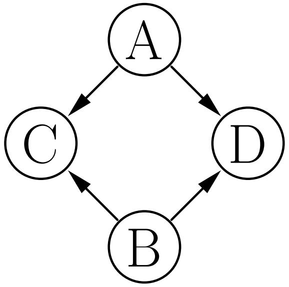
-39190.67

Step 11 – Node L becomes parent node of KA.



Result

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.

(A) 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.

(B) 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.

(C) 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.

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.

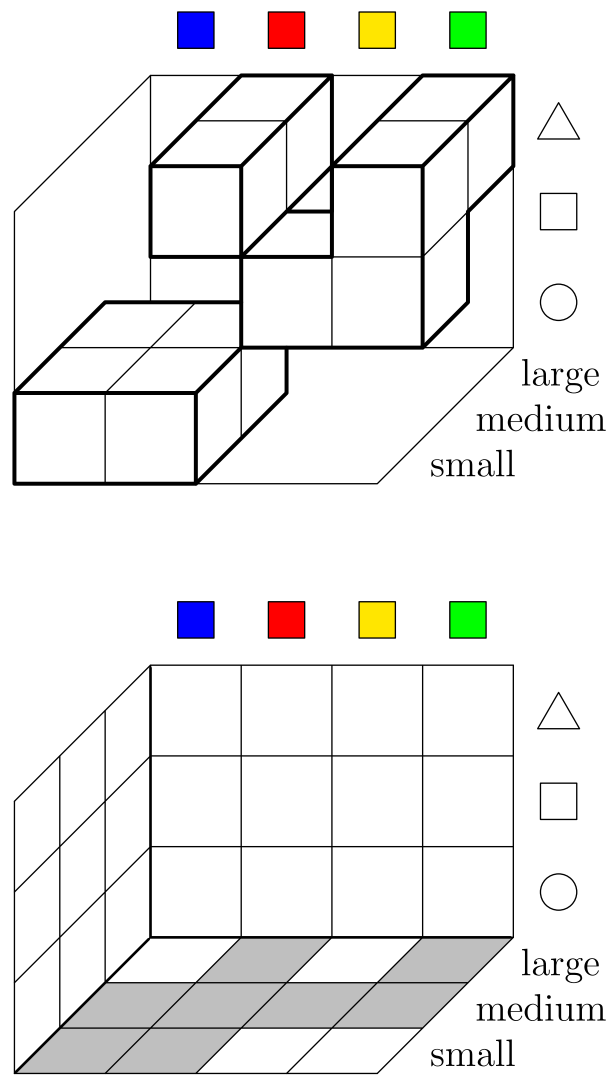
Assumptions:

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

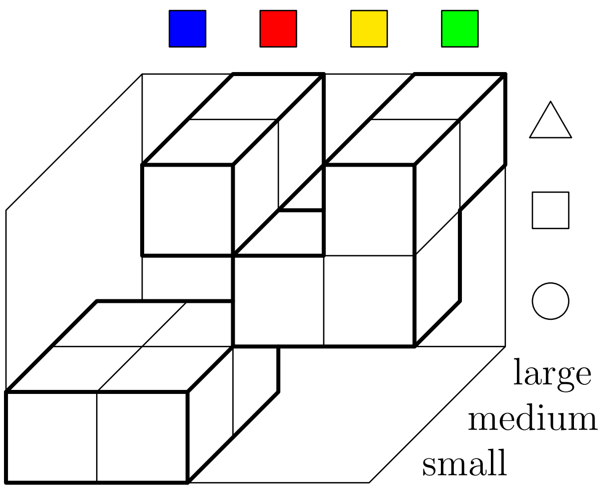
The Hartley information gain can be used directly to test for (approximate) **marginal independence**.

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
	$\log_2 1 + \log_2 2 - \log_2 2 = 0$
	$\log_2 2 + \log_2 3 - \log_2 4 \approx 0.58$
	$\log_2 1 + \log_2 1 - \log_2 1 = 0$
	$\log_2 2 + \log_2 2 - \log_2 2 = 1$
	average: ≈ 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: ≈ 0.33

Conditional Independence Tests: Simple Example

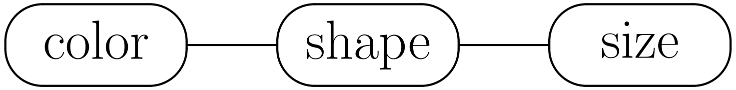
The Shannon information gain can be used directly to test for (approximate) **marginal independence**.

Conditional independence tests may be carried out by summing the information gain for all instantiations of the conditioning variables:

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

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 $S_{AB} \subseteq U \backslash \{A, B\}$ such that $A \perp\!\!\!\perp B \mid S_{AB}$ holds in $\widehat{P}$, i.e., A and B are independent in $\widehat{P}$ 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:

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

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.

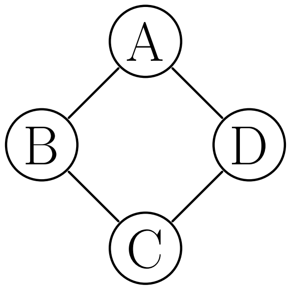
Step 1: Since there is no set rendering A and C, B and C and C and D independent, the edges $A - C$, $B - C$, and $C - D$ are inserted.

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.

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 **)

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

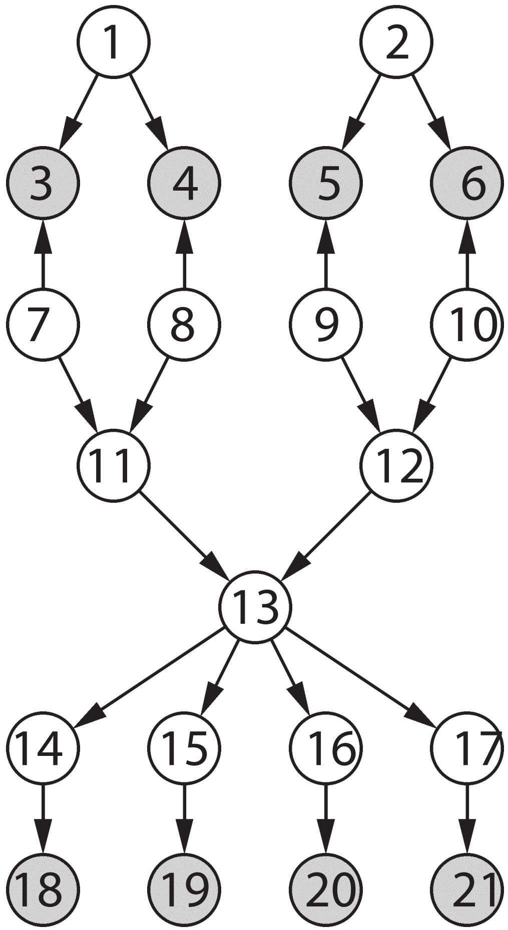
Optimum Weight Spanning Tree Construction

measure	edges	params.	train	test
I_{gain}	20.0	285.9	-12122.6	-12339.6
χ^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_{gain}	35.0	17.1	4.1	1342.2	-11229.3	-11817.6
χ^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
$L_{red}^{(rel)}$	22.5	0.6	0.1	219.9	-11389.5	-11508.2

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 – truedamph.gr. 1	17 – factor 43
8 – truedamph.gr. 2	18 – lysis40
9 – true sire ph.gr. 1	19 – lysis41
10 – true sire ph.gr. 2	20 – lysis 42
	21 – lysis 43

The grey nodes correspond to observable attributes.

Fields of Application (DaimlerChrysler AG)

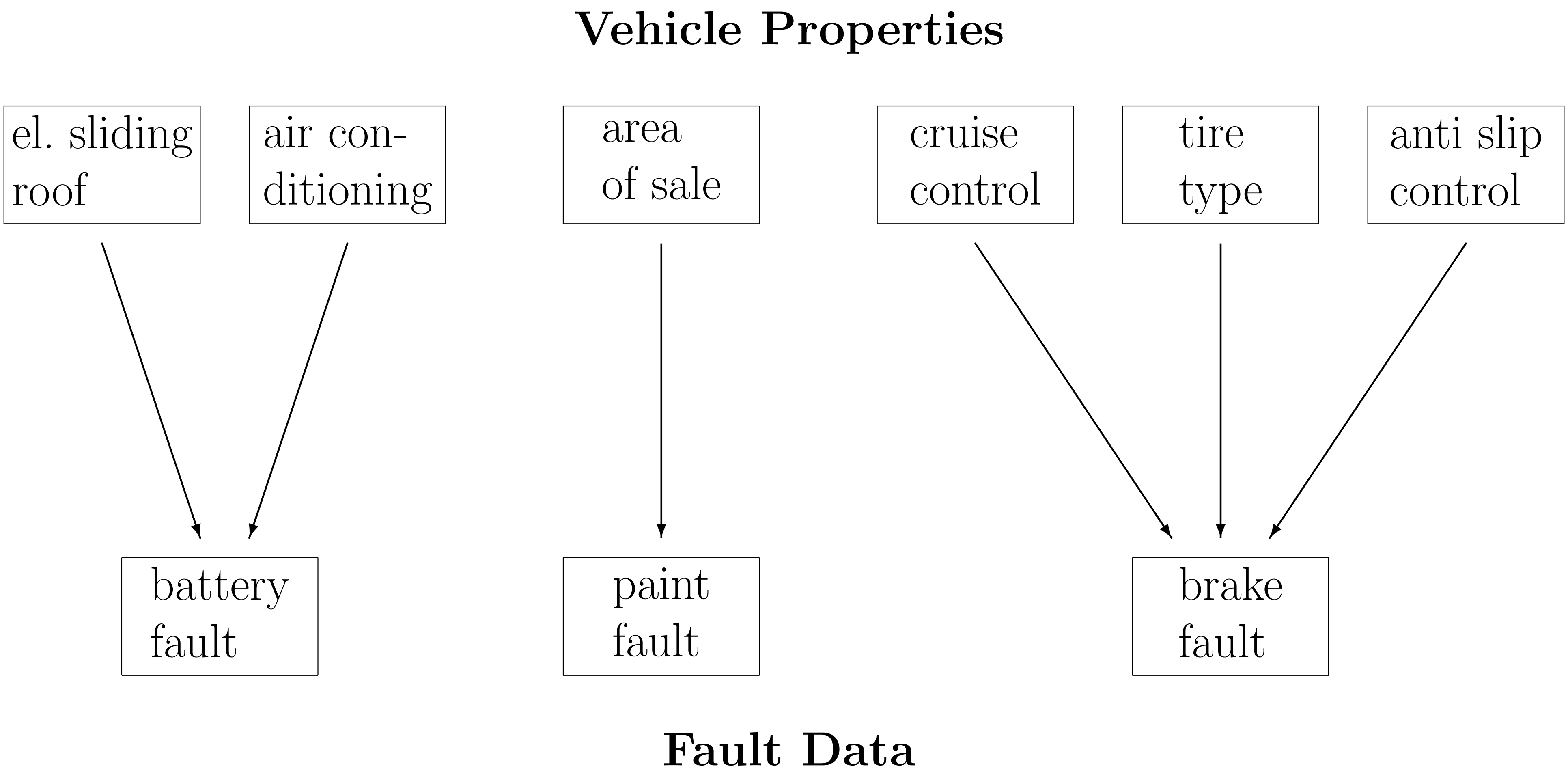
Improving the Product Quality by Detecting Weaknesses

- Learn decision trees or inference network
  for vehicle properties and failures.
- Look for suspicious conditional failure rates.
- Find causes of these suspicious rates.
- Optimize design of vehicle.

