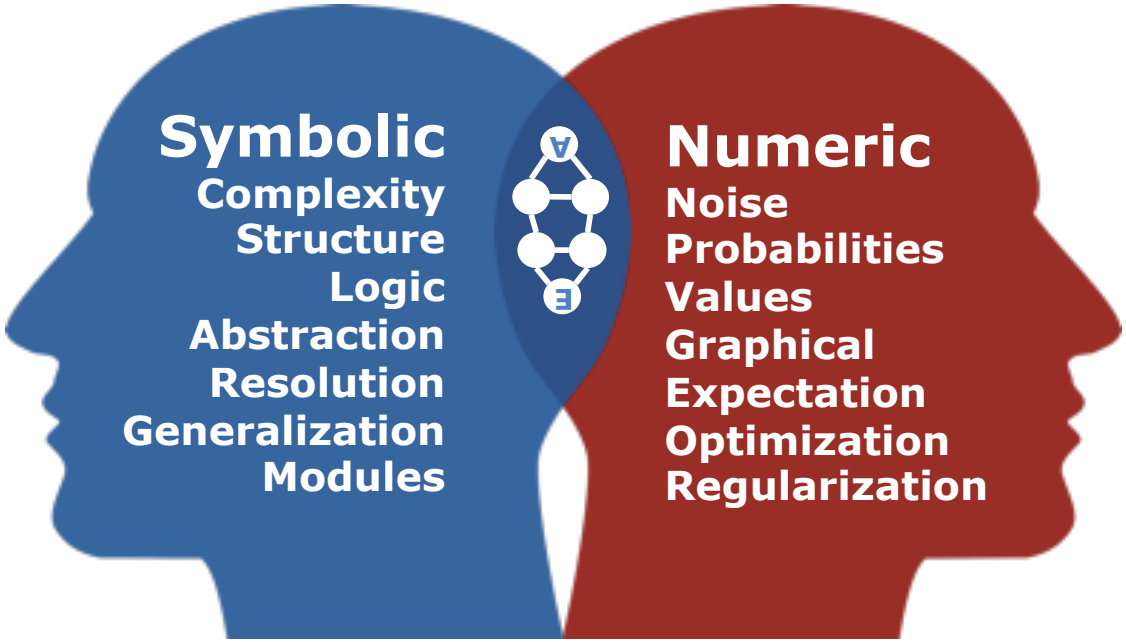
TECHNISCHE UNIVERSITÄT DARMSTADT



Statistical Relational AI

Graphical Models – Representation, Inference, Learning

Kristian Kersting

Thanks to Vincent Conitzer, Rina Dechter, Luc De Raedt, Pedro Domingos, Peter Flach, Dieter Fensel, Florian Ficher, Vibhav Gogate, Carlos Guestrin, Daphne Koller, Nir Friedman, Ray Mooney, Sriraam Natarajan, David Poole, Fabrizio Riguzzi, Dan Suciu, Guy van den Broeck, and many others for making their slides publically available

Goals

- What are Bayesian networks?
- Joint and conditional distribution
- (Conditional) Independence
- Local Markov Assumption
- D-separation
- Variable Elimination
- Expectation-Maximization
- Local search for model selection

St. Paul's Cathredal, London, UK

Let's start on Bayesian Networks

TECHNISCHE UNIVERSITÄT DARMSTADT

- One of the most exciting recent advancements in statistical AI

Judea Pearl won the ACM Turing Award 2012 for his fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning

- Compact representation for exponentially-large probability distributions
- Fast marginalization
- Exploit conditional independencies

Representing Distributions by Enumeration

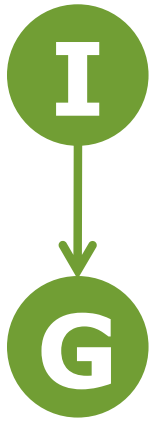
- Consider $P(X_i)$
  - Assign a probability to each $x_i \in \mathrm{Val}(X_i)$
  - Number of parameters assuming $|\mathrm{Val}(X_i)| = k$
    ? $k-1$

- Now, consider $P(X_1,\ldots,X_n)$
  - How many parameters assuming $|\mathrm{Val}(X_i)| = k$ **?**
  - Same thing, $k^n - 1$

- **Bayesian networks will often need fewer parameters. What is the trick?**

Conditional Parameterization (2 nodes)

- **Grade is influenced by Intelligence**
- Make use of **chain rule**



$$P(G,I) = P(G \mid I) \cdot P(I)$$

$$P(I = VH, G = B)$$

$$= P(I = VH) \cdot P(G = B \mid I = VH)$$

$$= 0.85 \cdot 0.1$$

$$= 0.085$$

	VH	H
P(I)	0.85	0.15

	I=	VH	H
P(G|I)	A	0.9	0.5
	B	0.1	0.5

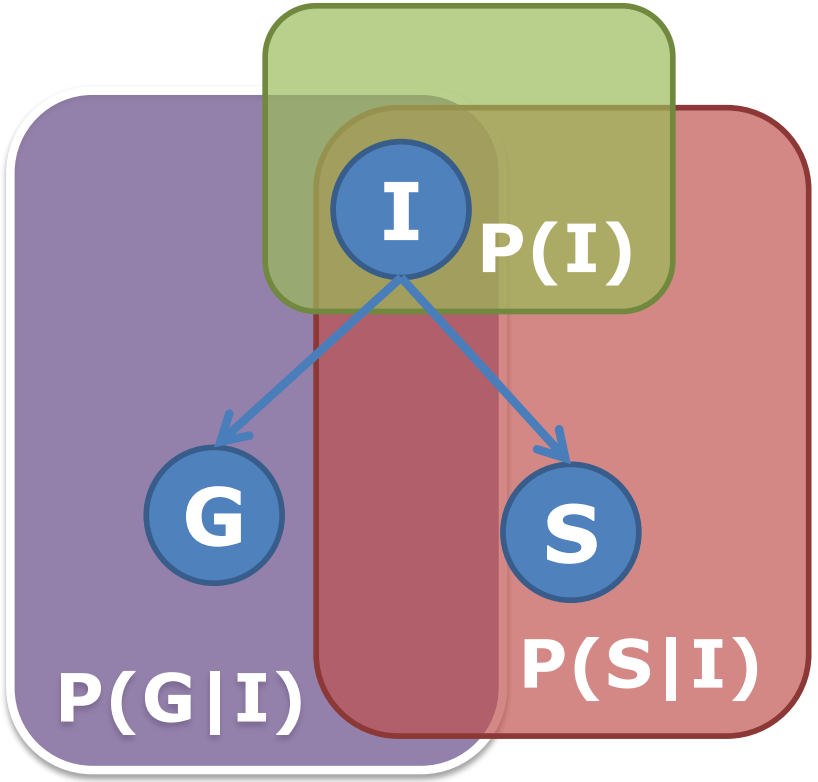
Represent conditional distributions graphically

Same # of parameters as by enumeration

Conditional Parameterization (3 nodes) and what if variables are independent?

- **Grade and SAT score are influenced by Intelligence**
- **but** $\left(G \perp S \middle| I\right)$ **, i.e.,** $P(G \mid S, I) = P(G \mid I)$

$$
\begin{aligned}
& P\left(G, S, I\right) \\
&= P(G, S \mid I) \cdot P(I) \\
&= P(G \mid S, I) \cdot P(S \mid I) \cdot P(I) \\
&= P(G \mid I) \cdot P(S \mid I) \cdot P(I)
\end{aligned}
$$



Indpendence can lead to smaller # of parameters as by enumeration

Can we even get a linear complexity?

TECHNISCHE UNIVERSITÄT DARMSTADT

- $\left(X_i \perp X_j\right), \forall i, j$ is not enough
- We must assume that $\left(\mathbf{X} \perp \mathbf{Y}\right), \forall \mathbf{X}, \mathbf{Y}$ subsets of $\left\{X_1, \ldots, X_n\right\}$

Let X1 and X2 be drawn from Bernoulli(0.5) and X3 = X1 xor X2. Now, P(Xi,Xj) = 1/4 = P(Xi)P(Xj) (use a table to show this). Since X3 depends deterministically on X1 and X2 it cannot be the case that X1,X2 are independent of X3.

- Now, we can write $P\left(X_1, \ldots, X_n\right) = \prod_{i=1}^{n} P\left(X_i\right)$
- How many parameters?

$$n \cdot (k-1) = O(n)$$

The Naïve Bayes Model

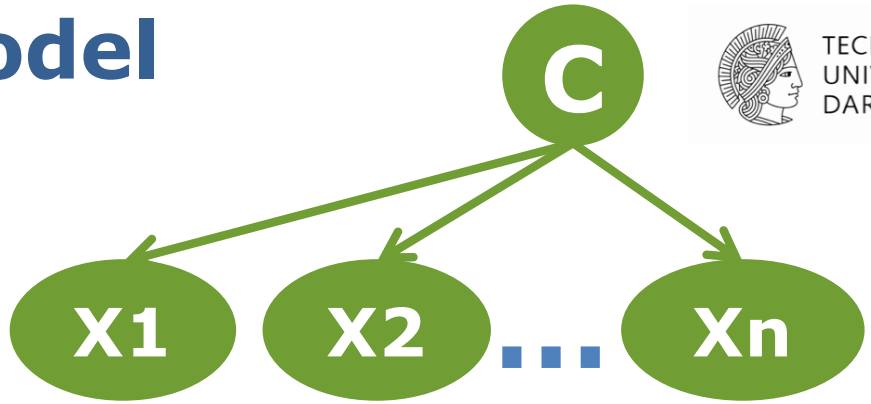
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Now, your first real Bayesian network !**

- **Class variable:** C
- **Evidence variables:** $\{X_1,\ldots,X_n\}$
- **Assume** that $(\mathbf{X} \perp \mathbf{Y} \mid C), \forall \mathbf{X}, \mathbf{Y}$ subsets of $\{X_1,\ldots,X_n\}$

$$P(X_1,\ldots,X_n,C) = P(C)\cdot\prod_{i=1}^{n} P(X_i \mid C)$$

The Naïve Bayes Model



$$P(X_1,\ldots,X_n,C)$$
$$= P(C)\cdot P(X_1\,|\,C)\cdot P(X_2\,|\,X_1,C)\cdot\ldots\cdot P(X_n\,|\,X_1,\ldots X_{n-1},C)$$

So far, no assumptions. Now due to $(\mathbf{X}\perp\mathbf{Y}\,|\,C),\forall\mathbf{X},\mathbf{Y}$

it becomes the Naïve Bayes model

$$P(X_2\,|\,X_1,C)=P(X_2\,|\,C) \qquad P(X_n\,|\,X_1,\ldots X_{n-1},C)=P(X_n\,|\,C)$$

$$(X_1\perp X_2\,|\,C) \qquad (X_n\perp X_1X_2\ldots X_{n-1}\,|\,C)$$

SPAMCOP

TECHNISCHE UNIVERSITÄT DARMSTADT

- Uses a Naïve Bayes classifier
- M is spam if P(Spam|M) > P(NonSpam|M)
- Method
  - Tokenize message using Porter Stemmer
  - Estimate P(W|C) using m-estimate (a form of smoothing)
  - Remove words that do not satisfy certain conditions
  - **Train: 160 spams, 466 non-spams**
  - **Test: 277 spams, 346 non-spams**
- Results: ERROR RATE of 4.33%
  - Worse results using trigrams

So far

TECHNISCHE UNIVERSITÄT DARMSTADT

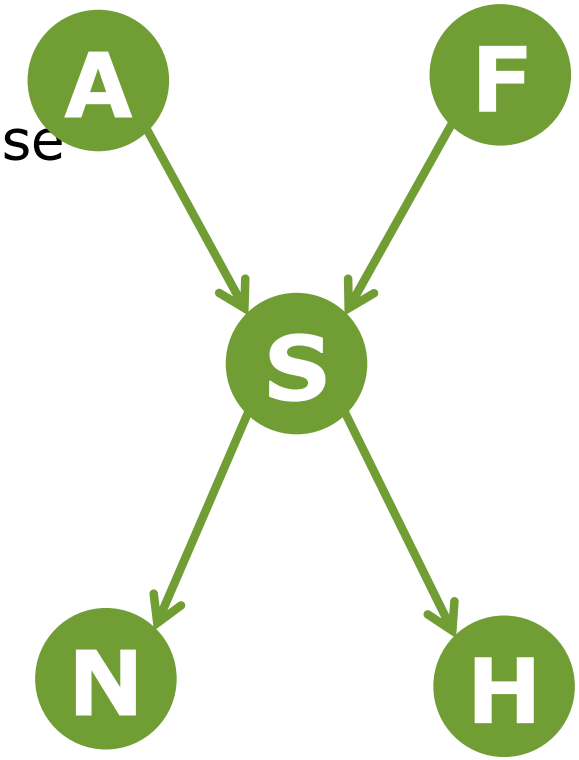
- We've heard of Bayes nets
- We've even seen how to use them for classifying spam

