Künstliche Intelligenz
Bayes'sche Netze

Jun.-Prof. Dr.-Ing. Stefan Lüdtke

Universität Rostock

Institut für Visual & Analytic Computing

Motivation

- Die Full Joint erlaubt es uns, jede Query zu beantworten
- Aber: Größe der Full Joint wächst exponentiell mit Anzahl der Variablen
- Können (bedingte) Unabhängigkeit ausnutzen, um Verteilung schlauer zu repräsentieren
- Bayes'sches Netz: Repräsentation einer Wahrscheinlichkeitsverteilung, in der die (Un)abhängigkeiten zwischen Variablen explizit gemacht werden

Bayesian networks

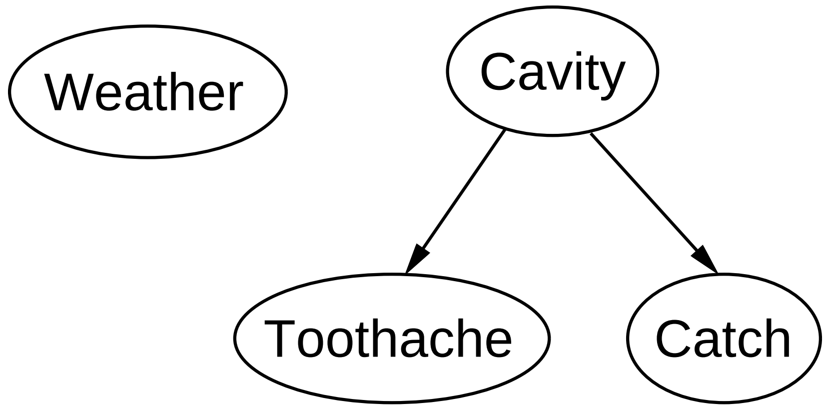
A simple, graphical notation for conditional independence assertions
and hence for compact specification of full joint distributions

Syntax:
- a set of nodes, one per variable
- a directed, acyclic graph (link $\approx$ "directly influences")
- a conditional distribution for each node given its parents:
  $\mathbf{P}(X_i|Parents(X_i))$

In the simplest case, conditional distribution represented as
a conditional probability table (CPT) giving the
distribution over X_i for each combination of parent values

Example

Topology of network encodes conditional independence assertions:



$Weather$ is independent of the other variables

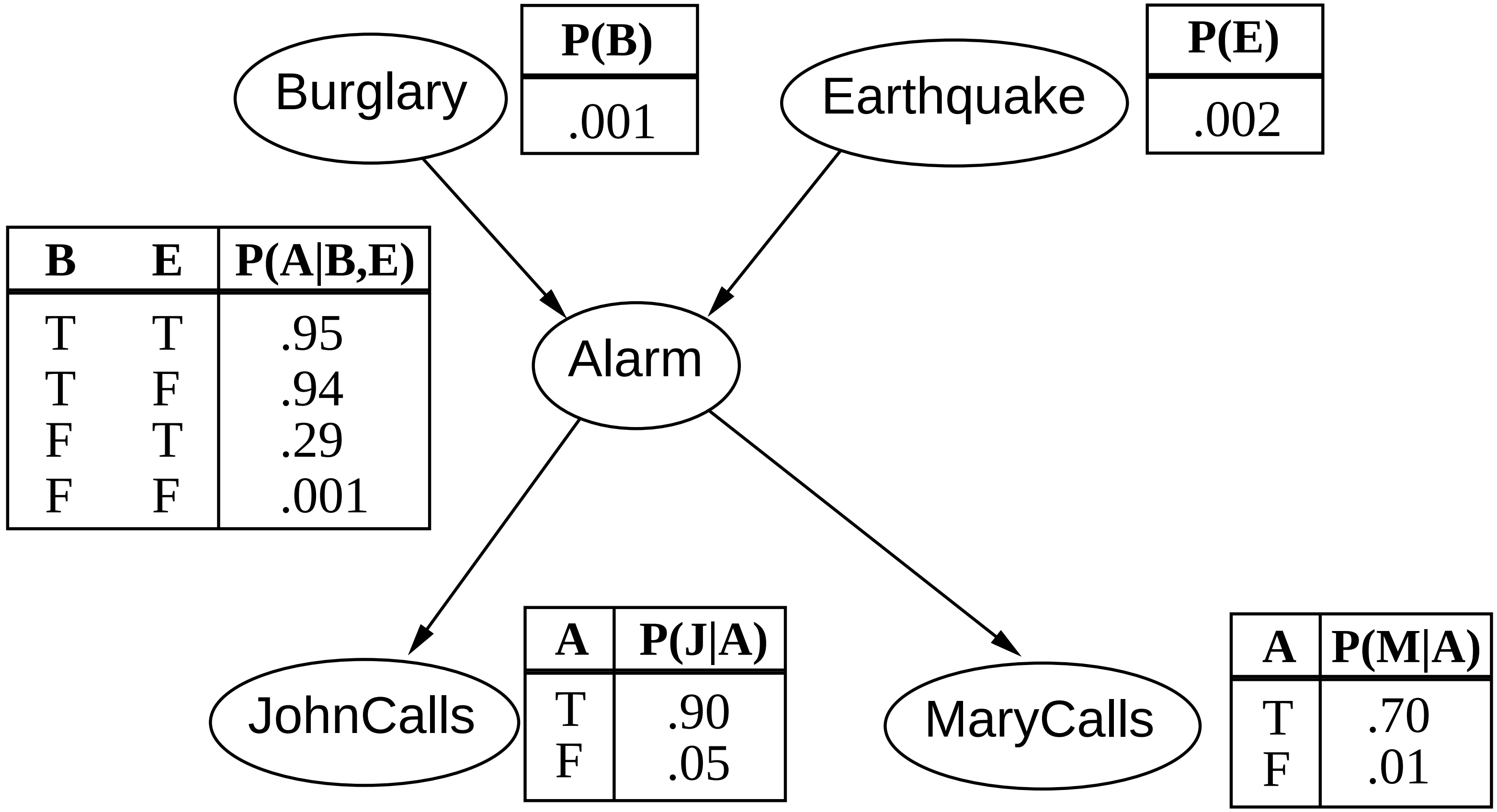
$Toothache$ and $Catch$ are conditionally independent given $Cavity$