Improve the Error Diagnosis in Service Garages

- Learn a decision tree or inference network
  for vehicle properties and failures.
- Record new faults.
- Test for most probable errors.

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



Example Subnet

Influence of special equipment on battery faults:

(fictitious) frequency of battery faults		air conditioning with	air conditioning 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.)

Revision of Probabilistic Graphical Models

Revision of Probabilistic Graphical Models

Graphical models are efficient for representing domain knowledge. After some time additional observations can change our underlying knowledge of the domain.

We need a way to incorporate those changes to avoid updating the whole knowledge base.

Idea: local changes should only lead to local adaptations of the knowledge base and neccessary consequences.

Example: It is known that a certain navigation system can only be included in the car if one of the corresponding radio systems is already installed. It is planned to sell 3000 instead of 1000 navigation systems in the next quartal. How many radio systems of each type should be bought.

Revision of Probabilistic Graphical Models

Prior Probability Distribution

New dimensional conditional probabilities

Revision

Principle of minimal Change

Posterior Probability Distribution including specified **and** inferred changes

Information-theoretically closest to the prior distribution

Revision Operator - Iterative Proportional Fitting

Iterative proportional fitting is a well-known algorithm for adapting the marginal distributions of a given joint distribution to desired values.

It consists in computing the following sequence of probability distributions:

$$p_U^{(0)}(u) \equiv p_U(u) \tag{1}$$

$$\forall 1, 2, \cdots : p_U^{(i)}(u) \equiv p_U^{(i-1)}(u) \frac{p_{A_j}(a)}{p_{A_j}^{(i-1)}(a)} \tag{2}$$

In each step the probability distribution is modified in such a way that the resulting distribution satisfies the given marginal distribution A_j. However, this will, in general, change the marginal distribution for an earlier adapted variable A_k.

Therefore, the adaptation has to be iterated, traversing the set of variables several times. The process is proofed to converge for non-contradicting revision statements.

Revision Algorithm

The revision algorithm sums up as follows:

forall $C \in \mathcal{C}$ **do**
    $p_C^{(0)}(c) \equiv p_C(c)$
$i \equiv 0$
repeat
$i \equiv i + 1;$
    forall $C \in \mathcal{C}$ **do**
        forall $j \in J_C$ **do**
            $p_C^{(i)}(c) \equiv p_C^{(i-1)}(c)\dfrac{p_{A_j}(a)}{p_{A_j}^{(i-1)}(a)};$
        do evidence propagation
    end
until convergence

Inconsistencies

Inconsistencies can emerge in the presence of:

Complex structure: dependencies between attributes
e.g. dependencies between car components

Many revision assignments: changes of the probability distribution
e.g. changing installation rate of component combinations

Inconsistencies are unincorporatable changes / inconsistent revision assignments

Inner Inconsistencies

Revision assignments are inconsistent independent of prior distribution

Outer Inconsistencies

Revision assignments inconsistent with zero-values in prior distribution

Example: Inner Inconsistencies

Insert revision assignments in probability distribution:

0.6	0.3		
0.2	0.25		
0.3			0.5
		0.1	

Inner inconsistencies can emerge as consequences of probability implications:

0.6	0.3	0.1		
0.2	0.25			
0.3			0.5	
0.1		0.1		

$\leftarrow 0.10 = 1 - 0.6 - 0.3$ (for the 0.1 in the top row)

$\leftarrow 0.10 = 0.6 - 0.2 - 0.3$ (for the 0.1 in the bottom row)

Example: Inner Inconsistencies

Insert revision assignments in probability distribution:

0.6	0.3		
0.2	0.25		
0.3			0.5
		0.1	

Inner inconsistencies can emerge as consequences of probability implications:

0.6	0.3	0.1		
0.2	0.25	0.0		← set to zero since column sum is already maximum
0.3		0.0	0.5	← set to zero since column sum is already maximum
0.1		0.1		

Example: Inner Inconsistencies

Insert revision assignments in probability distribution:

0.6	0.3		
0.2	0.25		
0.3			0.5
		0.1	

Inner inconsistencies can emerge as consequences of probability implications:

0.6	0.3	0.1		
0.2	0.25	0.0	0.45	$\leftarrow 0.45 = 0.2 + 0.25 + 0.0$
0.3		0.0	0.5	
0.1		0.1	0.05	

Example: Inner Inconsistencies

Insert revision assignments in probability distribution:

0.6	0.3		
0.2	0.25		
0.3			0.5
		0.1	

Inner inconsistencies can emerge as consequences of probability implications:

0.6	0.3	0.1	
0.2	0.25	0.0	0.45
0.3		0.0	0.5
0.1		0.1	?

Contradicting implications: $0.05 \neq 0.20$

column-sum $\Rightarrow 1 - 0.45 - 0.5 = 0.05$

row-sum $\Rightarrow 0.1 + 0.1 = 0.20$

Example: Outer Inconsistencies

Insert revision assignments in probability distribution with fixed zero values ($\times$):

0.1	0.5	0.4	
	$\times$	$\times$	0.3
	$\times$		0.5

Outer inconsistencies can emerge as consequences of probability implications:

0.1	0.5	0.4	
	$\times$	$\times$	0.3
	$\times$		0.5
	0.5		0.2

Contradicting implications: $0.5 > 0.2$

column-sum $\Rightarrow 0.5 - 0.0 - 0.0 = 0.5$

column-sum $\Rightarrow 1.0 - 0.3 - 0.5 = 0.2$

Example: Outer Inconsistencies

Even for an expert user it is not easy to configure revision statements without creating inconsistencies!

If the Revision-Operation fails, we need to explain the user how to change his desired revision statements. Otherwise no solution can be found.

Systematic Handling of Inconsistencies

1. Detection

Heuristical Methods

2. Automatic Resolution

Partition Mirrors

3. Analysis

Finding Minimal Explaining Set
Grouping Assignments

4. Explanation

Displaying minimal explaining set and structure

(Handling Revision Inconsistencies:
Creating Useful Explanations, *Schmidt, F.; Gebhardt, J.; Kruse, R.*)

Decision Graphs - Influence Diagrams

Descriptive Decision Theory

Descriptive Decision Theory tries to simulate human behavior in finding the right or best decision for a given problem

Example:

- Company can chose one of two places for a new store
- Option 1: 125.000 EUR profit per year
- Option 2: 150.000 EUR profit per year

Company should take Option 2, because it maximized the profit.

Decisions under Uncertainty

In real world not every thing is known, so there are uncertainties in the model

Example:

- There are plans for restructure the local traffic, which changes the predicted profit
- Option 1: 125.000 EUR profit per year
- Option 2: 80.000 EUR profit per year

  With modification Option 1 is the better one and without modification Option 2 is the better one

To model these variations in the environment we use so called Decision Tables

	z_1 (no modification)	z_2 (restructure)
a_1 (Option 1)	$125.000 = e_{11}$	$125.000 = e_{12}$
a_2 (Option 2)	$150.000 = e_{21}$	$80.000 = e_{22}$

Probability-based Decisions

In many cases probabilities could be assigned to each option

Objective Probabilities based on mathematic or statistic background

Subjective Probabilities based on intuition or estimations

Example:

- The management estimates the probability for the restructure to 30%

  The decision can be chosen by expectation value

	z_1 (no modification) $p_1 = 0.7$	z_2 (restructure) $p_2 = 0.3$	Expectation Value
a_1 (Option 1)	$125.000 = e_{11}$	$125.000 = e_{12}$	125.000
a_2 (Option 2)	$150.000 = e_{21}$	$80.000 = e_{22}$	129.000

Option 2 has the higher expectation value and should be used

Domination

An alternative a_1 dominates a_2 if the value of a_1 is always greater of (or equal to) the value of a_2

$\forall_j e_{1j} \geq e_{2j}$ ´

Example:

	z_1	z_2
a_1	$150.000 = e_{11}$	$90.000 = e_{12}$
a_2	$125.000 = e_{21}$	$80.000 = e_{22}$

Alternative a_2 could be dropped

Domination - Example 2

Some more alternatives:

	z_1	z_2	z_3	z_4	z_5	
a_1	0	20	10	60	25	dominated by a_3
a_2	-20	80	10	10	60	
a_3	20	60	20	60	50	
a_4	55	40	60	10	40	
a_5	50	10	30	5	20	dominated by a_4

- a_3 dominated a_1
- a_4 dominated a_5

  Alternatives a_1 and a_5 could be dropped

Probability Domination

	z_1 $p_1 = 0.3$	z_2 $p_2 = 0.2$	z_3 $p_1 = 0.4$	z_4 $p_2 = 0.1$
a_1	20	40	10	50
a_2	60	30	50	20

Probability Domination means that the cumulated probability for the payout for is always higher

Algorithm:

- Order payout by value in a decreasing order
- Cumulate probabilities

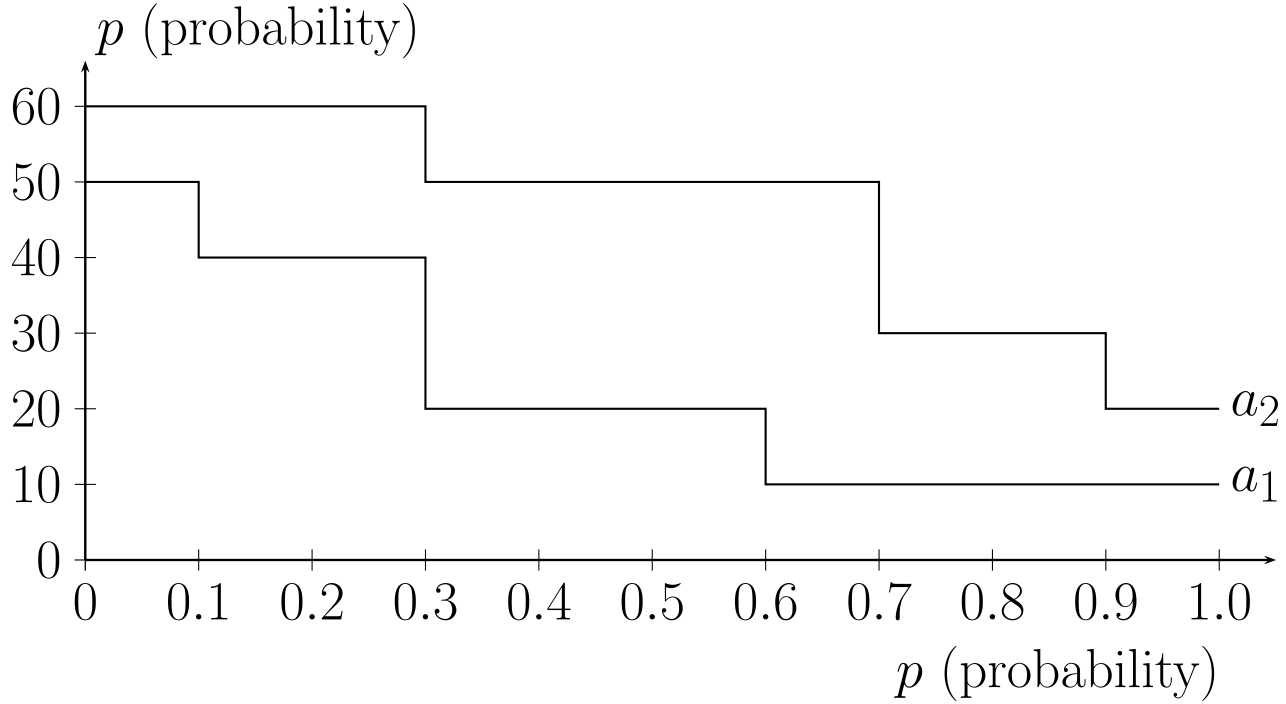
Example:

- a_1 : 50(0.1) 40(0.2) 20(0.3) 10(0.4)
- a_2 : 60(0.3) 50(0.4) 30(0.2) 20(0.1)

Probability Domination

Example:

- a_1 : $\quad 50(0.1) \quad 40(0.2) \quad 20(0.3) \quad 10(0.4)$
- a_2 : $\quad 60(0.3) \quad 50(0.4) \quad 30(0.2) \quad 20(0.1)$



a_2 dominates a_1.

Multi Criteria Decisions - Example

	Sales e_1	Profit e_2	Environment Pollution e_3
a_1	800	7000	-4
a_2	600	7000	-2
a_3	400	6000	0
a_4	200	4000	0

Efficient Alternatives

- Only focus on alternatives which are not dominated by others
- Example: Drop a_4

Finding a decision

- If multiple alternatives are effective we need an algorithm to choose the preferred one
- Simplest algorithm: Chose one target (most important, alphabetical) and optimize for this value

Multi Criteria Decisions - Utility Function

Goal find a function $U(e_1, e_2, \ldots, e_n)$ as a combination of all targets, which could be optimized

Linear combination

- Simplest variant: Linear combination of all targets
- $U(e_1, e_2, \ldots, e_i) = \sum_{i=1}^{n} \omega_i \cdot e_i$

Example

- $\omega_1 = 10, \quad \omega_2 = 1, \quad \omega_3 = 500$

	Sales e_1	Profit e_2	Environment Pollution e_3	$U(e_1, e_2, e_3)$
a_1	800	7000	-4	**13000**
a_2	600	7000	-2	12000
a_3	400	6000	0	10000

Decision under Uncertainty

	z_1	z_2	z_3	z_4
a_1	60	30	50	60
a_2	10	10	10	140
a_3	-30	100	120	130

Think about, how you would decide!

Decision Rules

- Maximin - Rule
- Maximax - Rule
- Hurwicz - Rule
- Savage-Niehans - Rule
- Laplace - Rule

Maximin - Rule

	z_1	z_2	z_3	z_4	Minimum
a_1	60	30	50	60	**30**
a_2	10	10	10	140	10
a_3	-30	100	120	130	-30

Chose the one with the highest minimum

Contra: To pessimistic, only focus on one column

Example

	z_1	z_2	z_3	z_4	Minimum
a_1	1,000,000	1,000,000	0.99	1,000,000	0.99
a_2	1	1	1	1	**1**

Maximax - Rule

	z_1	z_2	z_3	z_4	Maximum
a_1	60	30	50	60	60
a_2	10	10	10	140	**140**
a_3	-30	100	120	130	130

Chose the one with the highest maximum

Contra: To optimistic, only focus on one column

Example

	z_1	z_2	z_3	z_4	Maximum
a_1	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000
a_2	1,000,001	1	1	1	**1,000,001**

Hurwicz - Rule

	z_1	z_2	z_3	z_4	Max	Min	$\Phi(a_i)$
a_1	60	30	50	60	60	30	$0.4 \cdot 60 + 0.6 \cdot 30 = 42$
a_2	10	10	10	140	140	10	$0.4 \cdot 140 + 0.6 \cdot 10 = \mathbf{62}$
a_3	-30	100	120	130	130	-30	$0.4 \cdot 130 + 0.6 \cdot (-30) = 34$

Combination of Maximin and Maximax - Rule

$\Phi(a) = \lambda \cdot \max(e_i) + (1 - \lambda) \cdot \min(e_i)$

λ represents readiness to assume risk

Contra: Only focus on two column

Example $(\min(a_1) < \min(a_2), \max(a_1) < \max(a_2) \Rightarrow \text{chose } a_2)$

	z_1	z_2	z_3	z_4	Max	Min
a_1	1,000,000	1,000,000	1,000,000	0.99	1,000,000	0.99
a_2	1,000,001	1	1	1	**1,000,001**	**1**

Savage-Niehans - Rule

	z_1	z_2	z_3	z_4
a_1	**60**	30	50	60
a_2	10	10	10	**140**
a_3	-30	**100**	**120**	130

Rule of minimal regret

Algorithm:

- Find the maximal value for every column
- Subtract value from maximal value
- Use alternative with the lowest regret

Regret Table:

	z_1	z_2	z_3	z_4	Max
a_1	60 - 60 = 0	70	70	80	**80**
a_2	60 - 10 = 50	90	110	0	110
a_3	60 - (-30) = 90	0	0	10	90

Savage-Niehans - Rule II

	z_1	z_2	z_3	z_4
a_1	1,000	1,000,000	1,000,000	1,000,000
a_2	1,001	0	0	0

Another example

we chose a_1

Regret Table:

	z_1	z_2	z_3	z_4	Max
a_1	1	0	0	0	**1**
a_2	0	1,000,000	1,000,000	1,000,000	1,000,000

Savage-Niehans - Rule III

	z_1	z_2	z_3	z_4
a_1	1,000	1,000,000	1,000,000	1,000,000
a_2	1,001	0	0	0
a_3	2,000,000	-1,000,000	-1,000,000	-1,000,000

Same example, but we add alternative a_3

Now we chose a_2

Regret Table:

	z_1	z_2	z_3	z_4	Max
a_1	1,999,000	0	0	0	1,999,000
a_2	1,998,999	1,000,000	1,000,000	1,000,000	**1,998,999**
a_2	0	2,000,000	2,000,000	2,000,000	2,000,000

	z_1	z_2	z_3	z_4	Mean
a_1	60	30	50	60	50
a_2	10	10	10	140	42.5
a_3	-30	100	120	130	**80**

Chose the one with the highest mean value

Contra:

- Not every condition has the same probability
- Duplication of one condition could change the result

Most people would also chose a_3 in this example

Rule - Axioms

The following axioms should be fulfilled by the rules

Addition to a column
The decision should not be changed, if a fixed value is added to a column

Additional rows
The preference relation between two alternatives should not be changed, if a new row is added

Domination
If a_1 dominates a_2, a_2 could not be optimal

Join of equal columns
The preference relation between to alternatives should not change, if two columns with the same outcomes are joined to a common column

Decision Rules Conclusion

Rule	Example Result	Addition to a row	Additional Rows	Domination	Join of equal Rows
Maximin	a_1		$\surd$		$\surd$
Maximax	a_2		$\surd$		$\surd$
Hurwicz	a_2		$\surd$		$\surd$
Savage-Niehans	a_1	$\surd$		$\surd$	$\surd$
Laplace	a_3	$\surd$	$\surd$	$\surd$	

No Rule fulfills all axioms $\Rightarrow$ no perfect rule

Common usage: Remove duplicate Columns and use Laplace

Better: Define subjective probabilities and use them

Preference Orderings

A *preference ordering* $\succeq$ is a ranking of all possible states of affairs (worlds) S

- these could be outcomes of actions, truth assignments, states in a search problem, etc.
- $s \succeq t$: means that state s is *at least as good as* t
- $s \succ t$: means that state s *is strictly preferred to* t

We insist that $\succeq$ is

- reflexive: i.e., s $\succeq$ s for all states s
- transitive: i.e., if s $\succeq$ t and t $\succeq$ w, then s $\succeq$ w
- connected: for all states s,t, either s $\succeq$ t or t $\succeq$ s

Preference Orderings

Note that transitivity is not always given in decision making

Consider the following set of dice (Efron Dice)

- Die A has sides: 2, 2, 4, 4, 9, 9
- Die B has sides: 1, 1, 6, 6, 8, 8
- Die C has sides: 3, 3, 5, 5, 7, 7

The probability that A rolls a higher number than B, the probability that B rolls higher than C, and the probability that C rolls higher than A are all $5/9$, so this set of dice is nontransitive. In fact, it has the even stronger property that, for each die in the set, there is another die that rolls a higher number than it more than half the time.

Why Impose These Conditions?

Structure of preference ordering imposes certain "rationality requirements" (it is a weak ordering)

E.g., why transitivity?

- Suppose you (strictly) prefer coffee to tea, tea to OJ, OJ to coffee
- If you prefer X to Y, you will trade me Y plus $1 for X
- I can construct a "money pump" and extract arbitrary amounts of money from you

Utilities

Rather than just ranking outcomes, we are often able to quantify our degree of preference

A *utility function* $U : S \to \mathbb{R}$ associates a realvalued *utility* with each outcome.

- $U(s)$ measures the *degree* of preference for s

Note: U induces a preference ordering $\succeq_U$ over S defined as: s $\succeq_U$ t iff $U(s) \geq U(t)$

- $\succeq_U$ will be reflexive, transitive, connected

Expected Utility

Under conditions of uncertainty, each decision d induces a distribution Pr_d over possible outcomes

- $Pr_d(s)$ is probability of outcome s under decision d

The *expected utility* of decision d is defined

$$EU(d) = \sum_{s \in S} Pr_d(s) U(s)$$

The *principle of maximum expected utility (MEU)* states that the optimal decision under conditions of uncertainty is that with the greatest expected utility.

Decision Problems: Uncertainty

A *decision problem under uncertainty* is:

- a set of *decisions* D
- a set of *outcomes* or states S
- an *outcome function* $Pr : D \to \Delta(S)$
  $\Delta(S)$ is the set of distributions over S (e.g., Pr_d)
- ◦ a *utility function* U over S

A solution to a decision problem under uncertainty is any $d^* \in D$ such that $EU(d^*) \succeq EU(d)$ for all $d \in D$

Expected Utility: Notes

Where do utilities come from?