- Now, we'll learn
  - the semantics of BNs and
  - **relate them to independence assumptions encoded by the graph**
  - **Here we stopped**

„Causal“ Structure

TECHNISCHE UNIVERSITÄT DARMSTADT

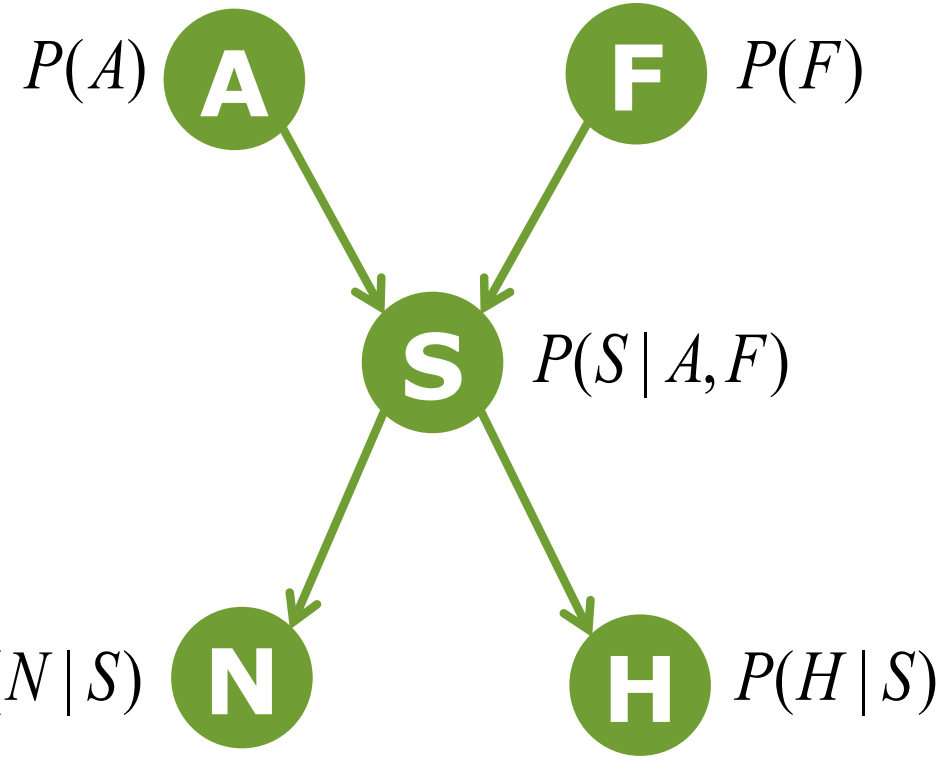
- Suppose we know the following:
  - The flu causes sinus inflammation
  - Allergies cause sinus inflammation
  - Sinus inflammation causes a runny nose
  - Sinus inflammation causes headaches
- How are these connected?



Factored Joint Distributions

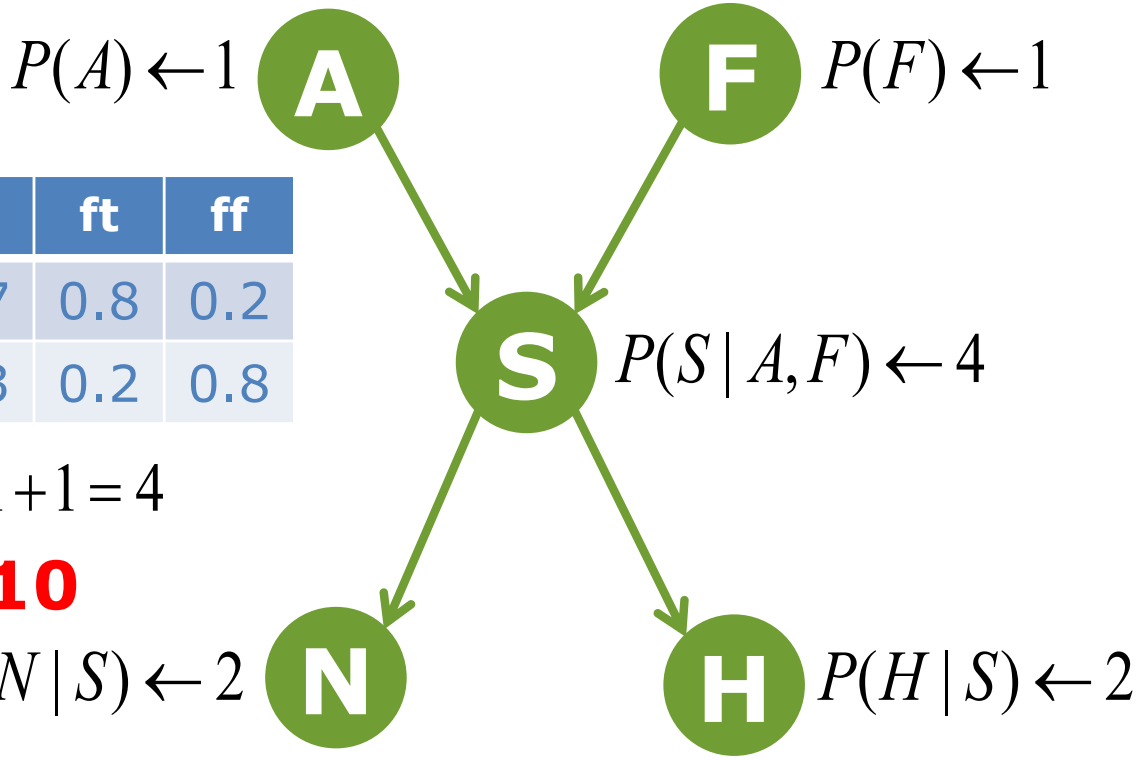
- Intuition: Nodes = Variables, Edges = Influences

$$
\begin{aligned}
P(A,F,S,H,N) &= P(A) \\
&\cdot P(F) \\
&\cdot P(S \mid A,F) \\
&\cdot P(N \mid S) \\
&\cdot P(H \mid S)
\end{aligned}
$$



Number of Parameters

- **Sparse structure**: compact representation for exponentially-large probability distributions
- Enumerative representation of binary variables: **2^5-1=31 parameters**



	A,F	tt	tf	ft	ff
S	t	0.9	0.7	0.8	0.2
	f	0.1	0.3	0.2	0.8

$$1+1+1+1=4$$

- **Using BNs: only 10 parameters**

Key: Independence Assumptions

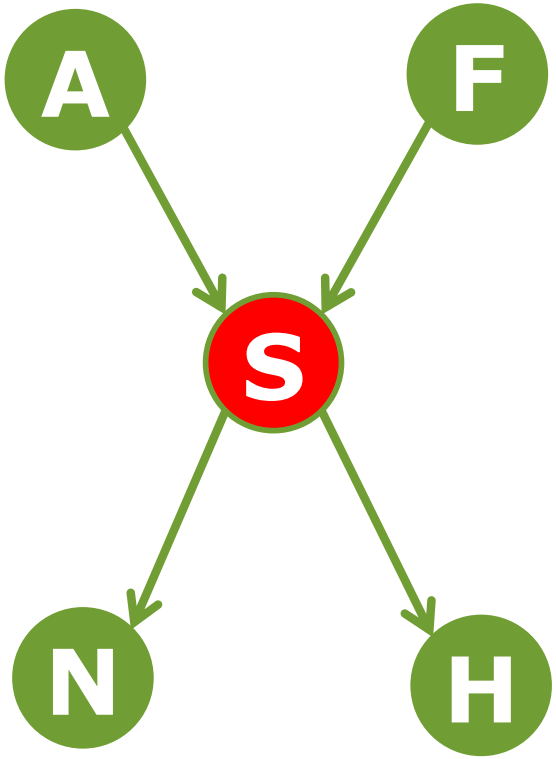
TECHNISCHE UNIVERSITÄT DARMSTADT

$$\neg(F \perp H)$$

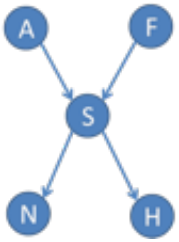
$$(F \perp H \mid S)$$

$$\neg(N \perp A)$$

$$(N \perp A \mid S)$$



Knowing sinus separates symptoms from causes



Recap: (Marginal) Independence

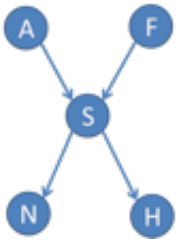
- Flu and Allergy are marginally independent

$$(A \perp F) \qquad P(A,F) = P(A) \cdot P(F)$$

A	t	f
t	0.3	0.7

F	t	f
t	0.1	0.9

A,F	t	f
t	0.3*0.1=0.03	0.3*0.9=0.27
f	0.7*0.1=0.07	0.7*0.9=0.63



Recap: (Conditional) Independence

- Flu and Headache are (not) marginally independent $\neg(F \perp H)$ $P(F,H) \neq P(F) \cdot P(H)$
- Flu and headache are independent given Sinus infection $(F \perp H \mid S)$ $P(F,H \mid S) = P(F \mid S) \cdot P(H \mid S)$

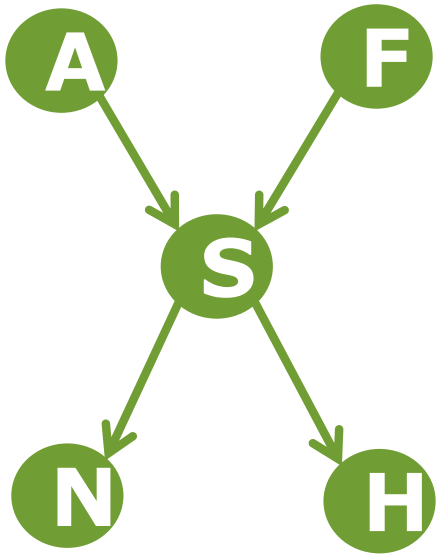
$$P(F \mid H,S) = P(F \mid S)$$

- Generally: $(X_1 \perp X_2 \ldots X_n \mid C)$ iff

$$P(X_1, X_2, \ldots, X_n \mid C) = P(X_1 \mid C) \cdot P(X_2, \ldots, X_n \mid C)$$
$$P(X_1 \mid X_2, \ldots, X_n, C) = P(X_1 \mid C)$$

Local Markov Assumption
(Second most important slide!)

TECHNISCHE UNIVERSITÄT DARMSTADT



A variable X is independent of its non-descendants given its parents and only its parents
($Xi \perp \text{NonDescendants}_{Xi} \mid \text{Pa}_{Xi}$)

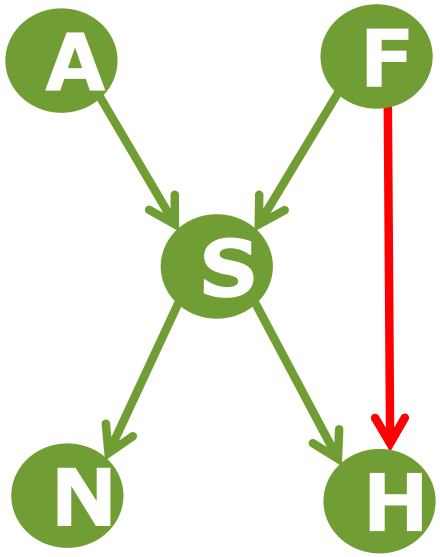
$$\text{Pa}_F = \varnothing$$
$$\text{NonDescendants}_F = \{A\}$$
$$(F \perp A)$$

$$\text{Pa}_S = \{F, A\}$$
$$\text{NonDescendants}_S = \varnothing$$
$$(S \perp ?? \mid F, A)$$

NO ASSUMPTIONS

$$\text{Pa}_N = \{S\}$$
$$\text{NonDescendants}_N = \{F, A, H\}$$
$$(N \perp \{F, A, H\} \mid S)$$

Local Markov Assumption



A variable X is independent of its non-descendants given its parents and only its parents
($Xi \perp \text{NonDescendants}_{Xi} \mid \text{Pa}_{Xi}$)