Example

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

Variables: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$
Network topology reflects "causal" knowledge:
- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

Example contd.



P(B)
.001

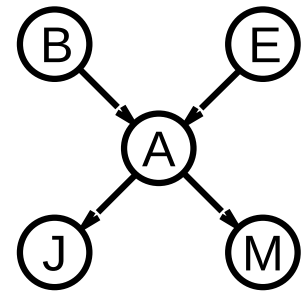
P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Compactness



A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values

Each row requires one number p for $X_i = true$
(the number for $X_i = false$ is just $1 - p$)

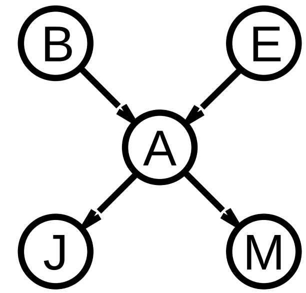
If each variable has no more than k parents,
the complete network requires $O(n \cdot 2^k)$ numbers

I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution

For burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)

Global semantics

Global semantics defines the full joint distribution
as the product of the local conditional distributions:



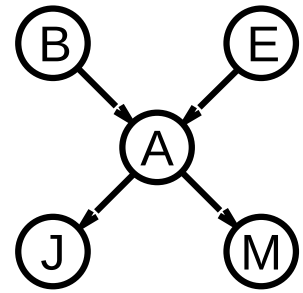
$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$$=$$

Global semantics

"Global" semantics defines the full joint distribution
as the product of the local conditional distributions:



$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

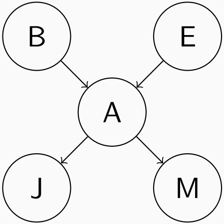
$$\begin{aligned} &= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\ &\approx 0.00063 \end{aligned}$$

Belief Netork (Example)

Recap

$P(b)$
0.01

$P(e)$
0.002



B	E	$P(a)$
$\top$	$\top$	0.95
$\top$	$\bot$	0.94
$\bot$	$\top$	0.29
$\bot$	$\bot$	0.001

A	$P(j)$
$\top$	0.90
$\bot$	0.05

A	$P(m)$
$\top$	0.70
$\bot$	0.01

What is $P(b \mid j, m)$?