- underlying foundations of utility theory tightly couple utility with action/choice
- a utility function can be determined by asking someone about their preferences for actions in specific scenarios (or "lotteries" over outcomes)

Utility functions needn't be unique

- if I multiply U by a positive constant, all decisions have same relative utility
- if I add a constant to U, same thing
- *U is unique up to positive affine transformation*

Complications

Outcome space is large

- like all of our problems, states spaces can be huge
- don't want to spell out distributions like Pr_d explicitly
- Solution: Bayes nets (or related: *influence diagrams*)

Decision space is large

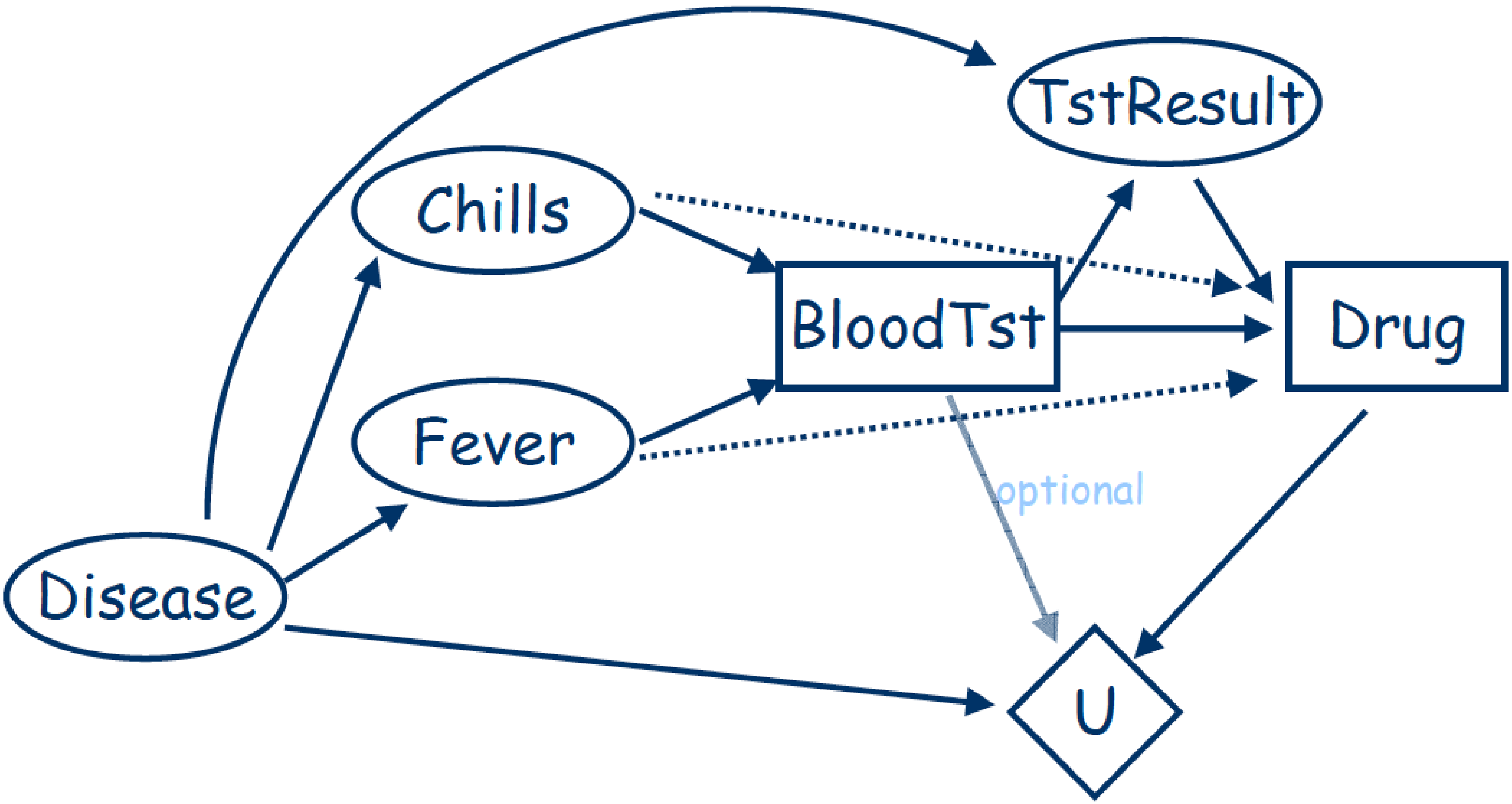
- usually our decisions are not one-shot actions
- rather they involve sequential choices (like plans)
- if we treat each plan as a distinct decision, decision space is too large to handle directly
- Solution: use dynamic programming methods to *construct* optimal plans (actually generalizations of plans, called policies. . . like in game trees)

Decision Networks

Decision networks (also known as *influence diagrams*) provide a way of representing sequential decision problems

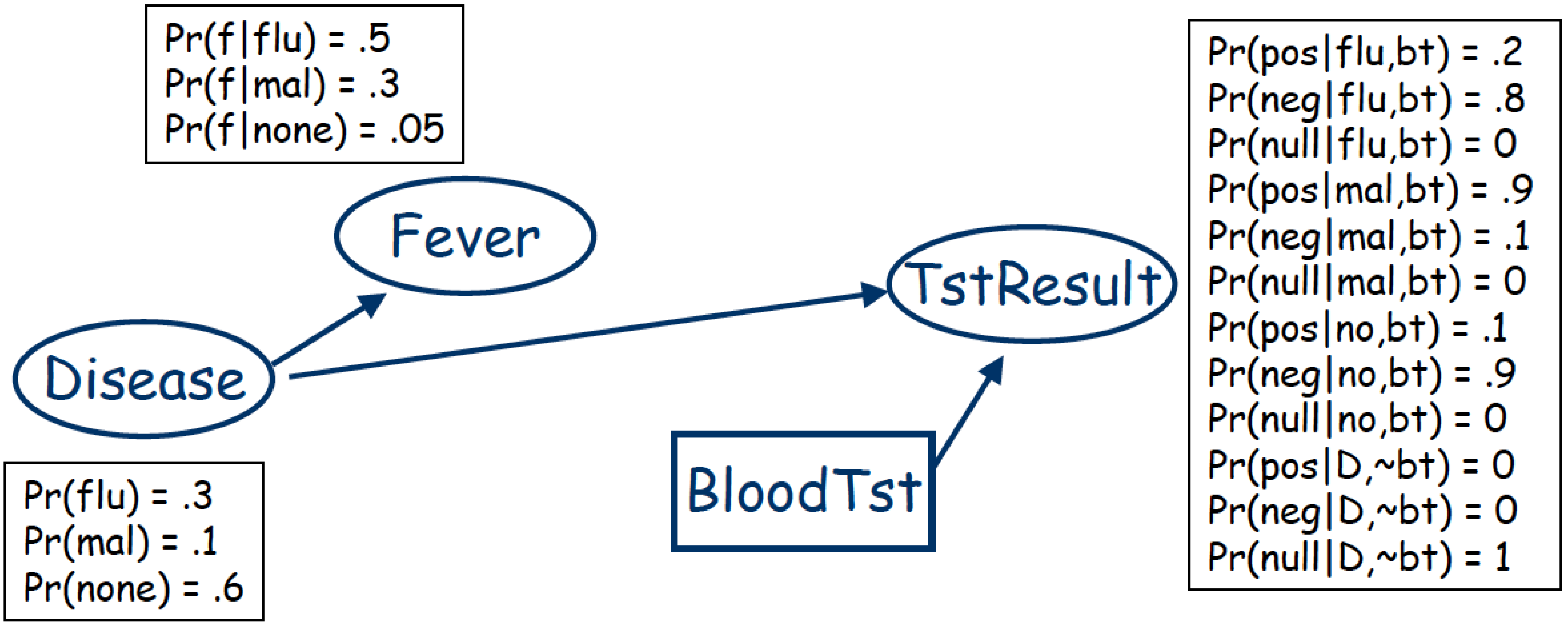
- basic idea: represent the variables in the problem as you would in a BN
- add decision variables – variables that you “control”
- add utility variables – how good different states are

Sample Decision Network



Chance nodes

- random variables, denoted by circles
- as in a BN, probabilistic dependence on parents



Decision nodes

- variables decision maker sets, denoted by squares
- parents reflect *information available* at time decision is to be made

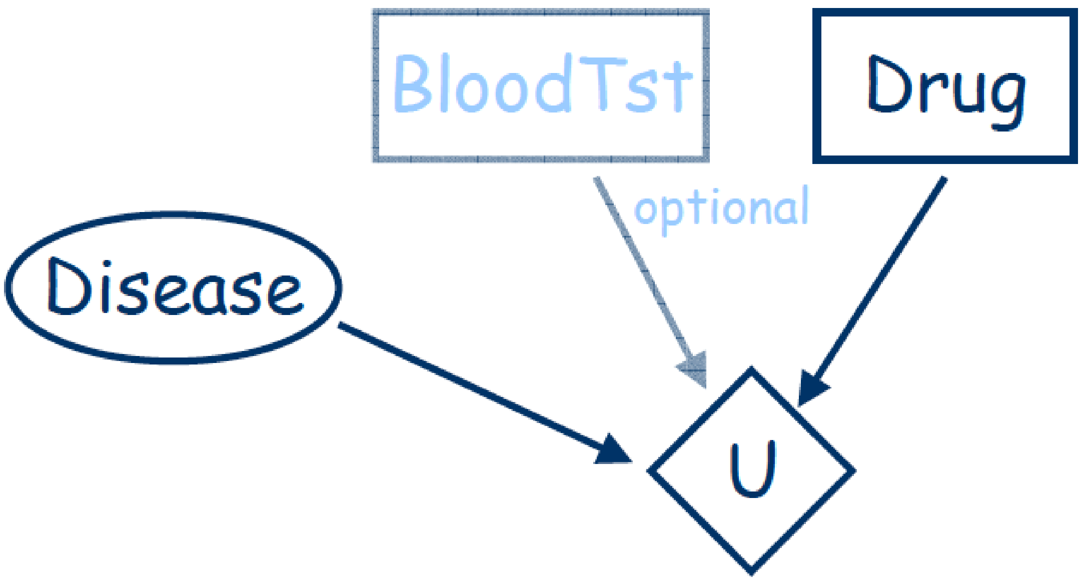
In example decision node: the actual values of Chills and Fever will be observed before the decision to take test must be made

- agent can make different decisions for each instantiation of parents (i.e., policies)

Value node

- specifies utility of a state, denoted by a diamond
- utility depends *only on state of parents* of value node
- generally: only one value node in a decision network

Utility depends only on disease and drug



U(fludrug, flu) = 20
U(fludrug, mal) = -300
U(fludrug, none) = -5
U(maldrug, flu) = -30
U(maldrug, mal) = 10
U(maldrug, none) = -20
U(no drug, flu) = -10
U(no drug, mal) = -285
U(no drug, none) = 30