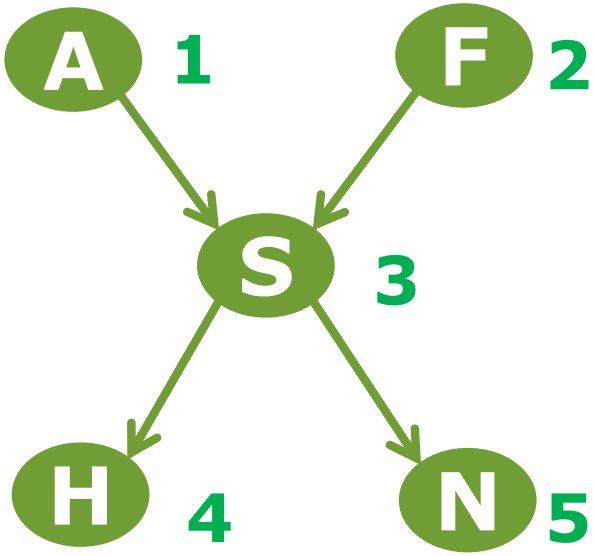
before edge included

$$\text{Pa}_H = \{S\}$$
$$\text{NonDescendants}_H = \{A, F, N\}$$
$$(H \perp \{A, F, N\} \mid S)$$

after edge included

$$\text{Pa}_H = \{F, S\}$$
$$\text{NonDescendants}_S = \{A, N\}$$
$$(H \perp \{A, N\} \mid F, S)$$

Joint Distribution



Consider **topological orders**

Now, to interpret a BN

1. Choose particular **chain rule order**
2. Apply independence assumption

$$P(A,F,S,H,N) = P(F)P(A)P(S \mid F,A)P(H \mid S)P(N \mid S)$$

$$P(A,F,S,H,N) = P(F)P(A \mid F)P(S \mid F,A)P(H \mid S,F,A)P(N \mid S,F,A,H)$$

$A \perp F$ $P(A)$ $P(S \mid F,A)$ $H \perp \{F,A,N\} \mid S$ $H \perp \{F,A\} \mid S$ $P(H \mid S)$ $N \perp \{F,A,H\} \mid S$ $P(N \mid S)$

We can decompose due to the local Markov assumtion

A General Bayesian Network (Most important slide!)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Set of random variables $\{X_1,\ldots,X_n\}$
- Directed acyclic graph
  - loops are ok but **no directed cycles**
- CPT with each Xi: $P\left(X_i \mid \mathrm{Pa}\left(X_i\right)\right)$
- Joint distribution $P\left(X_1,\ldots,X_n\right)=\prod_{i=1}^{n} P\left(X_i \mid \mathrm{Pa}\left(X_i\right)\right)$

- **Local Makov assumption**

A variable X is independent of its non-descendants given its parents and only ist parents: ($Xi \perp NonDescendants_{Xi} \mid Pa_{Xi}$)

OK, so we know now Bayesian networks

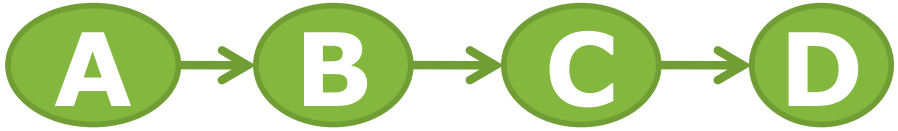
Let's turn towards inference …

… but first let's clarify one of the most important modelling tools of Bayesian networks:

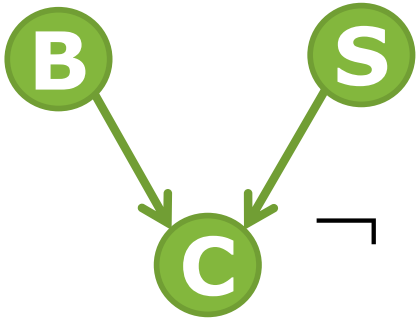
d-separation

Independencies encoded in BN

- We said: all you need is the local Markov assumption **(Xi ⊥ NonDescendants$_{Xi}$ | Pa$_{Xi}$)**
- But then we can also talk about other (in)dependencies (such as explaining away)



$$A \perp D \mid C$$



$$B \perp S$$

$$\neg\ B \perp S \mid C$$

- **So, what are the independencies encoded by a BN?**
  - Only assumption is local Markov but many other independencies can be derived using the algebra of conditional independencies!

Understanding independencies in BNs (with 3 nodes)

Local Markov Assumption: A variable X is independent of ist non-descendants given its parents and only ist parents: $(Xi \perp NonDescendants_{Xi} \mid Pa_{Xi})$

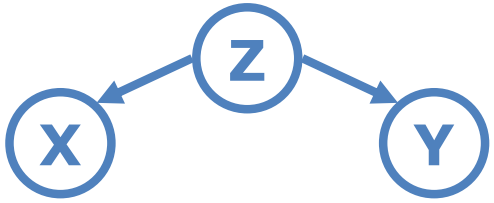
Indirect causal effect:

$$X \perp Y \mid Z$$
$$\neg X \perp Y$$

Indirect evidential effect:

$$X \perp Y \mid Z$$
$$\neg X \perp Y$$

Common cause:

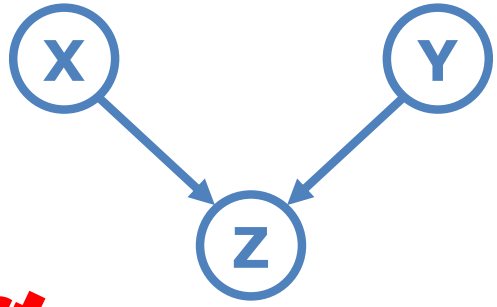


$$X \perp Y \mid Z$$
$$\neg X \perp Y$$

Represent all the same distributions

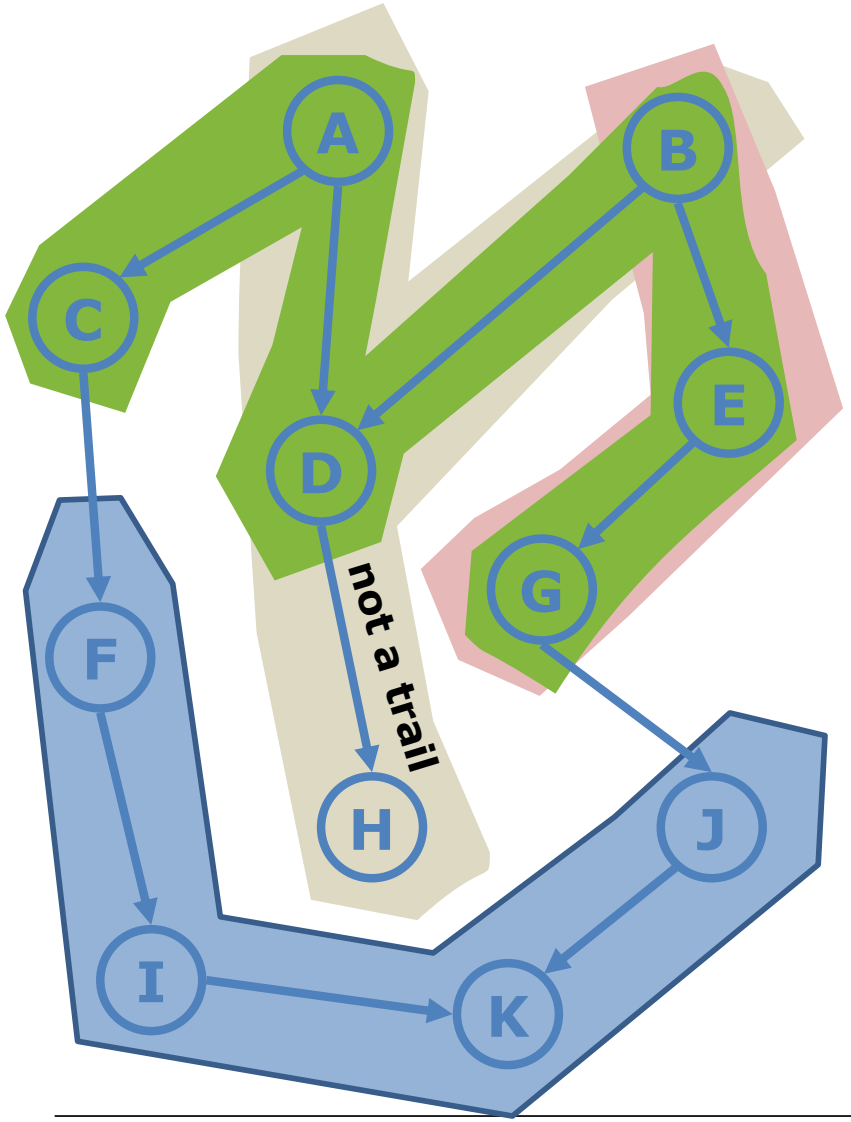
inverted

(v-structure) Common effect:



$$X \perp Y$$
$$\neg X \perp Y \mid Z$$

This can be generalized using the notion of active trails



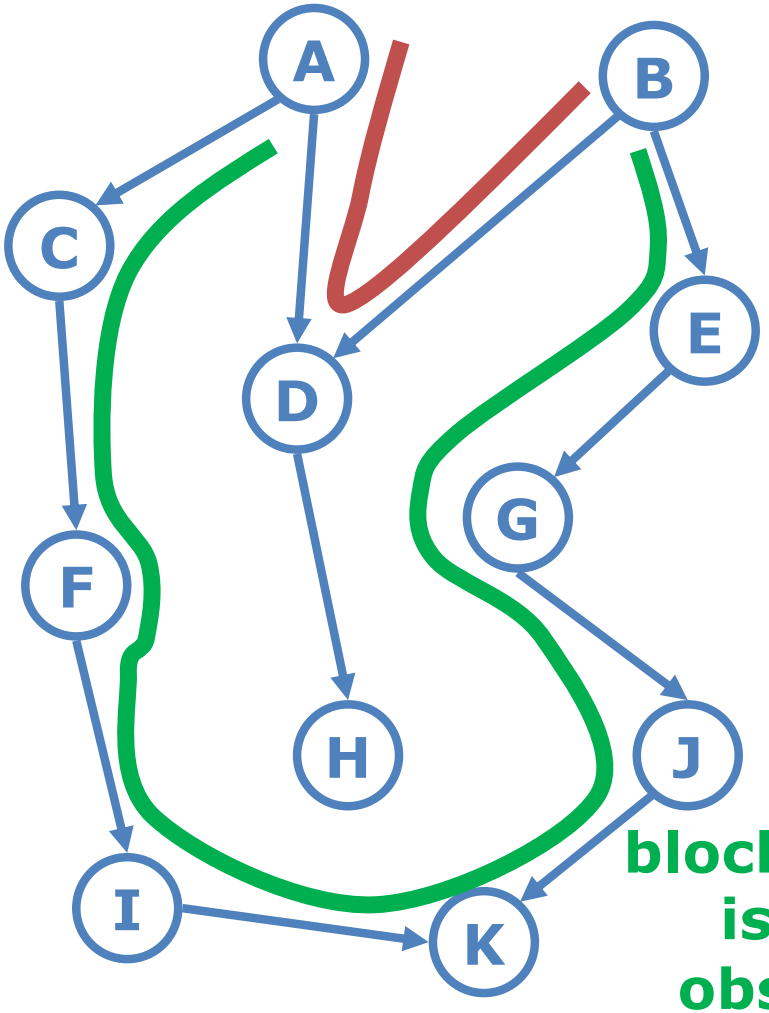
A trail is an undirected path that never visits a node twice

X and **Y** are independent given **Z** if all trails between **X** and **Y** are NOT active w.r.t. to **Z**

- A trail $X'_1 - X'_2 - \cdots - X'_k$ is **active** (when some variables $\boldsymbol{O} \subseteq \{X_1, \ldots, X_m\}$ are observed) **if** for each consecutive triplet in the trail it holds:
  - $X_{i-1} \rightarrow X_i \rightarrow X_{i+1}$, and X_i is **not observed** ($X_i \notin \boldsymbol{O}$)
  - $X_{i-1} \leftarrow X_i \leftarrow X_{i+1}$, and X_i is **not observed** ($X_i \notin \boldsymbol{O}$)
  - $X_{i-1} \leftarrow X_i \rightarrow X_{i+1}$, and X_i is **not observed** ($X_i \notin \boldsymbol{O}$)
  - $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, and X_i **is observed** ($X_i \in \boldsymbol{O}$), or **one of its descendents (v-structure)**

Intuitively, information flows a long active trials!