Inference Tasks

Recap

Basic inference task in BNs: Compute the posterior probability of a (set of) variable(s) given some evidence.
Notation:

- Let N be a Bayesian Network over the set of variables V
- Let X be a random variable we are interested in (query variable)
- Let $\mathrm{E} = \{E_1, E_2, \ldots, E_n\}$ be a set of evidence variables
- Let e be one particular observed event, i.e., an assignment of values to E
- We call $\mathrm{Y} = \mathrm{V} \setminus (\{X\} \cup \mathrm{E})$ the set of hidden (non-query) variables

Basic inference task:
Compute $P(X \mid \mathrm{e})$ wrt. the dependencies defined by N

Exact Inference by Enumeration

- From the laws of conditional probabilities we know:

$$P(X \mid \mathrm{e}) = \frac{P(X, \mathrm{e})}{P(\mathrm{e})}$$

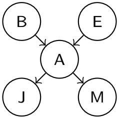
- Using the laws of probability, we find:

$$P(X \mid \mathrm{e}) = \frac{\sum_{\mathrm{y}} P(X, \mathrm{e}, \mathrm{y})}{P(\mathrm{e})}$$

  with y being an assignment to all variables in Y.
- I.e., we can compute $P(X \mid \mathrm{e})$ by summing over all possible assignments of the hidden variables.

Compute $P(b \mid j, m)$ by Enumeration

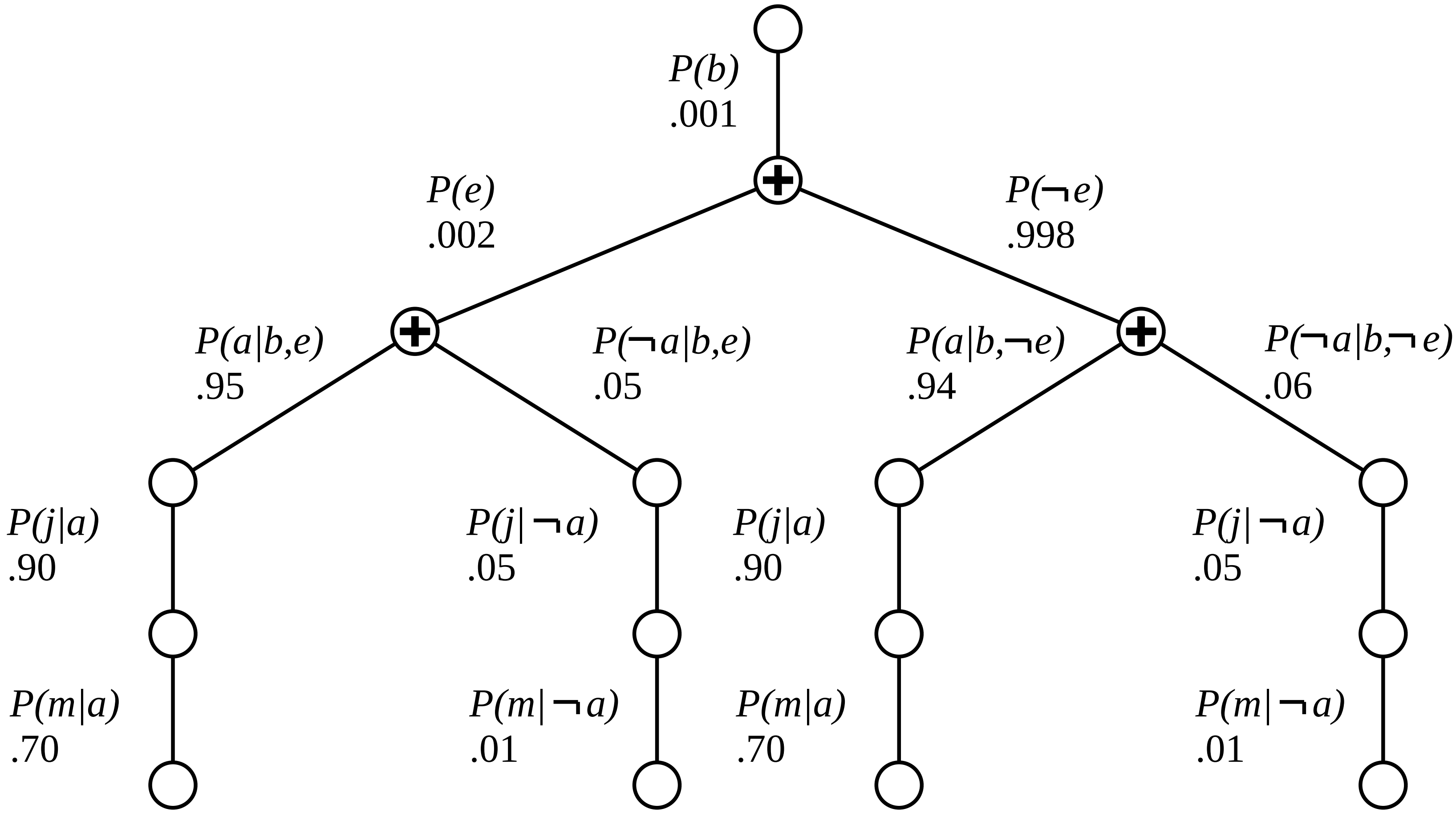
- From
  - the definitions for conditional probabilities, the chain rule, and
  - independence assumptions of the domain, we get:



$$
\begin{aligned}
P(b \mid j, m) &= \frac{P(b, j, m)}{P(j, m)} = c \cdot P(b, j, m) \\
&= c \cdot \sum_{e' \in \mathrm{dom}_E} \sum_{a' \in \mathrm{dom}_A} P(b, j, m, e', a') \\
&= c \cdot \sum_{e' \in \mathrm{dom}_E} \sum_{a' \in \mathrm{dom}_A} P(b)P(e')P(a' \mid b, e')P(j \mid a')P(m \mid a') \\
&= c \cdot P(b) \sum_{e' \in \mathrm{dom}_E} P(e') \sum_{a' \in \mathrm{dom}_A} P(a' \mid b, e')P(j \mid a')P(m \mid a')
\end{aligned}
$$

- Notation:
  - lower case letters represent assigned random variables, i.e., b is the short-hand notation for $B = \top$
  - "primed" lower case letters represent variables over which sums are computed, e' only occurs within the $\sum_{e' \in \mathrm{dom}_E}$

Evaluation tree



Enumeration is inefficient: repeated computation
e.g., computes $P(j|a)P(m|a)$ for each value of e

Complexity of the Inference by Enumeration

- In the worst case, we have to sum out almost all the variables
- For n Boolean variables, the complexity is in the order of $O(n2^n)$

What is $P(b \mid j, m)$ again?

$$\begin{aligned}
P(b \mid j, m) &= c \cdot P(b) \sum_{e' \in \mathrm{dom}_E} P(e') \sum_{a' \in \mathrm{dom}_A} P(a' \mid b, e') P(j \mid a') P(m \mid a') \\
&= c \cdot P(b) \sum_{e' \in \mathrm{dom}_E} P(e') \big(P(a \mid b, e') P(j \mid a) P(m \mid a) + \\
&\qquad\qquad P(\neg a \mid b, e') P(j \mid \neg a) P(m \mid \neg a) \big) \\
&= c \cdot P(b) \Big(P(e) \big(P(a \mid b, e) P(j \mid a) P(m \mid a) + \\
&\qquad\qquad P(\neg a \mid b, e) P(j \mid \neg a) P(m \mid \neg a) \big) + \\
&\qquad P(\neg e) \big(P(a \mid b, \neg e) P(j \mid a) P(m \mid a) + \\
&\qquad\qquad P(\neg a \mid b, \neg e) P(j \mid \neg a) P(m \mid \neg a) \big) \Big)
\end{aligned}$$

Internal structure of the Formula

- There is an internal structure due to the summation over possible assignments:

$$
\begin{aligned}
P(b \mid j, m) = c \cdot P(b) \Big(P(e) \big(P(a \mid b, e) & P(j \mid a) P(m \mid a) + \\
P(\neg a \mid b, e) & P(j \mid \neg a) P(m \mid \neg a) \big) + \\
P(\neg e) \big(P(a \mid b, \neg e) & P(j \mid a) P(m \mid a) + \\
P(\neg a \mid b, \neg e) & P(j \mid \neg a) P(m \mid \neg a) \big) \Big)
\end{aligned}
$$

- Certain sub-formulae can be re-used, e.g.:
  - $P(m \mid a)$
  - $P(m \mid \neg a)$
  - $P(j \mid a)$
  - $P(j \mid \neg a)$

Complexity of the Inference by Enumeration and Caching

- In the worst case, we have to sum out almost all the variables
- But we can safe some results for re-usage