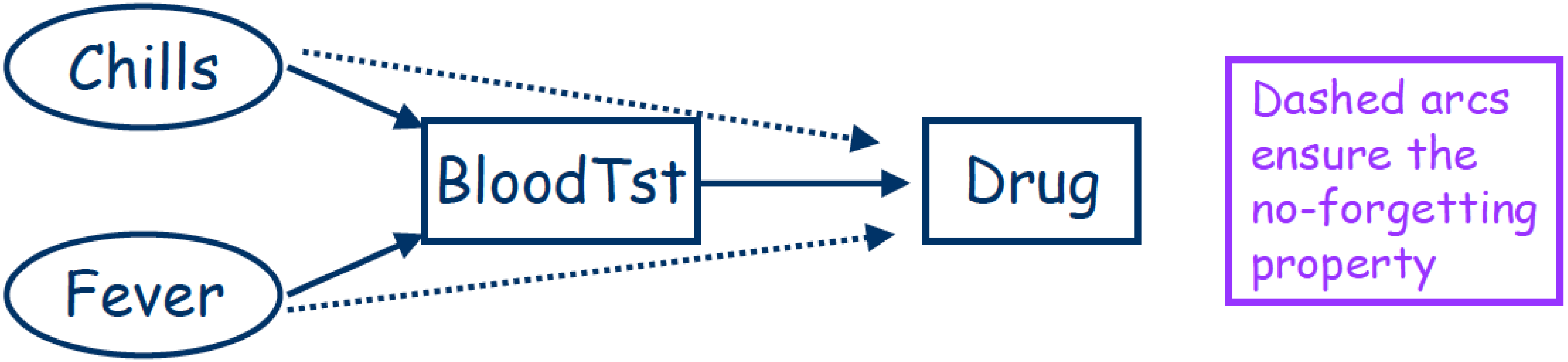
Decision Networks: Assumptions

Decision nodes are totally ordered

- decision variables $D_1, D_2, \ldots, D_n$
- decisions are made in sequence
- e.g., BloodTst (yes,no) decided before Drug (fd,md,no)

No-forgetting property

- any information available when decision D_i is made is available when decision D_j is made (for $i < j$)
- thus all parents of D_i are parents of D_j



Policies

Let $Par(D_i)$ be the parents of decision node D_i

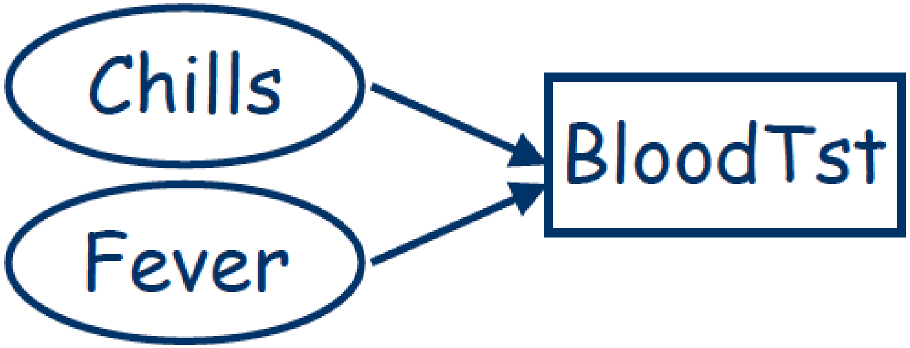
- $Dom(Par(D_i))$ is the set of assignments to parents

A policy δ is a set of mappings δ_i, one for each decision node D_i

- $\delta_i : Dom(Par(D_i)) \to (D_i)$
- δ_i associates a decision with each parent assignment for D_i

For example, a policy for BT might be:

$$\delta_{BT}(c, f) = bt$$
$$\delta_{BT}(c, \sim f) =\sim bt$$
$$\delta_{BT}(\sim c, f) = bt$$
$$\delta_{BT}(\sim c, \sim f) =\sim bt$$



Policies

Value of a policy δ is the expected utility given that decision nodes are executed according to δ

Given associates $\boldsymbol{x}$ to the set $\boldsymbol{X}$ of all chance variables, let $\delta(\boldsymbol{x})$ denote the assignment to decision variables dictated by δ

- e.g., assigned to D_1 determined by it's parents' assignment in $\boldsymbol{x}$
- e.g., assigned to D_2 determined by it's parents' assignment in $\boldsymbol{x}$ along with whatever was assigned to D1
- etc.

Value of δ:

$$EU(\delta) = \sum_{\boldsymbol{X}} P(\boldsymbol{X}, \delta(\boldsymbol{X})U(\boldsymbol{X}, \delta(\boldsymbol{X}))$$

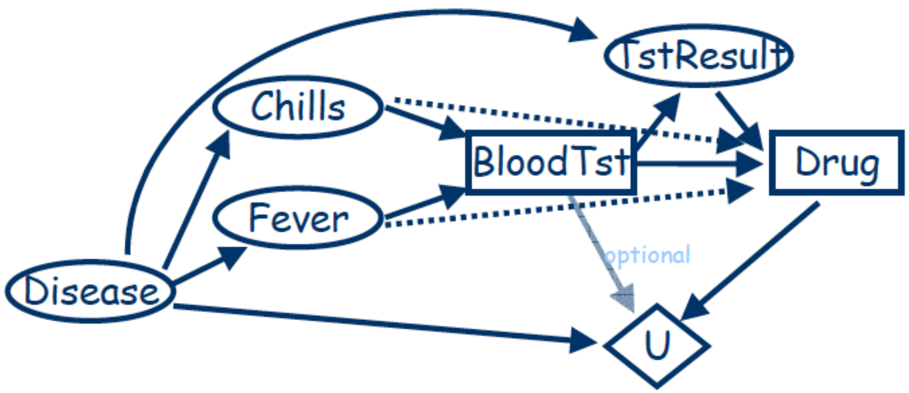
An *optimal policy* is a policy δ^* such that $EU(\delta^*) \geq EU(\delta)$ for all policies δ

Computing the Best Policy

We can work backwards as follows

First compute optimal policy for Drug (last decision)

- for each assignment to parents (C,F,BT,TR) and for each decision value (D = md,fd,none), *compute the expected value* of choosing that value of D
- set policy choice for each value of parents to be the value of D that has max value
- eg: $\delta_D(c, f, bt, pos) = md$



Computing the Best Policy

Next compute policy for BT given policy $\delta_D(C, F, BT, TR)$ just determined for Drug

- since $\delta_D(C, F, BT, TR)$ is fixed, we can treat Drug as a normal random variable with deterministic probabilities
- i.e., for any instantiation of parents, value of Drug is fixed by policy δ_D
- this means we can solve for optimal policy for BT just as before
- only uninstantiated variables are random variables (once we fix *its* parents)

Example

You want to buy a used car, but there's a good chance it is a "lemon" (i.e., prone to breakdown). Before deciding to buy it, you can take it to a mechanic for inspection. S/he will give you a report on the car, labelling it either "good" or "bad". A good report is positively correlated with the car being sound, while a bad report is positively correlated with the car being a lemon.

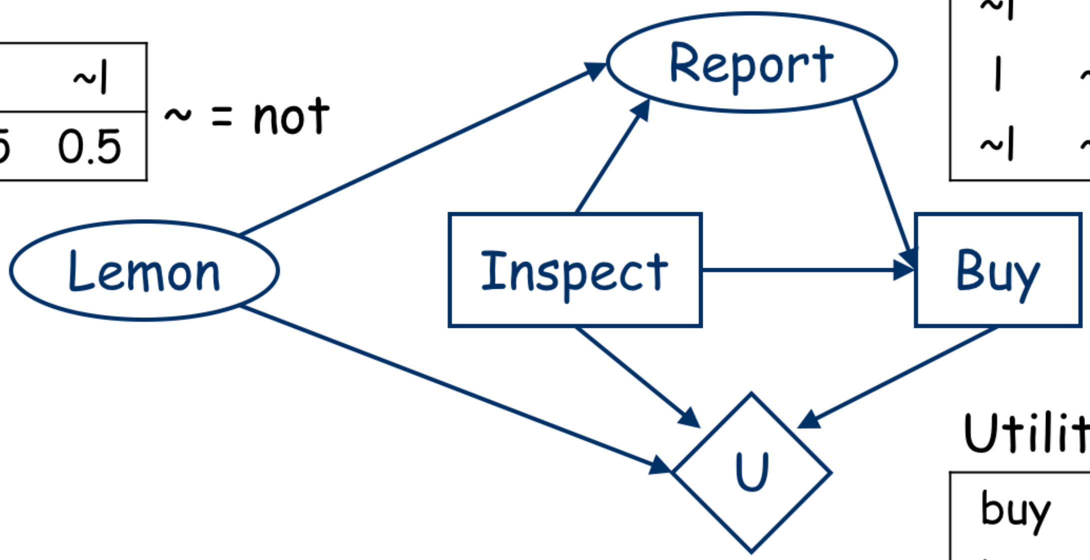
The report costs $50 however. So you could risk it, and buy the car without the report.

Owning a sound car is better than having no car, which is better than owning a lemon.

Car Buyer's Network

l	~l
0.5	0.5

~ = not



Report: good, bad, none

		g	b	n
l	i	0.2	0.8	0
~l	i	0.9	0.1	0
l	~i	0	0	1
~l	~i	0	0	1

Utility for Lemon and Buy

buy	l	-600
buy	~l	1000
~buy	l	-300
~buy	~l	-300

-50 if inspect

Evaluate Last Decision: Buy (1)

$EU(B|I,R) = \sum_L P(L|I,R,B)U(L,B)$

$I = i, R = g$:

$$\begin{aligned}
EU(buy) &= P(l|i,g)U(l,buy) + P(\sim l|i,g)U(\sim l,buy) - 50 \\
&= 0.18 \cdot (-600) + 0.82 \cdot 1000 - 50 = 662 \\
EU(\sim buy) &= P(l|i,g)U(l,\sim buy) + P(\sim l|i,g)U(\sim l,\sim buy) - 50 \\
&= -300 - 50 = -350 (-300 \text{ indep. of lemon})
\end{aligned}$$

So optimal $\delta_{Buy}(i,g) = buy$

$I = i, R = b$:

$$\begin{aligned}
EU(buy) &= P(l|i,b)U(l,buy) + P(\sim l|i,b)U(\sim l,buy) - 50 \\
&= 0.89 \cdot (-600) + .11 \cdot 1000 - 50 = -474 \\
EU(\sim buy) &= P(l|i,b)U(l,\sim buy) + P(\sim l|i,b)U(\sim l,\sim buy) - 50 \\
&= -300 - 50 = -350 (-300 \text{ indep. of lemon})
\end{aligned}$$

So optimal $\delta_{Buy}(i,b) = \sim buy$

Evaluate Last Decision: Buy (2)