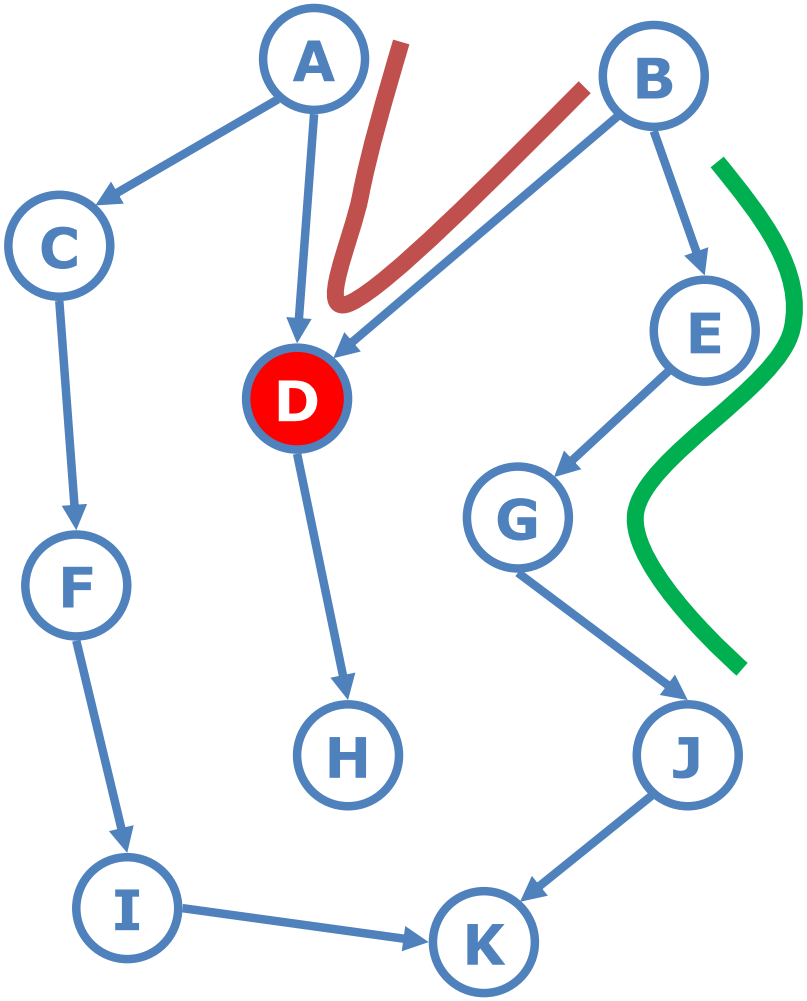
Active and Blocked Trails in BNs – Some Examples –



blocked as D and H are NOT observed

blocked as K is NOT observed

Active and Blocked Trails in BNs – Some Examples –

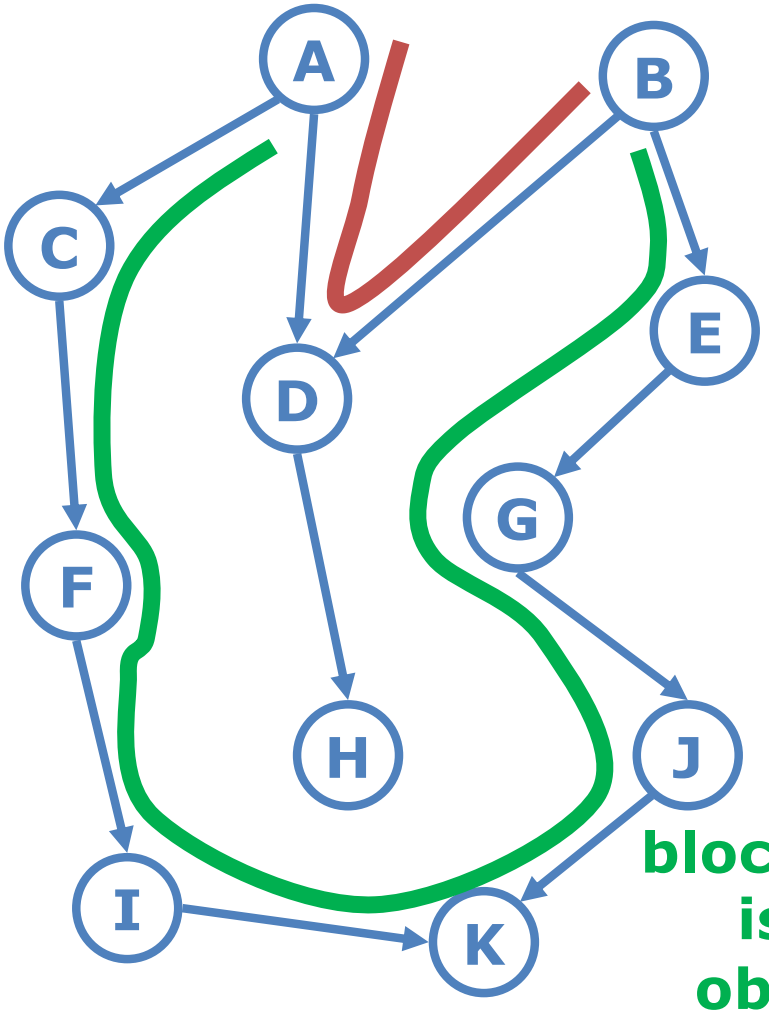


active

active

Information can flow if there is a trail x...y that is not blocked by z (active trail)