Exact Inference by Variable Elimination

- We can utilise the structure of the equation
- Sum out the variables from "right to left" and store the intermediate results

- Problem: How to store the intermediate results?
- Solution: The intermediate results are called *Factors*

Factorisation of $P(B \mid j, m)$

- The equation for the distribution over B given j and m:

$$P(B \mid j, m) = c \cdot \underbrace{\underbrace{P(B)}_{f_1(B)} \cdot \underbrace{\sum_{e' \in \mathrm{dom}_E} \underbrace{\underbrace{P(e')}_{f_2(E)} \cdot \underbrace{\sum_{a' \in \mathrm{dom}_A} \underbrace{\underbrace{P(a' \mid b, e')}_{f_3(A,B,E)} \cdot \underbrace{\underbrace{\underbrace{P(j \mid a')}_{f_4(J,A)}}_{f'_4(A) = f_4^j} \cdot \underbrace{\underbrace{P(m \mid a')}_{f_5(M,A)}}_{f'_5(A) = f_5^m}}_{f_6(A) = (f'_4 * f'_5)}}_{f_7(A,B,E) = (f_3 * f_6)}}_{f_8(B,E) = (\sum_A f_7)}}_{f_9(B,E) = (f_2 * f_8)}}_{f_{10}(B) = (\sum_E f_9)}}_{f_{11}(B) = (f_1 * f_{10})}$$

- Operations of factors:
  - Primitive sub-expressions are renamed, e.g., $P(B) = f_1(B)$
  - Values can be assigned to variables, e.g., $f'_4(A) = f_4^j = f_4^{J=\top}$
  - Factors can be multiplied, e.g., $f_6(A) = (f_4 * f_5)$
  - Variable can be summed out, e.g., $f_8(B, E) = (\sum_A f_7)$

Factors

- An n-dimensional *factor* f is a (representation of a) function from n random variables $X_1, \ldots, X_n$ to a positive real number.
  - A factor can be a probability distribution (summing up to 1)
  - but it does not need to be
- Notation for factor f over $X_1, \ldots, X_j$: $f(X_1, \ldots, X_j)$
- A simple example of a factor over three binary random variables

$f(X, Y, Z)$:

X	Y	Z	*val*
$\top$	$\top$	$\top$	0.1
$\top$	$\top$	$\bot$	0.9
$\top$	$\bot$	$\top$	0.2
$\top$	$\bot$	$\bot$	0.8
$\bot$	$\top$	$\top$	0.4
$\bot$	$\top$	$\bot$	0.6
$\bot$	$\bot$	$\top$	0.3
$\bot$	$\bot$	$\bot$	0.7

Operations on Factors: Assignment of Values to Variables

We can assign some or all of the variables of a factor:

- $f(X_1{=}v_1, X_2, \ldots, X_j)$, where $v_1 \in dom(X_1)$:
  is a factor over $X_2, \ldots, X_j$.
- $f(X_1{=}v_1, X_2{=}v_2, \ldots, X_j{=}v_j)$ is a number,
  it is the value of f when each X_i has value v_i.
- Notation for $f(X_1{=}v_1, X_2, \ldots, X_j)$: $f(X_1, X_2, \ldots, X_j)^{X_1=v_1}$

Operations on Factors: Assignment of Values to Variables (Example)

f :

X	Y	Z	val
$\top$	$\top$	$\top$	0.1
$\top$	$\top$	$\bot$	0.9
$\top$	$\bot$	$\top$	0.2
$\top$	$\bot$	$\bot$	0.8
$\bot$	$\top$	$\top$	0.4
$\bot$	$\top$	$\bot$	0.6
$\bot$	$\bot$	$\top$	0.3
$\bot$	$\bot$	$\bot$	0.7

$f^{X=t}$:

Y	Z	val
$\top$	$\top$	0.1
$\top$	$\bot$	0.9
$\bot$	$\top$	0.2
$\bot$	$\bot$	0.8

$f^{X=t,Z=f}$:

Y	val
$\top$	0.9
$\bot$	0.8

$$f^{X=t,Y=f,Z=f} = 0.8$$

Operations on Factors: Product of Two Factors

The *product* of factor $f_1(X, Y)$ and $f_2(Y, Z)$, where Y are the variables in common, is the factor $(f_1 * f_2)(X, Y, Z)$ defined by:

$$(f_1 * f_2)(X, Y, Z) = f_1(X, Y) f_2(Y, Z).$$

Operations on Factors: Product of Two Factors (Example)

f_1:

A	B	val
$\top$	$\top$	0.1
$\top$	$\bot$	0.9
$\bot$	$\top$	0.2
$\bot$	$\bot$	0.8

f_2:

B	C	val
$\top$	$\top$	0.3
$\top$	$\bot$	0.7
$\bot$	$\top$	0.6
$\bot$	$\bot$	0.4

$(f_1 * f_2)$:

A	B	C	val
$\top$	$\top$	$\top$	0.03
$\top$	$\top$	$\bot$	0.07
$\top$	$\bot$	$\top$	0.54
$\top$	$\bot$	$\bot$	0.36
$\bot$	$\top$	$\top$	0.06
$\bot$	$\top$	$\bot$	0.14
$\bot$	$\bot$	$\top$	0.48
$\bot$	$\bot$	$\bot$	0.32

Operations on Factors: Summing out a Variable from a Factor

We can *sum out* a variable, say X_1 with domain $\{v_1, \ldots, v_k\}$, from factor $f(X_1, \ldots, X_j)$. This results in a factor on $X_2, \ldots, X_j$, defined as follows:

$$
\begin{aligned}
(\textstyle\sum_{X_1} f)(X_2, \ldots, X_j) &= f(X_1, \ldots, X_j)^{X_1 = v_1} + \cdots + f(X_1, \ldots, X_j)^{X_1 = v_k} \\
&= \sum_{v \in \mathrm{dom}(X_1)} f(X_1, X_2, \ldots, X_j)^{X_1 = v}
\end{aligned}
$$

Operations on Factors: Summing out a Variable from a Factor (Example)

f_3:

A	B	C	val
$\top$	$\top$	$\top$	0.03
$\top$	$\top$	$\bot$	0.07
$\top$	$\bot$	$\top$	0.54
$\top$	$\bot$	$\bot$	0.36
$\bot$	$\top$	$\top$	0.06
$\bot$	$\top$	$\bot$	0.14
$\bot$	$\bot$	$\top$	0.48
$\bot$	$\bot$	$\bot$	0.32

$(\sum_B f_3)$:

A	C	val
$\top$	$\top$	0.57
$\top$	$\bot$	0.43
$\bot$	$\top$	0.54
$\bot$	$\bot$	0.46

Exact Inference by Variable Elimination
Factorisation of $P(B \mid j, m)$

$$P(B \mid j,m) = c \cdot \underbrace{\underbrace{P(B)}_{f_1(B)} \cdot \underbrace{\sum_{e' \in \mathrm{dom}_E} \underbrace{\underbrace{P(e')}_{f_2(E)} \cdot \underbrace{\sum_{a' \in \mathrm{dom}_A} \underbrace{\underbrace{P(a' \mid b, e')}_{f_3(A,B,E)} \cdot \underbrace{\underbrace{\underbrace{P(j \mid a')}_{f_4(J,A)}}_{f_4'(A)=f_4^j} \cdot \underbrace{\underbrace{P(m \mid a')}_{f_5(M,A)}}_{f_5'(A)=f_5^m}}_{f_6(A)=(f_4' * f_5')}}_{f_7(A,B,E)=(f_3 * f_6)}}_{f_8(B,E)=(\sum_A f_7)}}_{f_9(B,E)=(f_2 * f_8)}}_{f_{10}(B)=(\sum_E f_9)}}_{f_{11}(B)=(f_1 * f_{10})}$$

- Assign the value $M = \top$ in $f_5'(A) = f_5^{M=\top}$
- Compute $f_6(A) = (f_4' * f_5')$
- Sum out A and compute $f_8(B, E) = (\sum_A f_7)$

Zusammenfassung

- Bayes'sche Netze sind natürliche Repräsentation für Verteilungen mit bedingten Unabhängigkeiten
- Topologie + CPTs= Kompakte Repräsentation der Full Joint
- Einfach zu spezifizieren, auch für Nichtexperten
- Exakte Inferenz durch Aufzählen nur in einfachen Fällen möglich
- Variable Elimination ist ein effizienterer exakter Inferenzalgorithmus
- Noch effizienter: Approximative Inferenzalgorithmen (hier nicht behandelt)