$I =\sim i, R = n$ (note: no inspection cost subtracted):

$$\begin{aligned} EU(buy) &= P(l|\sim i, n)U(l, buy) + P(\sim l|\sim i, n)U(\sim l, buy) \\ &= 0.5 \cdot (-600) + 0.5 \cdot 1000 = 200 \end{aligned}$$

$$\begin{aligned} EU(\sim buy) &= P(l|\sim i, n)U(l, \sim buy) + P(\sim l|\sim i, n)U(\sim l, \sim buy) - 50 \\ &= -300 - 50 = -350 (-300 \text{ indep. of lemon}) \end{aligned}$$

So optimal $\delta_{Buy}(\sim i, g) = buy$

So optimal policy for Buy is:

- $\delta_{Buy}(i, g) = buy; \delta_{Buy}(i, b) =\sim buy; \delta_{Buy}(\sim i, g) = buy$

Note: we don't bother computing policy for $(i, \sim g)$, $(\sim i, g)$, or $(\sim i, b)$, since these occur with probability 0

Evaluate First Decision: Inspect

$EU(I) = \sum_{L,R} P(L, R|I) U(L, \boldsymbol{\delta_{Buy}(I, R)})$,
where $P(R, L|I) = P(R|L, I) P(L|I)$

$$\begin{aligned} EU(i) &= 0.1 \cdot (-650) + 0.4 \cdot (-300) + 0.45 \cdot 1000 + 0.05 \cdot (-300) - 50 \\ &= 187.5 \\ EU(\sim i) &= P(l| \sim i, n) U(l, buy) + P(\sim l| \sim i, n) U(\sim l, buy) \\ &= .5 \cdot -600 + .5 \cdot 1000 = 200 \end{aligned}$$

So optimal $\delta_{Inspect}(\sim i) = buy$

	$P(R, L\|I)$	$\boldsymbol{\delta_{Buy}}$	$U(L, \boldsymbol{\delta_{Buy}})$
g, l	0.1	buy	$-600 - 50 = -650$
$g, \sim l$	0.45	buy	$1000 - 50 = 950$
b, l	0.4	$\sim buy$	$-300 - 50 = -350$
$b, \sim l$	0.05	$\sim buy$	$-300 - 50 = -350$

Value of Information

So optimal policy is: don't inspect, buy the car

- EU = 200

- Notice that the EU of inspecting the car, then buying it iff you get a good report, is 237.5 less the cost of the inspection (50). So inspection not worth the improvement in EU.

- But suppose inspection cost \$25: then it would be worth it ($EU = 237.5 - 25 = 212.5 > EU(\sim i)$)

- The *expected value of information* associated with inspection is 37.5 (it improves expected utility by this amount ignoring cost of inspection). How? Gives opportunity to change decision ($\sim$ *buy* if bad).

- You should be willing to pay up to \$37.5 for the report

Nonstandard Concepts for Handling Imprecise Data and Imprecise Probabilities

Problems with Probability Theory

Representation of Ignorance

We are given a die with faces $1, \ldots, 6$
What is the certainty of showing up face i ?

- Conduct a statistical survey (roll the die 10000 times) and estimate the relative frequency: $P(\{i\}) = \frac{1}{6}$
- Use subjective probabilities (which is often the normal case): We do not know anything (especially and explicitly we do not have any reason to assign unequal probabilities), so the most plausible distribution is a uniform one.

  Problem: Uniform distribution because of ignorance or extensive statistical tests

Experts analyze aircraft shapes: 3 aircraft types A, B, C
"It is type A or B with 90% certainty. About C, I don't have any clue and I do not want to commit myself. No preferences for A or B."

Problem: Ignorance hard to handle with Bayesian theory

Random Sets: Modeling Imprecise Data

"$A \subseteq X$ being an imprecise date" means: the true value x_0 lies in A but there are no preferences on A.

Ω	set of possible elementary events
$\Theta = \{\xi\}$	set of observers
$\lambda(\xi)$	importance of observer ξ

Some elementary event from Ω occurs and every observer $\xi \in O$ shall announce which elementary events she personally considers possible. This set is denoted by $\Gamma(\xi) \subseteq \Omega$. $\Gamma(\xi)$ is then an imprecise date.

$\lambda : 2^{\Theta} \to [0, 1]$	probability measure (interpreted as importance measure)
$(\Theta, 2^{\Theta}, \lambda)$	probability space
$\Gamma : \Theta \to 2^{\Omega}$	set-valued mapping

Imprecise Data (2)

Let $A \subseteq \Omega$:

a) $\Gamma^*(A) \stackrel{\text{Def}}{=} \{\xi \in \Theta \mid \Gamma(\xi) \cap A \neq \emptyset\}$

b) $\Gamma_*(A) \stackrel{\text{Def}}{=} \{\xi \in \Theta \mid \Gamma(\xi) \neq \emptyset \text{ and } \Gamma(\xi) \subseteq A\}$

Remarks:

a) If $\xi \in \Gamma^*(A)$, then it is *plausible* for ξ that the occurred elementary event lies in A.

b) If $\xi \in \Gamma_*(A)$, then it is *certain* for ξ that the event lies in A.

c) $\{\xi \mid \Gamma(\xi) \neq \emptyset\} \;=\; \Gamma^*(\Omega) \;=\; \Gamma_*(\Omega)$

Let $\lambda(\Gamma^*(\Omega)) > 0$. Then we call

$$P^*(A) = \frac{\lambda(\Gamma^*(A))}{\lambda(\Gamma^*(\Omega))} \qquad \text{the upper, and} \qquad P_*(A) = \frac{\lambda(\Gamma_*(A))}{\lambda(\Gamma_*(\Omega))} \qquad \text{the lower}$$

probability w. r. t. λ and Γ.

Example

$$\begin{aligned} \Theta &= \{a,b,c,d\} \\ \Omega &= \{1,2,3\} \\ \Gamma^*(\Omega) &= \{a,b,d\} \\ \lambda(\Gamma^*(\Omega)) &= {}^4\!/_6 \end{aligned} \qquad \begin{aligned} \lambda\colon\ a &\mapsto {}^1\!/_6 \\ b &\mapsto {}^1\!/_6 \\ c &\mapsto {}^2\!/_6 \\ d &\mapsto {}^2\!/_6 \end{aligned} \qquad \begin{aligned} \Gamma\colon\ a &\mapsto \{1\} \\ b &\mapsto \{2\} \\ c &\mapsto \emptyset \\ d &\mapsto \{2,3\} \end{aligned}$$

A	$\Gamma^*(A)$	$\Gamma_*(A)$	$P^*(A)$	$P_*(A)$
$\emptyset$	$\emptyset$	$\emptyset$	0	0
$\{1\}$	$\{a\}$	$\{a\}$	$\frac{1}{4}$	$\frac{1}{4}$
$\{2\}$	$\{b,d\}$	$\{b\}$	$\frac{3}{4}$	$\frac{1}{4}$
$\{3\}$	$\{d\}$	$\emptyset$	$\frac{1}{2}$	0
$\{1,2\}$	$\{a,b,d\}$	$\{a,b\}$	1	$\frac{1}{2}$
$\{1,3\}$	$\{a,d\}$	$\{a\}$	$\frac{3}{4}$	$\frac{1}{4}$
$\{2,3\}$	$\{b,d\}$	$\{b,d\}$	$\frac{3}{4}$	$\frac{3}{4}$
$\{1,2,3\}$	$\{a,b,d\}$	$\{a,b,d\}$	1	1

One can consider $P^*(A)$ and $P_*(A)$ as upper and lower probability bounds.

Imprecise Data (3)

Some properties of probability bounds:

a) $P^*\colon\; 2^\Omega \to [0,1]$

b) $0 \le P_* \le P^* \le 1, \qquad P_*(\emptyset) = P^*(\emptyset) = 0, \qquad P_*(\Omega) = P^*(\Omega) = 1$

c) $A \subseteq B \quad \Rightarrow \quad P^*(A) \le P^*(B) \quad \text{and} \quad P_*(A) \le P_*(B)$

d) $A \cap B = \emptyset \quad \not\Rightarrow \quad P^*(A) + P^*(B) = P^*(A \cup B)$

e) $P_*(A \cup B) \ge P_*(A) + P_*(B) - P_*(A \cap B)$

f) $P^*(A \cup B) \le P^*(A) + P^*(B) - P^*(A \cap B)$

g) $P_*(A) = 1 - P^*(\Omega \backslash A)$

One can prove the following generalized equation:

$$P_*(\bigcup_{i=1}^{n} A_i) \quad \geq \sum_{\emptyset \neq I : I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \cdot P_*(\bigcap_{i \in I} A_i)$$

These set functions also play an important role in theoretical physics (capacities, Choquet, 1955). Shafer did generalize these thoughts and developed a theory of belief functions.

Belief Revision

How is new knowledge incoporated?

Every observer announces the location of the ship in form of a subset of all possible ship locations. Given these set-valued mappings, we can derive upper and lower probabilities with the help of the observer importance measure. Let us assume the ship is certainly at sea.

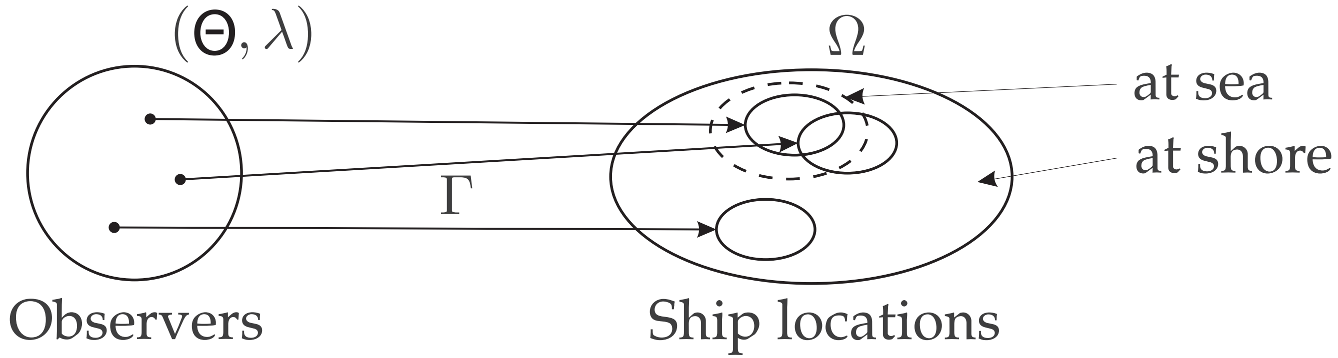
How do the upper/lower probabilities change?