D-Separation – Some Examples

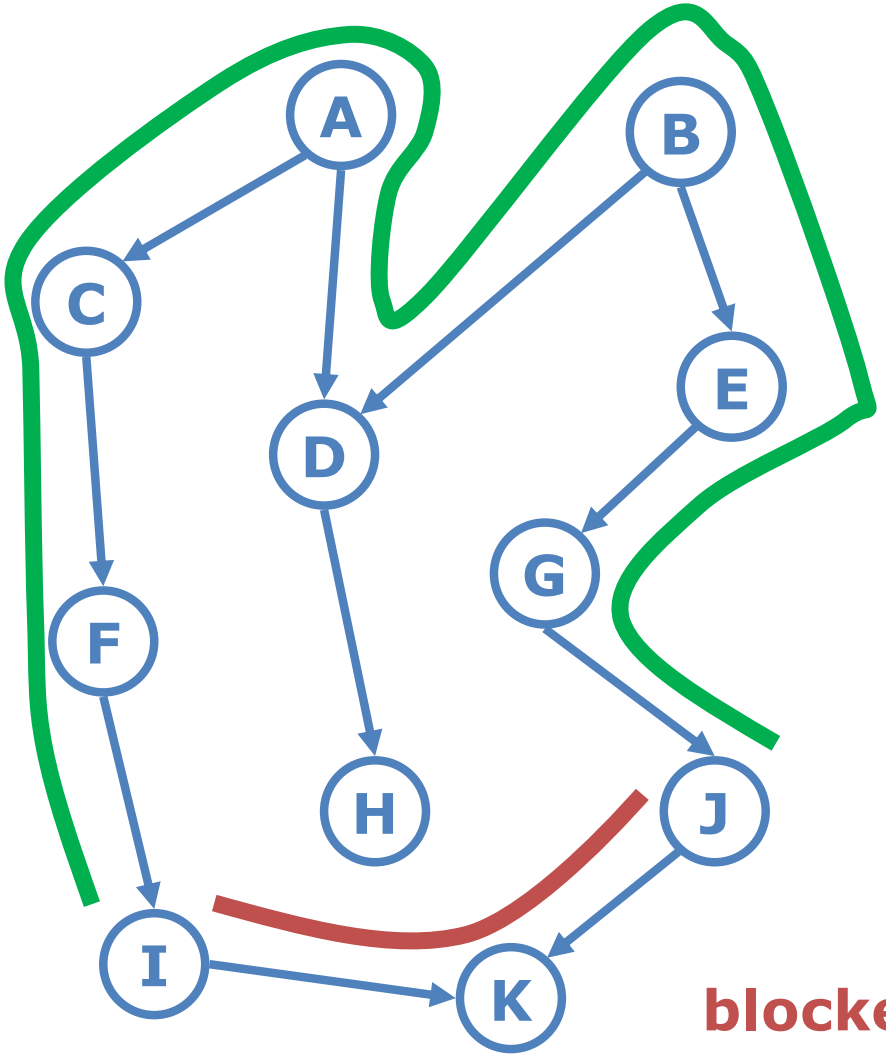


$$A \perp B$$

blocked as D and H are NOT observed

blocked as K is NOT observed

D-Separation – Some Examples

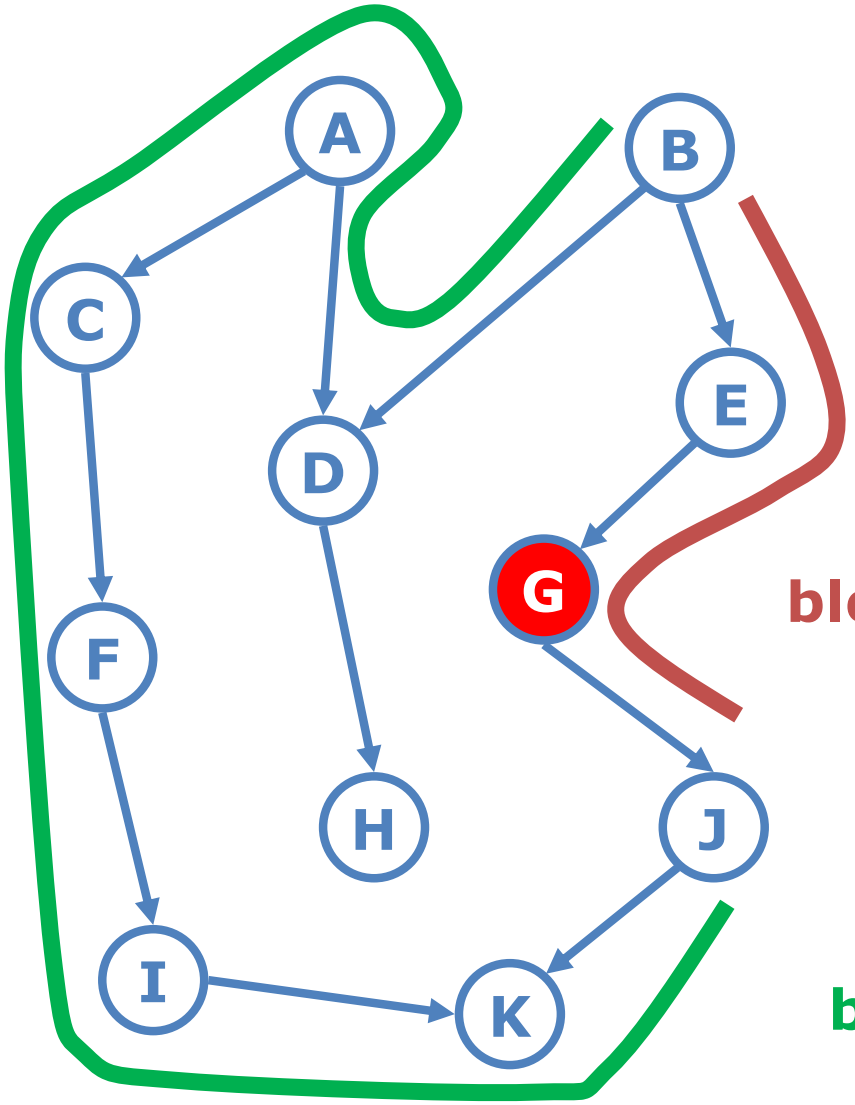


$A \perp B$ $I \perp J$

blocked

blocked

D-Separation – Some Examples



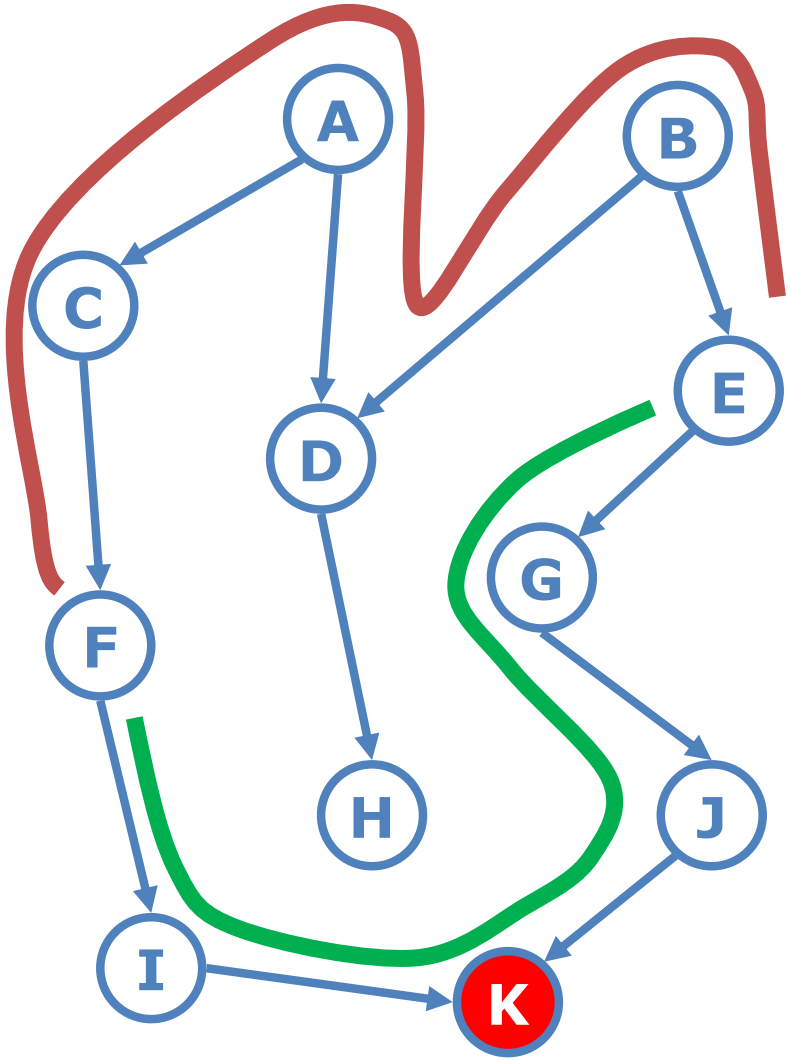
$$A \perp B \qquad I \perp J$$

$$B \perp J \mid G$$

blocked

blocked

D-Separation – Some Examples



$$A \perp B \qquad I \perp J$$

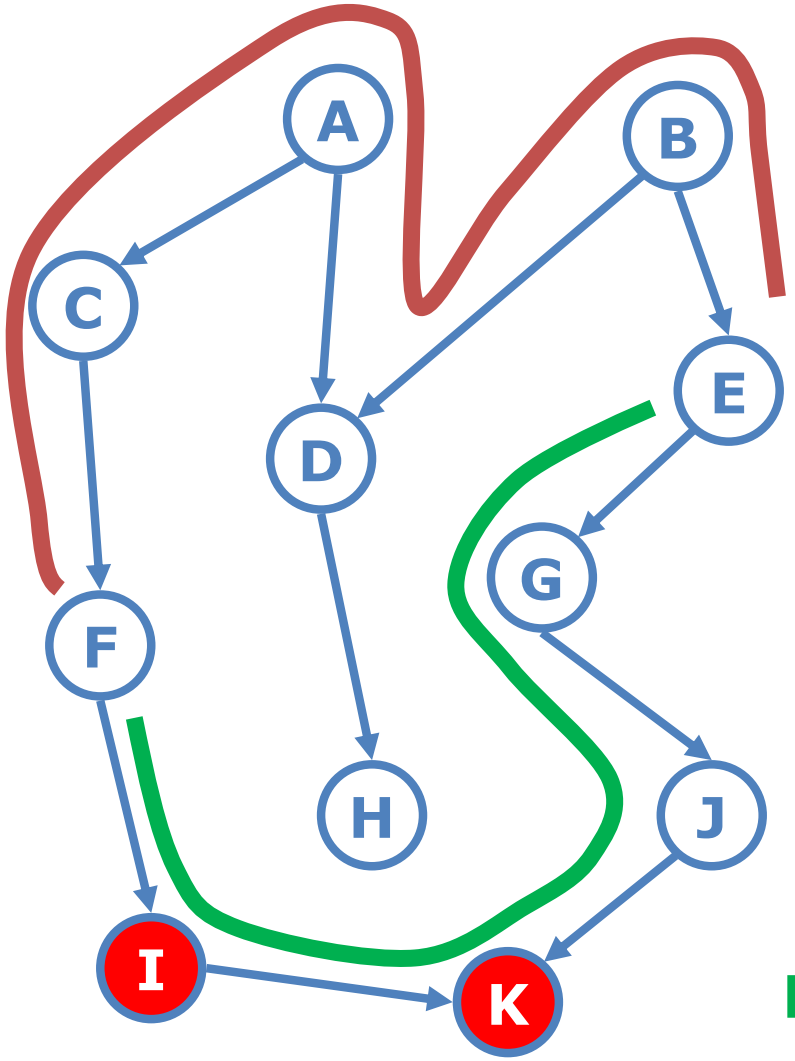
blocked

$$B \perp J \mid G$$

$$\neg E \perp F \mid K$$

Active !!!

D-Separation – Some Examples



$$A \perp B \qquad I \perp J$$

blocked

$$B \perp J \mid G$$

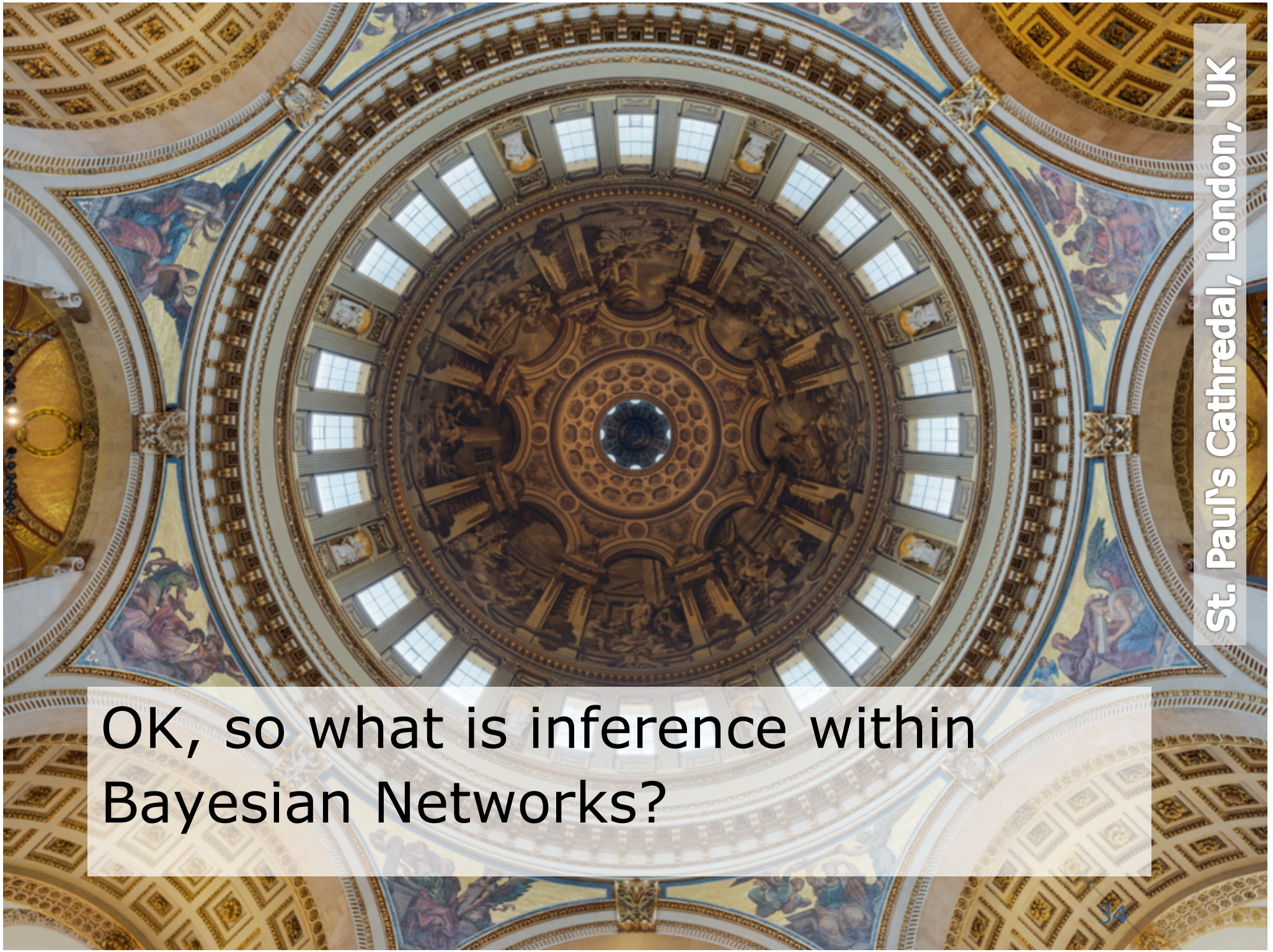
$$\neg\, E \perp F \mid K$$

$$E \perp F \mid K, I$$

blocked !!!

The difference is

$$E \perp F \mid I$$

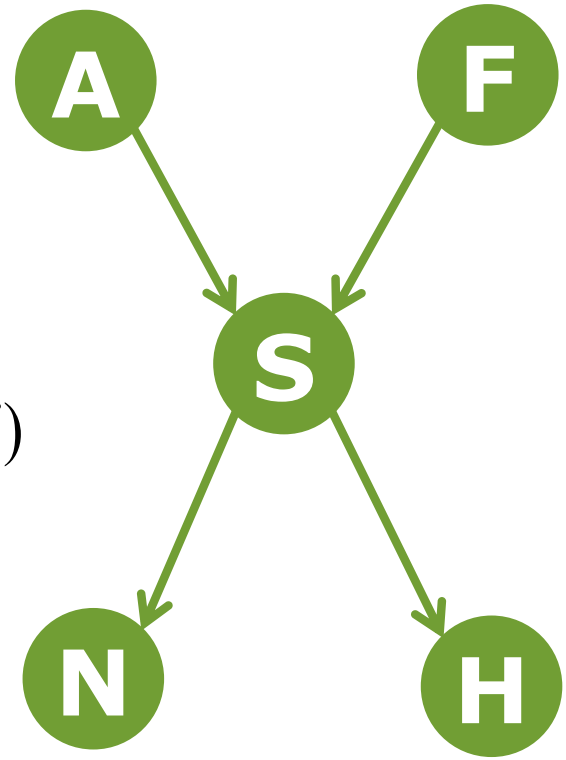


OK, so what is inference within Bayesian Networks?

Possible Queries

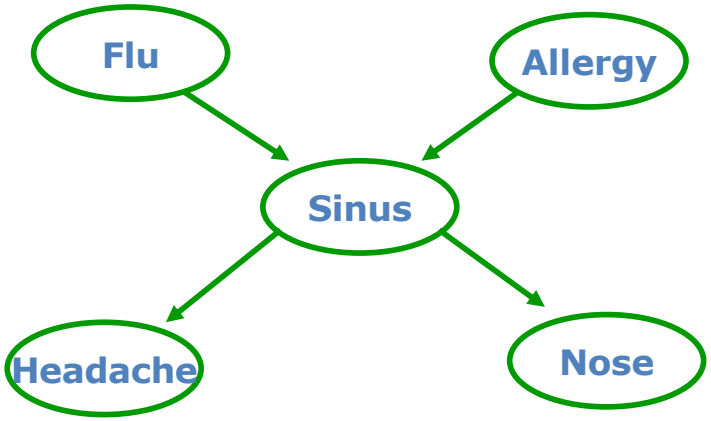
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Inference**
  - Assume H=t, N=f. What is the probability P(A=t|H=t,N=f) of an allergic reaction?
- **Most probable explanation**
  - $\max_{f,a,s} P(F=f, A=a, S=s \mid H=t, N=f)$
- **Active data collection**
  - What is next best test, i.e., variable to observe



General probabilistic inference

- Query: $P(X \mid e)$



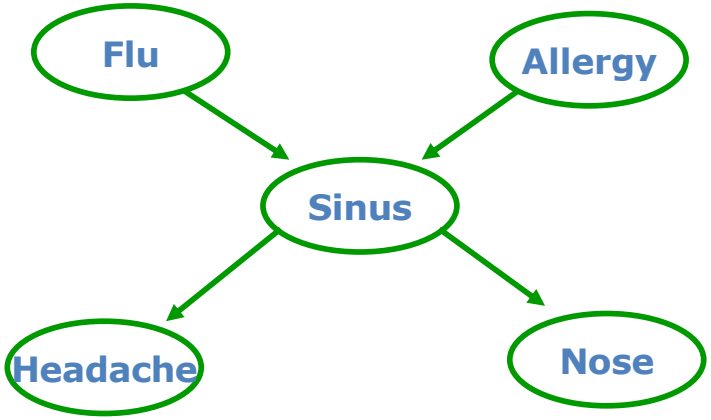
- Using def. of cond. prob.:

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

- Normalization:

$$P(X \mid e) \propto P(X, e)$$

Probabilistic inference example



$$P(A \mid N=t) \propto P(A, N=t)$$

$$= \sum_f \sum_s \sum_h P(A, f, s, h, N=t)$$

$$= \sum_f \sum_s \sum_h P(f)\, P(A)\, P(s \mid f, A)\, P(h \mid s)\, P(N=t \mid s)$$

8 terms

4 multiplies

$2^3 \leftarrow$ # eliminated vars

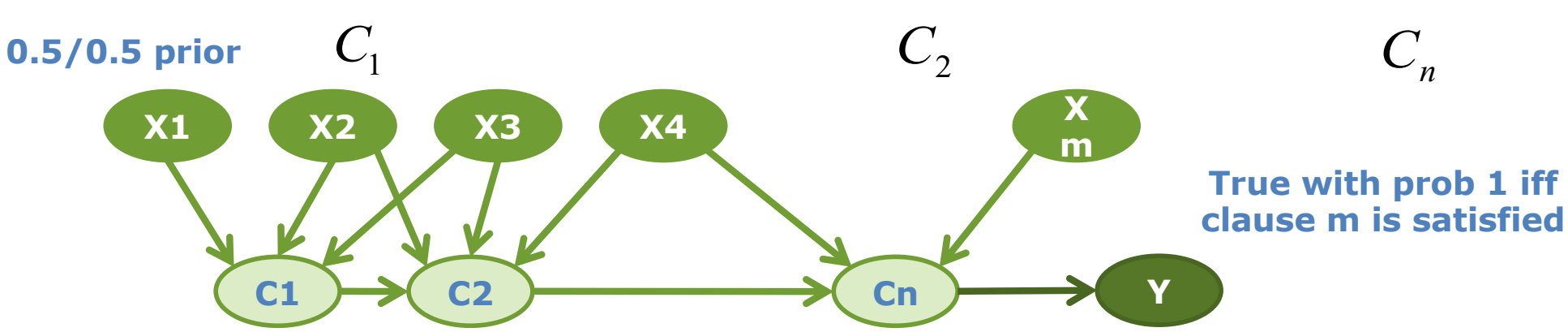
32 multiplies

Inference seems exponential in number of variables!

Complexity of conditional probability queries

- How hard is it to compute P(X|**E**=**e**)?
- Consider a reduction to 3-SAT with empty evidence E
- Does a satisfying assignment exist?

$$\underbrace{(\overline{X}_1 \vee X_2 \vee X_3)}_{C_1} \wedge \underbrace{(\overline{X}_2 \vee X_3 \vee X_4)}_{C_2} \wedge \underbrace{\ldots}_{C_n}$$

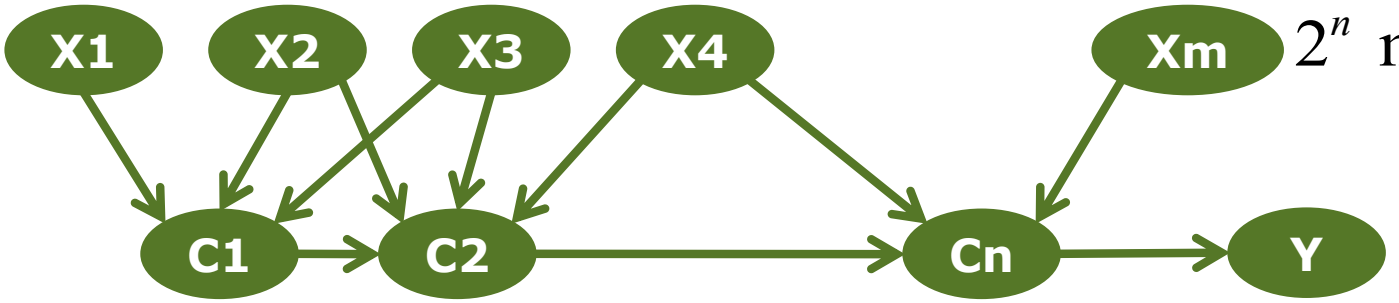


P(Y=t)>0 iff 3-SAT formula is satisfiable

Complexity of conditional probability queries

- How hard is it to compute P(X|**E**=**e**)?
  - At least NP-hard, but even harder!
  - #P problems such as counting the number of satisfiable configurations (model counting)

0.5/0.5 prior



2^n many assignments

$$p(Y=t)=\frac{\#\text{ sat assignment}}{2^n}$$

Is Inference in BNs hopeless?

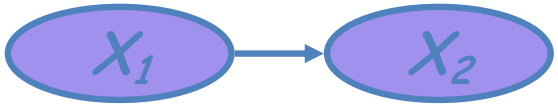
TECHNISCHE UNIVERSITÄT DARMSTADT

- In general, yes!
  - Even approximate!

Theorem:
Inference in Bayesian networks
(even approximate) is NP-hard

- In practice
  - Exploit structure
  - Many effective approximation algorithms (some with guarantees)

Inference in Simple Chains



How do we compute $P(x_2)$?

$$P(x_2) = \sum_{x_1} P(x_1, x_2) = \sum_{x_1} \underbrace{P(x_1) P(x_2 \mid x_1)}_{\text{CPDs}}$$

Inference in Simple Chains (cont.)

How do we compute $P(x_3)$?

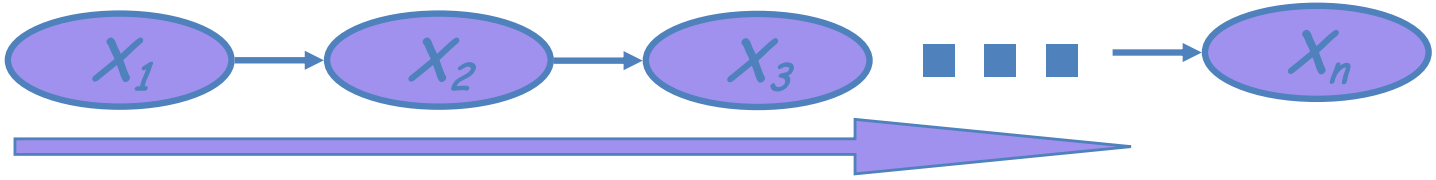
$$P(x_3) = \sum_{x_2} P(x_2, x_3) = \sum_{x_2} P(x_2) P(x_3 \mid x_2)$$

computed CPD

We already know how to compute $P(x_2)$ ...

$$P(x_2) = \sum_{x_1} P(x_1, x_2) = \sum_{x_1} P(x_1) P(x_2 \mid x_1)$$

Inference in Simple Chains (cont.)

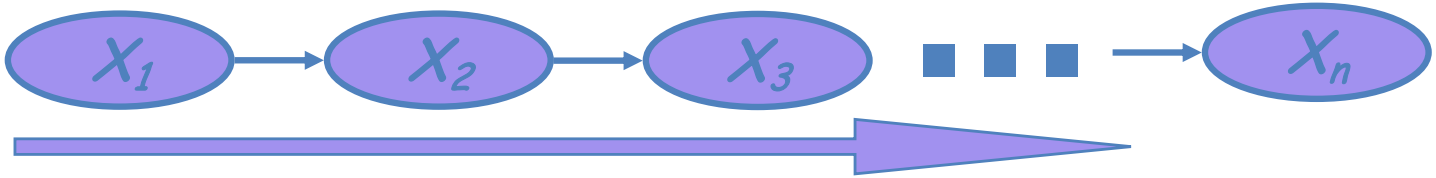


How do we compute $P(x_n)$?

Iteratively compute $P(x_1), P(x_2), P(x_3), \ldots$ **using**

$$P(x_{i+1}) = \sum_{x_i} \underbrace{P(x_i)}_{\text{computed}} \underbrace{P(x_{i+1} \mid x_i)}_{\text{CPD}}$$

Complexity of inference: Simple Chains



$$O\left(n \cdot k^2\right) \text{ v.s. exponentially in } n$$

Variable Elimination

TECHNISCHE UNIVERSITÄT DARMSTADT

General idea:

- Write query in the form

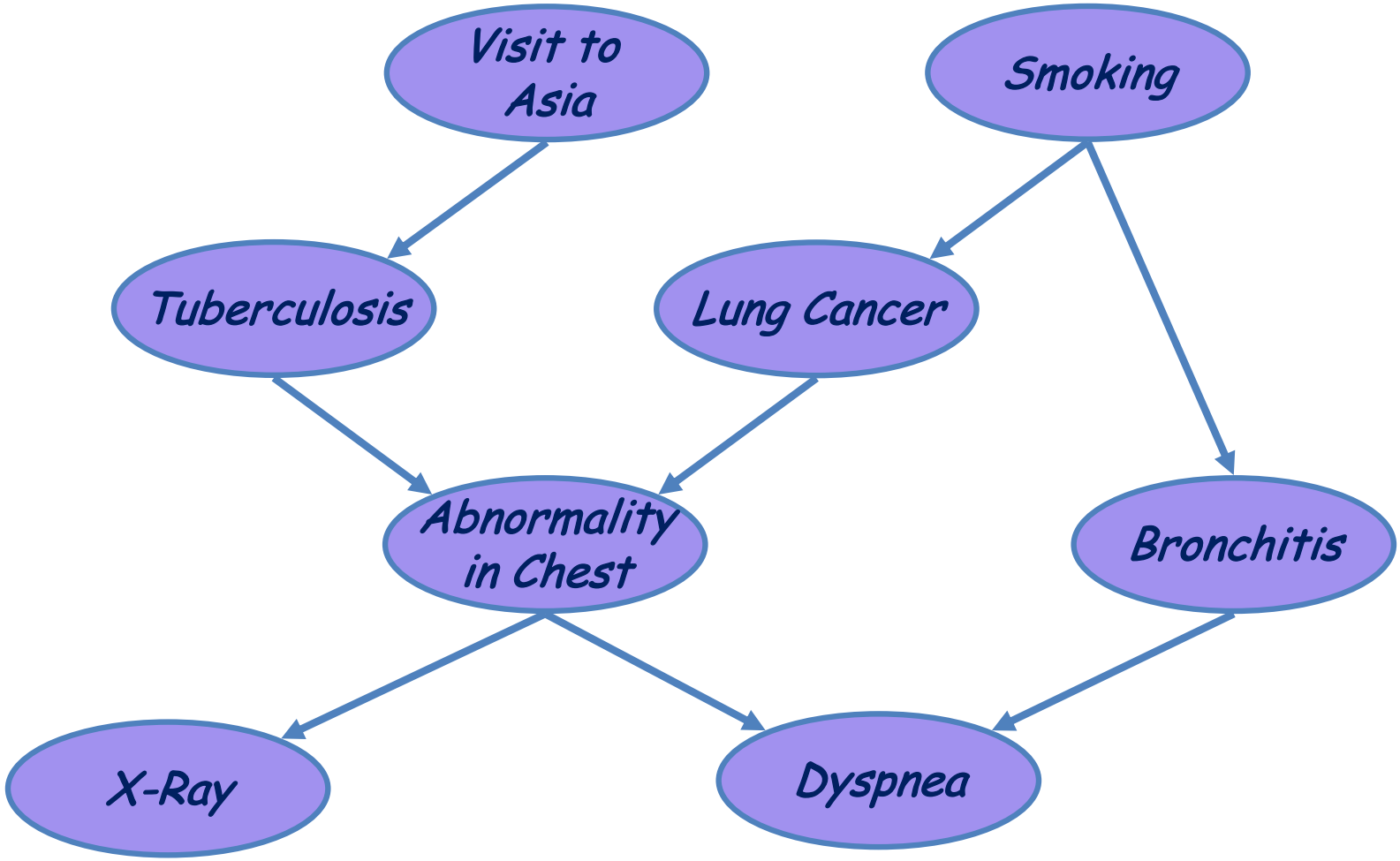
$$P(x_n, \boldsymbol{e}) = \sum_{x_k} \cdots \sum_{x_3} \sum_{x_2} \prod_i P(x_i \mid pa_i)$$

- Iteratively
  - Move all irrelevant terms outside of innermost sum
  - Perform innermost sum, getting a new term
  - Insert the new term into the product

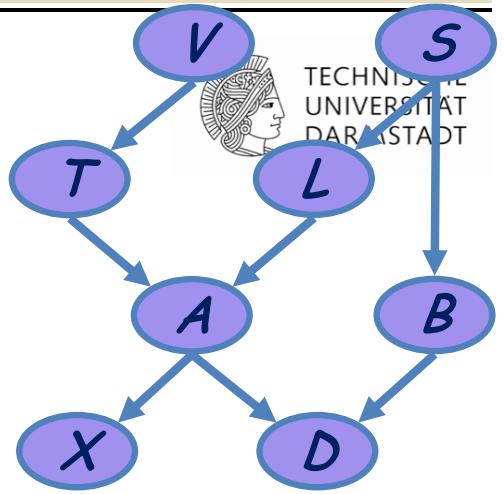
A More Complex Example

TECHNISCHE UNIVERSITÄT DARMSTADT

- "Asia" network:



- We want to compute *P(d)*
- Need to eliminate: *v,s,x,t,l,a,b*



Initial factors

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

Eliminate: *v*

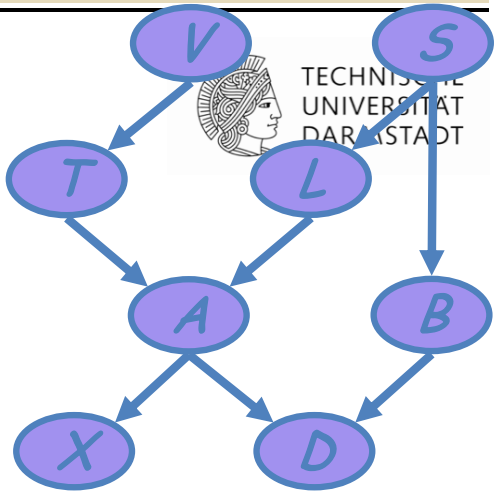
Compute: $f_v(t) = \sum_v P(v)P(t|v)$

$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

Note: $f_v(t) = P(t)$

In general, result of elimination is not necessarily a probability term

- **We want to compute** $P(d)$
- **Need to eliminate:** v,s,x,t,l,a,b



Initial factors

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

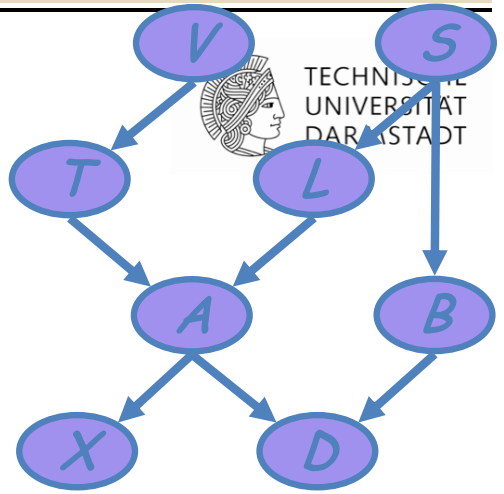
Eliminate: s

Compute: $f_s(b,l) = \sum_s P(s)P(b|s)P(l|s)$

$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$

Summing on s results in a factor with two arguments $f_s(b,l)$
In general, result of elimination may be a function of several variables

- **We want to compute** $P(d)$
- **Need to eliminate:** v,s,x,t,l,a,b



Initial factors

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$

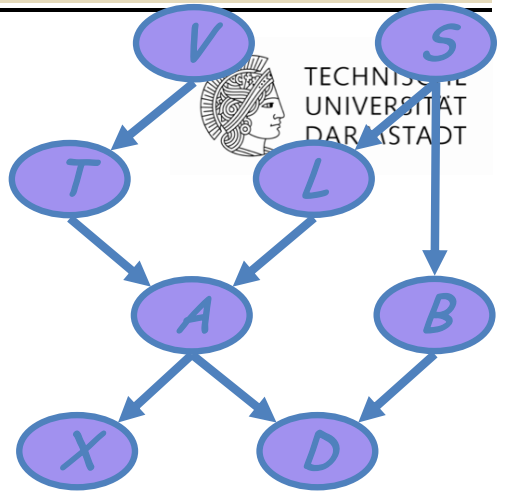
Eliminate: x

Compute: $f_x(a) = \sum_x P(x|a)$

$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a|t,l)P(d|a,b)$$

Note: $f_x(a) = 1$ **for all values of** a **!!**

- **We want to compute *P(d)***
- **Need to eliminate: *v,s,x,t,l,a,b***



Initial factors

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$

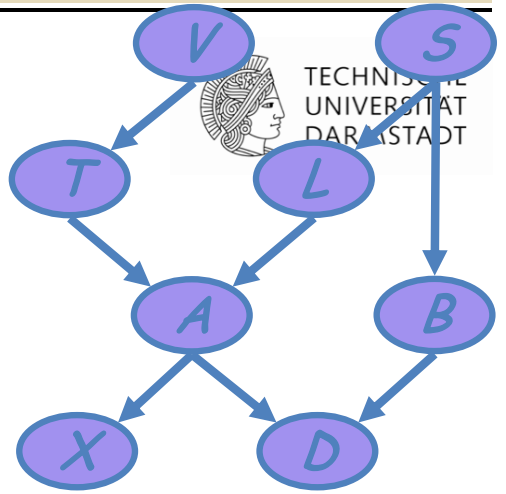
$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a|t,l)P(d|a,b)$$

Eliminate: *t*

Compute: $f_t(a,l)=\sum_t f_v(t)P(a|t,l)$

$$\Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d|a,b)$$

- **We want to compute *P(d)***
- **Need to eliminate: *v,s,x,t,l,a,b***



Initial factors

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a|t,l)P(d|a,b)$$

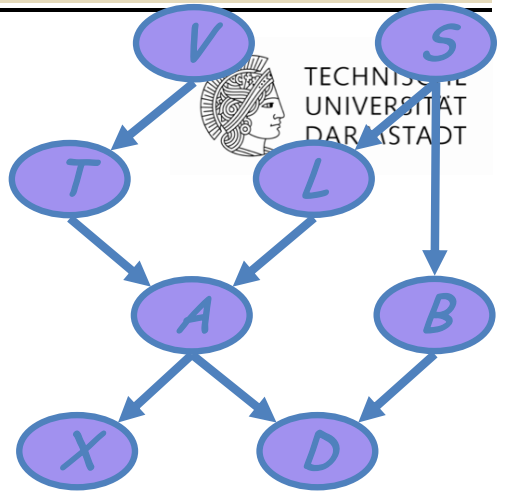
$$\Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d|a,b)$$

Eliminate: /

Compute: $f_l(a,b) = \sum_l f_s(b,l)f_t(a,l)$

$$\Rightarrow f_l(a,b)f_x(a)P(d|a,b)$$

- **We want to compute *P(d)***
- **Need to eliminate: *v,s,x,t,l,a,b***



Initial factors

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$

$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a|t,l)P(d|a,b)$$

$$\Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d|a,b)$$

$$\Rightarrow f_l(a,b)f_x(a)P(d|a,b) \Rightarrow \underline{f_a(b,d)} \Rightarrow \underline{f_b(d)}$$

Eliminate: ***a,b***

Compute: $f_a(b,d) = \sum_a f_l(a,b)f_x(a)p(d|a,b)$ $f_b(d) = \sum_b f_a(b,d)$

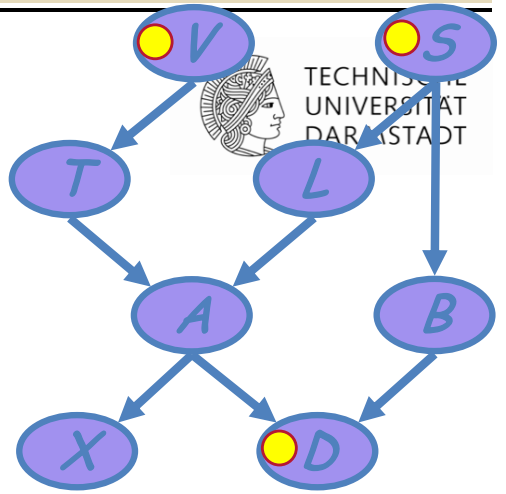
Variable Elimination

TECHNISCHE UNIVERSITÄT DARMSTADT

- We now understand variable elimination as a sequence of **rewriting** operations

- Computation depends on order of elimination

Dealing with Evidence



- How do we deal with evidence?

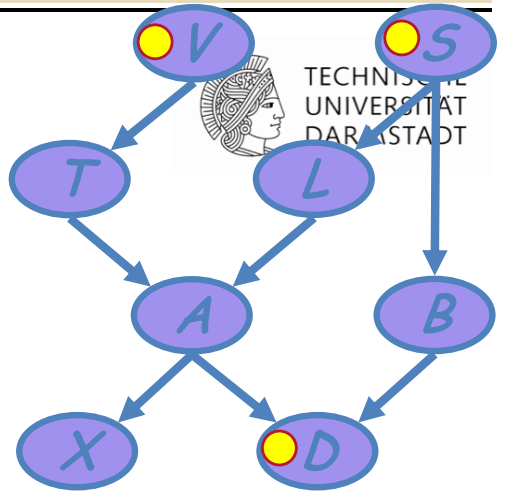
- Suppose get evidence

$$V = t,\ S = f,\ D = t$$

- We want to compute

$$P(L,\ V = t,\ S = f,\ D = t)$$

Dealing with Evidence



- We start by writing the factors:

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

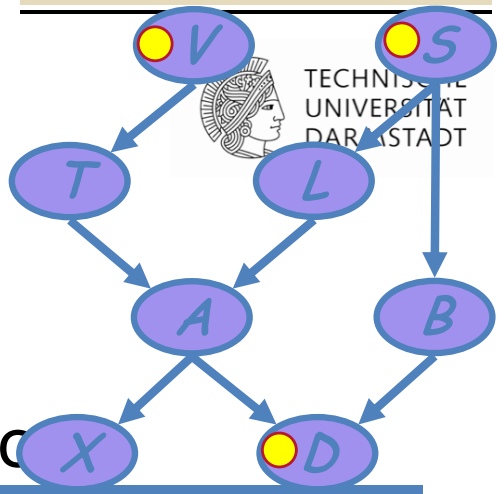
- Since we know that $V = t$, we don't need to eliminate V
- Instead, we can replace the factors $P(V)$ and $P(T|V)$ with

$$f_{P(V)} = P(V = t) \quad f_{p(T|V)}(T) = P(T|V = t)$$

- This "selects" the appropriate parts of the original factors given the evidence
- Note that $f_{p(V)}$ is a constant, and thus does not appear in elimination of other variables

- Given evidence $V = t, S = f, D = t$
- Compute $P(L, V = t, S = f, D = t)$

Dealing with Evidence

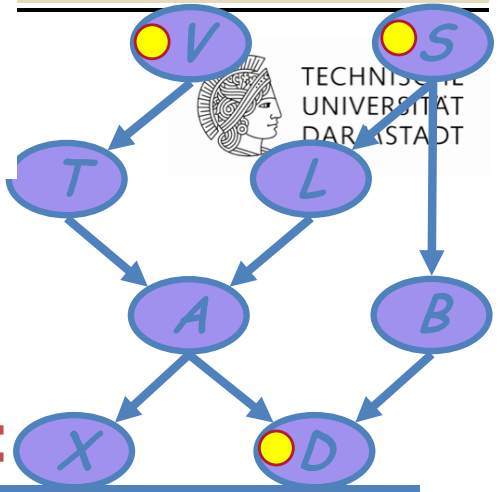


- Initial factors, after setting evidenc

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) P(x \mid a) f_{P(d|a,b)}(a,b)$$

- Given evidence $V = t, S = f, D = t$
- Compute $P(L, V = t, S = f, D = t)$

Dealing with Evidence



- Initial factors, after setting evidence:

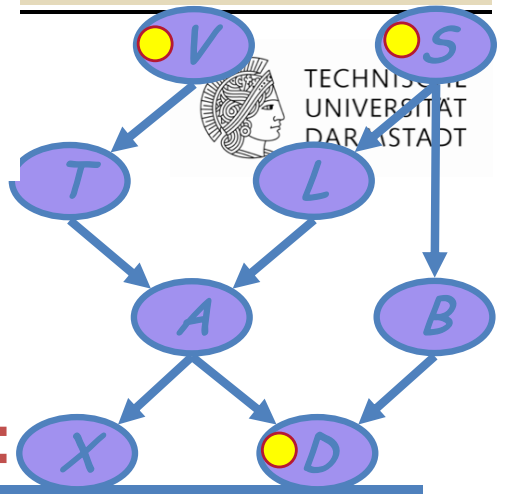
$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) P(x \mid a) f_{P(d|a,b)}(a,b)$$

- Eliminating x, we get

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Given evidence *V = t, S = f, D = t*
- Compute *P(L, V = t, S = f, D = t)*

Dealing with Evidence



- Initial factors, after setting evidence:

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) P(x \mid a) f_{P(d|a,b)}(a,b)$$

- Eliminating *x*, we get

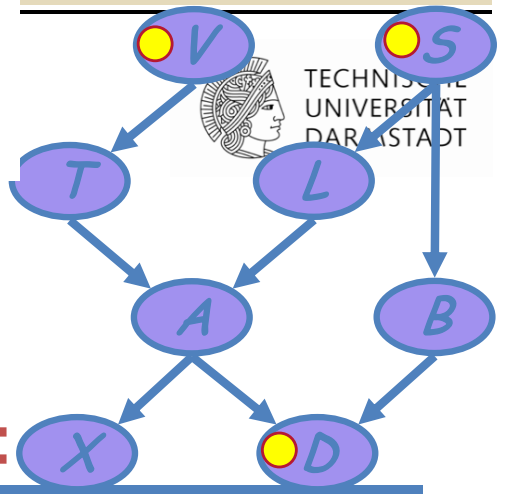
$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Eliminating *t*, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_{P(b|s)}(b) f_t(a,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Given evidence $V = t, S = f, D = t$
- Compute $P(L, V = t, S = f, D = t)$