Example

a) *Geometric Conditioning*
(observers that give partial or full wrong information are discarded)

$$P_*(A \mid B) = \frac{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq A \text{ and } \Gamma(\xi) \subseteq B\})}{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq B\})} = \frac{P_*(A \cap B)}{P_*(B)}$$

$$P^*(A \mid B) = \frac{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq B \text{ and } \Gamma(\xi) \cap A \neq \emptyset\})}{\lambda(\{\xi \in \Theta \mid \Gamma(\xi) \subseteq B\})} = \frac{P^*(A \cup \overline{B}) - P^*(\overline{B})}{1 - P^*(\overline{B})}$$

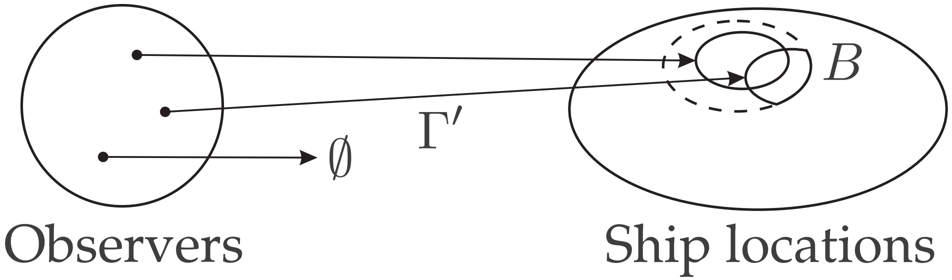


Belief Revision (2)

b) *Data Revision*
(the observed data is modified such that they fit the certain information)

$$(P_*)_B(A) = \frac{P_*(A \cup \overline{B}) - P_*(\overline{B})}{1 - P_*(B)}$$

$$(P^*)_B(A) = \frac{P^*(A \cap B)}{P^*(B)}$$



These two concepts have different semantics. There are several more belief revision concepts.

Combination of Random Sets

Let $(\Omega, 2^\Omega)$ be a space of events. Further be $(O_1, 2^{O_1}, \lambda_1)$ and $(O_2, 2^{O_2}, \lambda_2)$ spaces of independent observers.

We call $(O_1 \times O_2, \lambda_1 \cdot \lambda_2)$ the product space of observers and

$$\Gamma : O_1 \times O_2 \to 2^\Omega, \Gamma(x_1, x_2) = \Gamma_1(x_1) \cap \Gamma_2(x_2)$$

the combined observer function.

We obtain with

$$(P_L)_*(A) = \frac{(\lambda_1 \cdot \lambda_2)(\{(x_1, x_2) \mid \Gamma(x_1, x_2) \neq \emptyset \wedge \Gamma(x_1, x_2) \sqsubseteq A\})}{(\lambda_1 \cdot \lambda_2)(\{(x_1, x_2 \mid \Gamma(x_1, x_2) \neq \emptyset)\})}$$

the lower probability of A that respects both observations.

Example

$$\Omega = \{1,2,3\} \qquad \lambda_1\colon\ \{a\} \mapsto {}^1\!/\!_3 \qquad \lambda_2\colon\ \{c\} \mapsto {}^1\!/\!_2$$
$$\{b\} \mapsto {}^2\!/\!_3 \qquad \lambda_2\colon\ \{d\} \mapsto {}^1\!/\!_2$$

$$\begin{aligned} O_1 &= \{a,b\} & \Gamma_1\colon\ a &\mapsto \{1,2\} & \Gamma_2\colon\ c &\mapsto \{1\} \\ O_2 &= \{c,d\} & b &\mapsto \{2,3\} & d &\mapsto \{2,3\} \end{aligned}$$

Combination:

$$O_1 \times O_2 \ = \ \{\overline{ac}, \overline{bc}, \overline{ad}, \overline{bd}\}$$

$$\begin{aligned} \lambda\colon\ \{\overline{ac}\} &\mapsto {}^1\!/\!_6 & \Gamma\colon\ \overline{ac} &\mapsto \{1\} & \Gamma_*(\Omega) &= \{(x_1,x_2) \mid \Gamma(x_1,x_2) \neq \emptyset\} \\ \{\overline{ad}\} &\mapsto {}^1\!/\!_6 & \overline{ad} &\mapsto \{2\} & &= \{\overline{ac}, \overline{ad}, \overline{bd}\} \\ \{\overline{bc}\} &\mapsto {}^2\!/\!_6 & \overline{bc} &\mapsto \emptyset & & \\ \{\overline{bd}\} &\mapsto {}^2\!/\!_6 & \overline{bd} &\mapsto \{2,3\} & \lambda(\Gamma_*(\Omega)) &= {}^4\!/\!_6 \end{aligned}$$

Example (2)

A	$(P_*)_{\Gamma_1}(A)$	$(P_*)_{\Gamma_2}(A)$	$(P_*)_{\Gamma}(A)$
$\emptyset$	0	0	0
$\{1\}$	0	$1/2$	$1/4$
$\{2\}$	0	0	$1/4$
$\{3\}$	0	0	0
$\{1, 2\}$	$1/3$	$1/2$	$1/2$
$\{1, 3\}$	0	$1/2$	$1/4$
$\{2, 3\}$	$2/3$	$1/2$	$3/4$
$\{1, 2, 3\}$	1	1	1

Motivation

(Θ, Q)	Sensors	
Ω	possible results, $\Gamma : \Theta \to 2^{\Omega}$	
$P_* :$	$A \mapsto \sum_{B:B\subseteq A} m(B)$	Lower probability (Belief)
$P^* :$	$A \mapsto \sum_{B:B\cap A\neq\emptyset} m(B)$	Upper probability (Plausibility)
$m :$	$A \mapsto Q(\{\theta \in \Theta \mid \Gamma(\theta) = A\})$	mass distribution

Random sets: Dempster (1968)

Belief functions: Shafer (1974)
Development of a completely new uncertainty calculus as an alternative to Probability Theory

Belief Functions (2)

The function $\mathrm{Bel} : 2^\Omega \to [0,1]$ is called *belief function*, if it possesses the following properties:

$$\mathrm{Bel}(\emptyset) = 0$$

$$\mathrm{Bel}(\Omega) = 1$$

$$\forall n \in \mathbb{N}\colon\ \forall A_1, \ldots, A_n \in 2^\Omega :$$
$$\mathrm{Bel}(A_1 \cup \cdots \cup A_n) \geq \textstyle\sum_{\emptyset \neq I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \cdot \mathrm{Bel}(\bigcap_{i \in I} A_i)$$

If Bel is a belief function then for $m : 2^\Omega \to \mathbb{R}$ with $m(A) = \sum_{B : B \subseteq A} (-1)^{|A \setminus B|} \cdot \mathrm{Bel}(B)$ the following properties hold:

$$0 \leq m(A) \leq 1$$

$$m(\emptyset) = 0$$

$$\textstyle\sum_{A \subseteq \Omega} m(A) = 1$$

Belief Functions (3)

Let $|\Omega| < \infty$ and $f, g : 2^\Omega \to [0, 1]$.

$$\forall A \subseteq \Omega \colon \left(f(A) = \sum_{B : B \subseteq A} g(B)\right)$$

$$\Leftrightarrow$$

$$\forall A \subseteq \Omega \colon \left(g(A) = \sum_{B : B \subseteq A} (-1)^{|A \setminus B|} \cdot f(B)\right)$$

(g is called the *Möbius transformed* of f)

The mapping $m : 2^\Omega \to [0, 1]$ is called a *mass distribution*, if the following properties hold:

$m(\emptyset) = 0$

$\sum_{A \subseteq \Omega} m(A) = 1$

Example

A	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{2,3\}$	$\{1,3\}$	$\{1,2,3\}$
$m(A)$	0	$^1\!/_4$	$^1\!/_4$	0	0	0	$^2\!/_4$	0
$\mathrm{Bel}(A)$	0	$^1\!/_4$	$^1\!/_4$	0	$^2\!/_4$	$^1\!/_4$	$^3\!/_4$	1

Belief $\widehat{=}$ lower probability with modified semantic

$$\mathrm{Bel}(\{1,3\}) = m(\emptyset) + m(\{1\}) + m(\{3\}) + m(\{1,3\})$$
$$m(\{1,3\}) = \mathrm{Bel}(\{1,3\}) - \mathrm{Bel}(\{1\}) - \mathrm{Bel}(\{3\})$$

$m(A)$ measure of the trust/belief that exactly A occurs

$\mathrm{Bel}_m(A)$ measure of total belief that A occurs

$\mathrm{Pl}_m(A)$ measure of not being able to disprove A (plausibility)

$$\mathrm{Pl}_m(A) = \sum_{B:A\cap B\neq\emptyset} m(B) = 1 - \mathrm{Bel}(\overline{A})$$

Given one of m, Bel or Pl, the other two can be efficiently computed.

Knowledge Representation

$m(\Omega) = 1,\, m(A) = 0$ else	total ignorance
$m(\{\omega_0\}) = 1,\, m(A) = 0$ else	value (ω_0) known
$m(\{\omega_i\}) = p_i, \sum_{i=1}^{n} p_i = 1$	Bayesian analysis

Further kinds of partial ignorance can be modeled.

Belief Revision

Data Revision:

- Mass of A flows onto $A \cap B$.
- Masses are normalized to 1 ($\emptyset$-mass is destroyed)

Geometric Conditioning:

- Masses that do not lie completely inside B, flow off
- Normalize

The mass flow can be described by specialization matrices

Combinations of Mass Distributions

Motivation: Combination of m_1 and m_2

$m_1(A_i) \cdot m_2(B_j)$:	Mass attached to $A_i \cap B_j$, if only A_i or B_j are concerned
$\sum_{i,j:A_i \cap B_j = A} m_1(A_i) \cdot m_2(B_j)$:	Mass attached to A (after combination)

This consideration only leads to a mass distribution,
if $\sum_{i,j:A_i \cap B_j = \emptyset} m_1(A_i) \cdot m_2(B_j) = 0$.

If this sum is > 0 normalization takes place.

Combination Rule

If m_1 and m_2 are mass distributions over Ω with belief functions Bel_1 and Bel_2 and does further hold $\sum_{i,j:A_i \cap B_j = \emptyset} m_1(A_i) \cdot m_2(B_j) < 1$, then the function $m : 2^\Omega \to [0,1]\,, m(\emptyset) = 0$

$$m(A) = \frac{\sum_{B,C:B \cap C = A} m_1(B) \cdot m_2(C)}{1 - \sum_{B,C:B \cap C = \emptyset} m_1(B) \cdot m_2(C)}$$

is a mass distribution. The belief function of m is denoted as $\mathrm{comb}(\mathrm{Bel}_1, \mathrm{Bel}_2)$ or $\mathrm{Bel}_1 \oplus \mathrm{Bel}_2$. The above formula is called the combination rule.

Example

$$m_1(\{1,2\}) = {}^1\!/_3 \qquad m_2(\{1\}) = {}^1\!/_2$$
$$m_1(\{2,3\}) = {}^2\!/_3 \qquad m_2(\{2,3\}) = {}^1\!/_2$$

$$m = m_1 \oplus m_2:$$
$$\begin{aligned}
\{1\} &\mapsto \frac{{}^1\!/_6}{{}^4\!/_6} = {}^1\!/_4 \\
\{2\} &\mapsto \frac{{}^1\!/_6}{{}^4\!/_6} = {}^1\!/_4 \\
\emptyset &\mapsto 0 \\
\{2,3\} &\mapsto \frac{{}^2\!/_6}{{}^4\!/_6} = {}^1\!/_2
\end{aligned}$$

Combination Rule (2)

Remarks:

a) The result from the combination rule and the analysis of random sets is identical

b) There are more efficient ways of combination

c) $\mathrm{Bel}_1 \oplus \mathrm{Bel}_2 = \mathrm{Bel}_2 \oplus \mathrm{Bel}_1$

d) $\oplus$ is associative

e) $\mathrm{Bel}_1 \oplus \mathrm{Bel}_1 \neq \mathrm{Bel}_1$ (in general)

f) $\mathrm{Bel}_2 : 2^\Omega \to [0,1]\,, m_2(B) = 1$

$$\mathrm{Bel}_2(A) = \begin{cases} 1 & \text{if} B \subseteq A \\ 0 & \text{otherwise} \end{cases}$$

The combination of Bel_1 and Bel_2 yields the data revision of m_1 with B.

Decision Making with the Pignistic Transformation

The **pignistic transformation** Bet transforms a normalized mass function m into a probability measure $P_m = Bet(m)$ as follows:

$$P_m(A) = \sum_{\emptyset \neq B \subseteq \Omega} m(B) \frac{|A \cap B|}{|B|}, \forall A \subseteq \Omega.$$

It can be shown that

$$bel(A) \leq P_m(A) \leq pl(A)$$

Decision Making - Example

There are three possible murders

Let $m(\{John\}) = 0.48, \quad m(\{John, Mary\}) = 0.12,$
$m(\{Peter, John\}) = 0.32, \quad m(\Omega) = 0.08$

We have:

$$P_m(\{John\}) = 0.48 + \frac{0.12}{2} + \frac{0.32}{2} + \frac{0.08}{3} \approx 0.73$$

$$P_m(\{Peter\}) = \frac{0.32}{2} + \frac{0.08}{3} \approx 0.19$$

$$P_m(\{Mary\}) = \frac{0.12}{2} + \frac{0.08}{3} \approx 0.09$$

The picmistic transformation givs a reasonable "Ranking"

Imprecise Probabilities

Let x_0 be the true value but assume there is no information about $P(A)$ to decide whether $x_0 \in A$. There are only probability boundaries.

Let $\mathcal{L}$ be a set of probability measures. Then we call

$$(P_{\mathcal{L}})_* : 2^{\Omega} \to [0,1]\,, A \mapsto \inf\{P(A) \mid P \in \mathcal{L}\} \qquad \text{the lower and}$$

$$(P_{\mathcal{L}})^* : 2^{\Omega} \to [0,1]\,, A \mapsto \sup\{P(A) \mid P \in \mathcal{L}\} \qquad \text{the upper}$$

probability of A w. r. t. $\mathcal{L}$.

a) $(P_{\mathcal{L}})_*(\emptyset) = (P_{\mathcal{L}})^*(\emptyset) = 0; \quad (P_{\mathcal{L}})_*(\Omega) = (P_{\mathcal{L}})^*(\Omega) = 1$

b) $0 \leq (P_{\mathcal{L}})_*(A) \leq (P_{\mathcal{L}})^*(A) \leq 1$

c) $(P_{\mathcal{L}})^*(A) = 1 - (P_{\mathcal{L}})_*(\overline{A})$

d) $(P_{\mathcal{L}})_*(A) + (P_{\mathcal{L}})_*(B) \leq (P_{\mathcal{L}})_*(A \cup B)$

e) $(P_{\mathcal{L}})_*(A \cap B) + (P_{\mathcal{L}})_*(A \cup B) \not\geq (P_{\mathcal{L}})_*(A) + (P_{\mathcal{L}})_*(B)$

Belief Revision

Let $B \subseteq \Omega$ and $\mathcal{L}$ a class of probabilities. The we call

$$A \subseteq \Omega : (P_{\mathcal{L}})_*(A \mid B) = \inf\{P(A \mid B) \mid P \in \mathcal{L} \wedge P(B) > 0\} \qquad \text{the lower and}$$
$$A \subseteq \Omega : (P_{\mathcal{L}})^*(A \mid B) = \sup\{P(A \mid B) \mid P \in \mathcal{L} \wedge P(B) > 0\} \qquad \text{the upper}$$

conditional probability of A given B.

A class $\mathcal{L}$ of probability measures on $\Omega = \{\omega_1, \ldots, \omega_n\}$ is of type 1, iff there exist functions R_1 and R_2 from 2^Ω into $[0, 1]$ with:

$$\mathcal{L} = \{P \mid \forall A \subseteq \Omega : R_1(A) \leq P(A) \leq R_2(A)\}$$

Belief Revision (2)

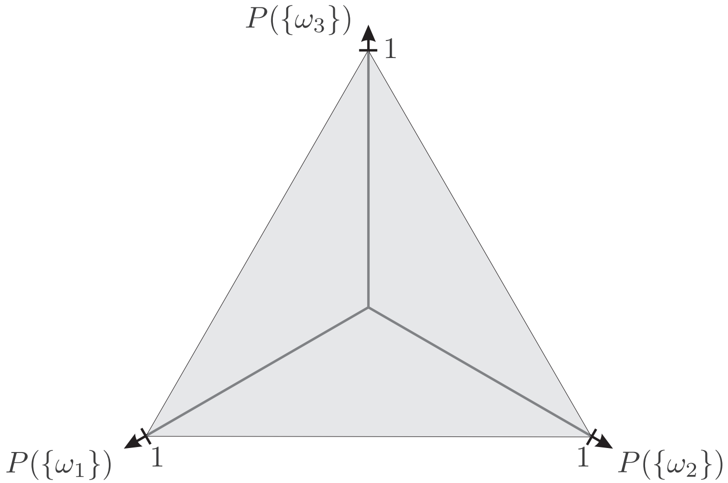
Intuition: P is determined by $P(\{\omega_i\})$, $i = 1, \ldots, n$ which corresponds to a point in $\mathbb{R}^n$ with coordinates $\big(P(\{\omega_1\}), \ldots, P(\{\omega_n\})\big)$.

If $\mathcal{L}$ is type 1, it holds true that:

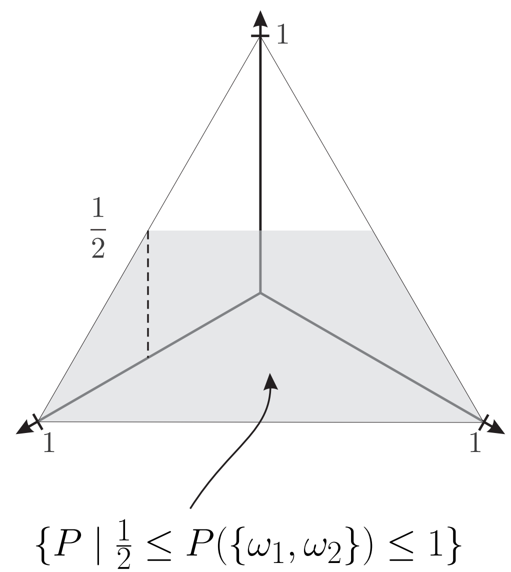
$$\begin{aligned}\mathcal{L} \Leftrightarrow \Big\{(r_1, \ldots, r_n) \in \mathbb{R}^n \mid \exists P\colon \forall A \subseteq \Omega\colon & \\ (P_{\mathcal{L}})_*(A) \leq P(A) \leq (P_{\mathcal{L}})^*(A) & \\ \text{and}\ \ r_i = P(\{\omega_i\}),\ i = 1, \ldots, n\Big\} &\end{aligned}$$

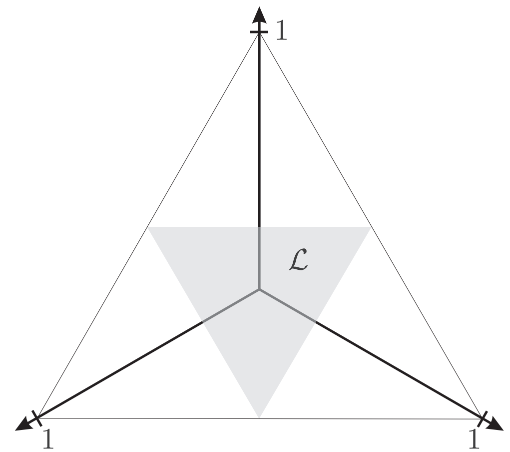
$$\Omega = \{\omega_1, \omega_2, \omega_3\}$$

$$\mathcal{L} = \{P \mid \tfrac{1}{2} \leq P(\{\omega_1, \omega_2\}) \leq 1,\quad \tfrac{1}{2} \leq P(\{\omega_2, \omega_3\}) \leq 1,\quad \tfrac{1}{2} \leq P(\{\omega_1, \omega_3\}) \leq 1\}$$



general restriction:
$0 \leq P(\{\omega_i\}) \leq 1$
$P(\{\omega_1\}) + P(\{\omega_2\}) + P(\{\omega_3\}) = 1$





Let $A_1 = \{\omega_1, \omega_2\}$, $A_2 = \{\omega_2, \omega_3\}$, $A_3 = \{\omega_1, \omega_3\}$

$$\begin{aligned} & P_*(A_1) + P_*(A_2) + P_*(A_3) - P_*(A_1 \cap A_2) - P_*(A_2 \cap A_3) - P_*(A_1 \cap A_3) + P_*(A_1 \cap A_2 \cap A_3) \\ &= \frac{1}{2} + \frac{1}{2} + \frac{1}{2} - 0 - 0 - 0 + 0 = \frac{3}{2} > 1 = P(A_1 \cup A_2 \cup A_3) \end{aligned}$$

Belief Revision (3)

If $\mathcal{L}$ is type 1 and $(P_{\mathcal{L}})^*(A \cup B) \geq (P_{\mathcal{L}})^*(A) + (P_{\mathcal{L}})^*(B) - (P_{\mathcal{L}})^*(A \cap B)$, then

$$(P_{\mathcal{L}})^*(A \mid B) = \frac{(P_{\mathcal{L}})^*(A \cap B)}{(P_{\mathcal{L}})^*(A \cap B) + (P_{\mathcal{L}})_*(B \cap \overline{A})}$$

and

$$(P_{\mathcal{L}})_*(A \mid B) = \frac{(P_{\mathcal{L}})_*(A \cap B)}{(P_{\mathcal{L}})_*(A \cap B) + (P_{\mathcal{L}})^*(B \cap \overline{A})}$$

Let $\mathcal{L}$ be a class of type 1. $\mathcal{L}$ is of type 2, iff

$$(P_{\mathcal{L}})_*(A_1 \cup \cdots \cup A_n) \geq \sum_{I:\emptyset \neq I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \cdot (P_{\mathcal{L}})_*(\bigcap_{i \in I} A_i)$$