Dealing with Evidence



- Initial factors, after setting evidence:

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) P(x \mid a) f_{P(d|a,b)}(a,b)$$

- Eliminating x, we get

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Eliminating t, we get

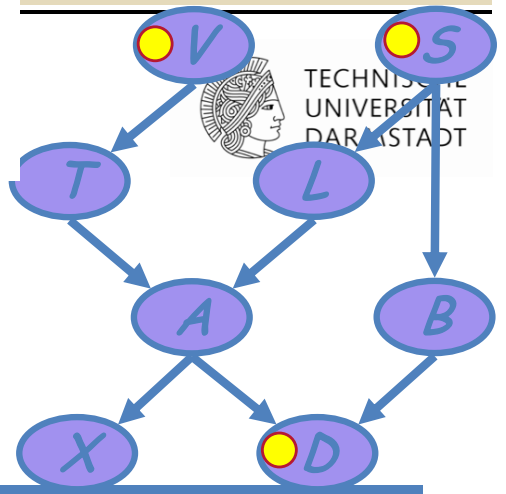
$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_{P(b|s)}(b) f_t(a,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Eliminating a, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_{P(b|s)}(b) f_a(b,l)$$

- Given evidence $V = t, S = f, D = t$
- Compute $P(L, V = t, S = f, D = t)$

Dealing with Evidence



- Initial factors, after setting evidence:

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) P(x \mid a) f_{P(d|a,b)}(a,b)$$

- Eliminating x, we get

$$f_{P(v)} f_{P(s)} f_{P(t|v)}(t) f_{P(l|s)}(l) f_{P(b|s)}(b) P(a \mid t,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Eliminating t, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_{P(b|s)}(b) f_t(a,l) f_x(a) f_{P(d|a,b)}(a,b)$$

- Eliminating a, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_{P(b|s)}(b) f_a(b,l)$$

- Eliminating b, we get

$$f_{P(v)} f_{P(s)} f_{P(l|s)}(l) f_b(l)$$

Summary: Variable elimination algorithm

- Given a BN and a query P(X,e) / P(e)
- Instantiate evidence **e**
- Prune non-active vars for {X,**e**}

IMPORTANT!!!

- Choose an ordering on variables, e.g., X_1, …, X_n
- Initial *factors* $\{f_1,\ldots,f_n\}$: $f_i = P(X_i|\mathbf{Pa}_{Xi})$ (CPT for X_i)
- For i = 1 to n, If $X_i \notin \{X,\mathbf{E}\}$ ← must be eliminated
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \sum_{X_i} \prod_{j=1} f_j$$

  - Variable X_i has been eliminated! Add g to the set of factors
- Normalize P(X,**e**) to obtain P(X|**e**)

Can also be used for finding the most probable explanation; useful for classification

$\sum_b \prod$ ← Elimination operator

factor B: P(b|a) P(d|b,a) P(e|b,c)

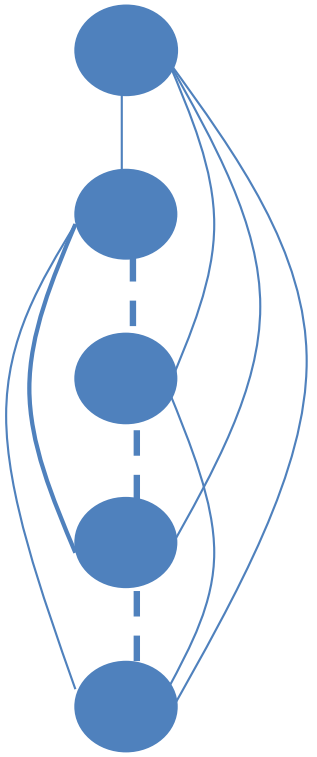
facotr C: P(c|a) $h^B(a,d,c,e)$

factor D: $h^C(a,d,e)$

factor E: e=0 $h^D(a,e)$

factor A: P(a) $h^E(a)$

$P(a|e=0)$



To do so, replace sum by max

TECHNISCHE UNIVERSITÄT DARMSTADT

$\sum$ is replaced by ***max*** :

$$MPE = \max_{a,e,d,c,b} P(a)P(c\,|\,a)P(b\,|\,a)P(d\,|\,a,b)P(e\,|\,b,c)$$

$\max_b \prod$ ← **Elimination operator**

factor B: P(b|a) P(d|b,a) P(e|b,c)

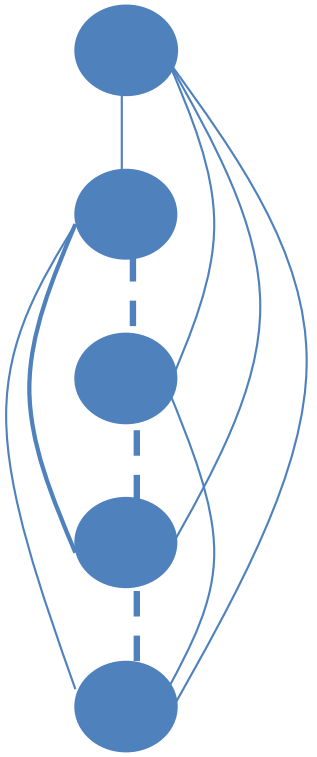
facotr C: P(c|a) $h^B(a,d,c,e)$

factor D: $h^C(a,d,e)$

factor E: e=0 $h^D(a,e)$

factor A: P(a) $h^E(a)$

$P(a|e=0)$



Generating the explanations

TECHNISCHE UNIVERSITÄT DARMSTADT

Two passes algorithm:
(Top-Down) Max Probs (Bottom-Up) Max Configuration

5. $b' = \arg\max_b P(b \mid a') \times P(d' \mid b, a') \times P(e' \mid b, c')$

4. $c' = \arg\max_c P(c \mid a') \times h^B(a', d', c, e')$

3. $d' = \arg\max_d h^C(a', d, e')$

2. $e' = 0$

1. $a' = \arg\max_a P(a) \cdot h^E(a)$

B: P(b|a) P(d|b,a) P(e|b,c)

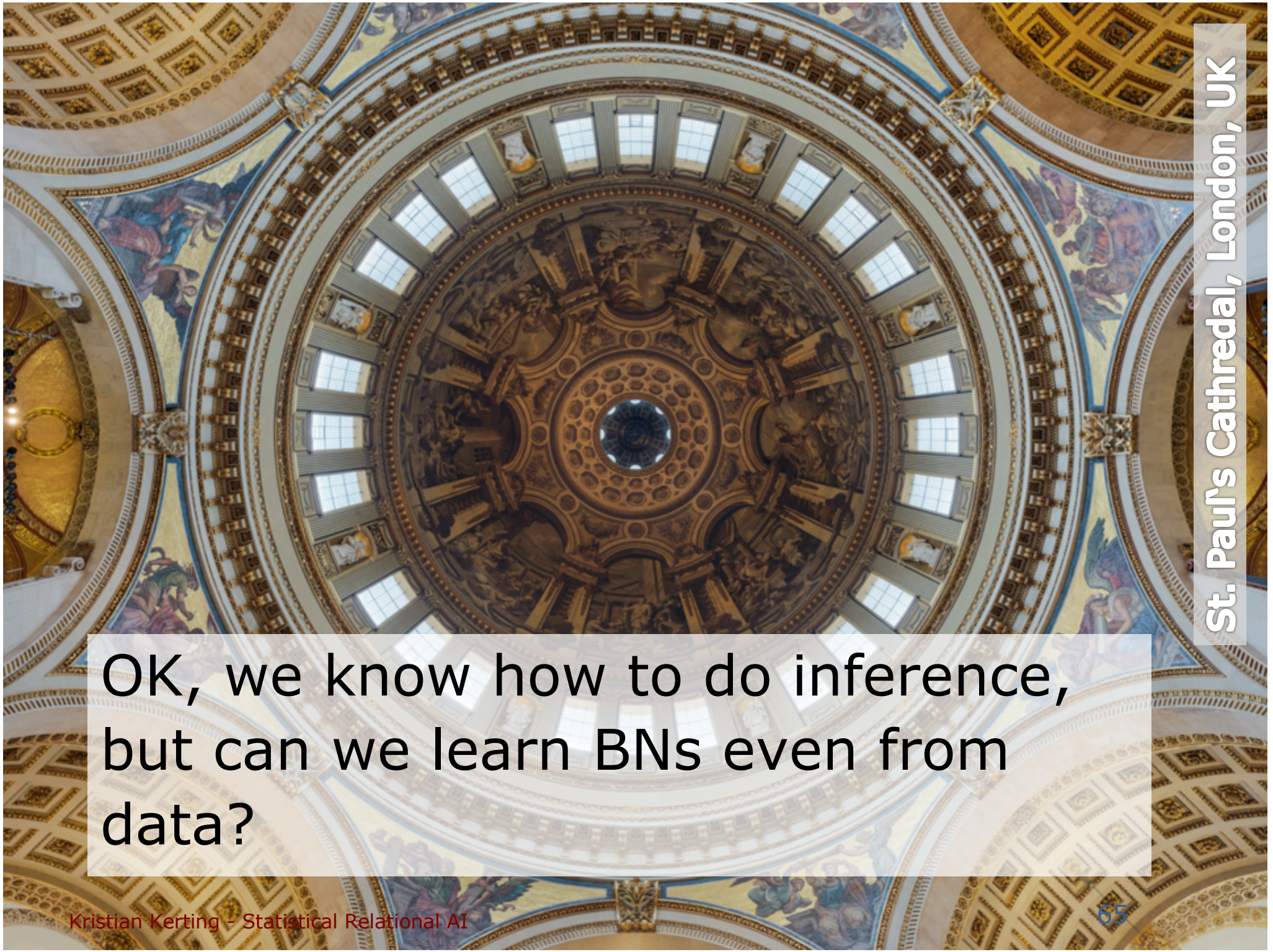
C: $h^B(a, d, c, e)$
P(c|a)

D: $h^C(a, d, e)$

E: e=0 $h^D(a, e)$

A: P(a) $h^E(a)$

Return (a', b', c', d', e')



OK, we know how to do inference, but can we learn BNs even from data?

Why bothering with learning?

TECHNISCHE UNIVERSITÄT DARMSTADT

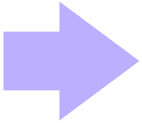
- **Bottleneck of knowledge aquisition**
  - Expensive, difficult
  - Normally, no expert is around
- **Data is cheap !**
  - Huge amount of data avaible, e.g.
    - Literature Databases
    - Web mining, e.g. log files
    - ....
    - Stopped here

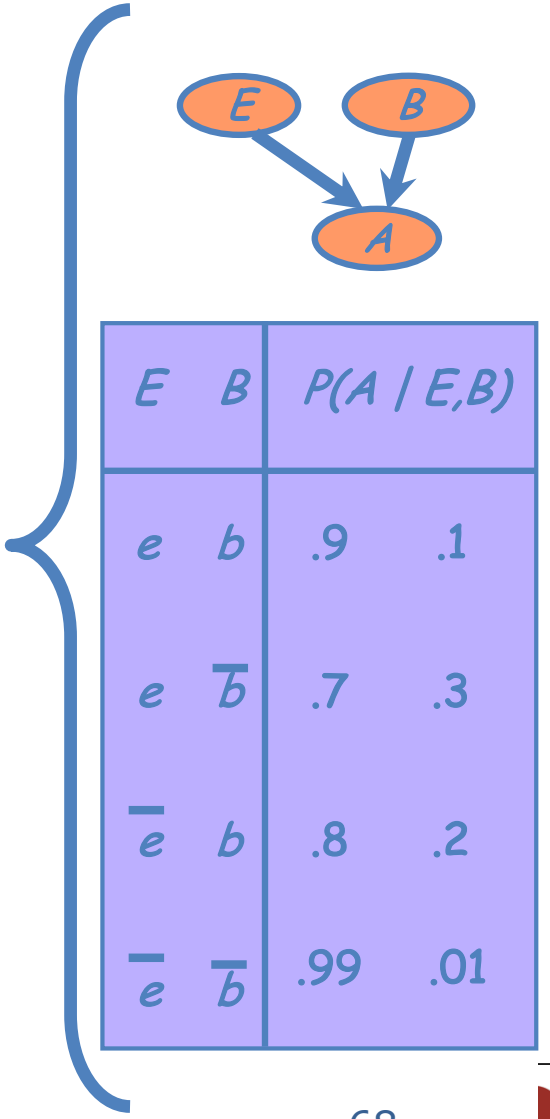
Why Learning Bayesian Networks?

TECHNISCHE UNIVERSITÄT DARMSTADT

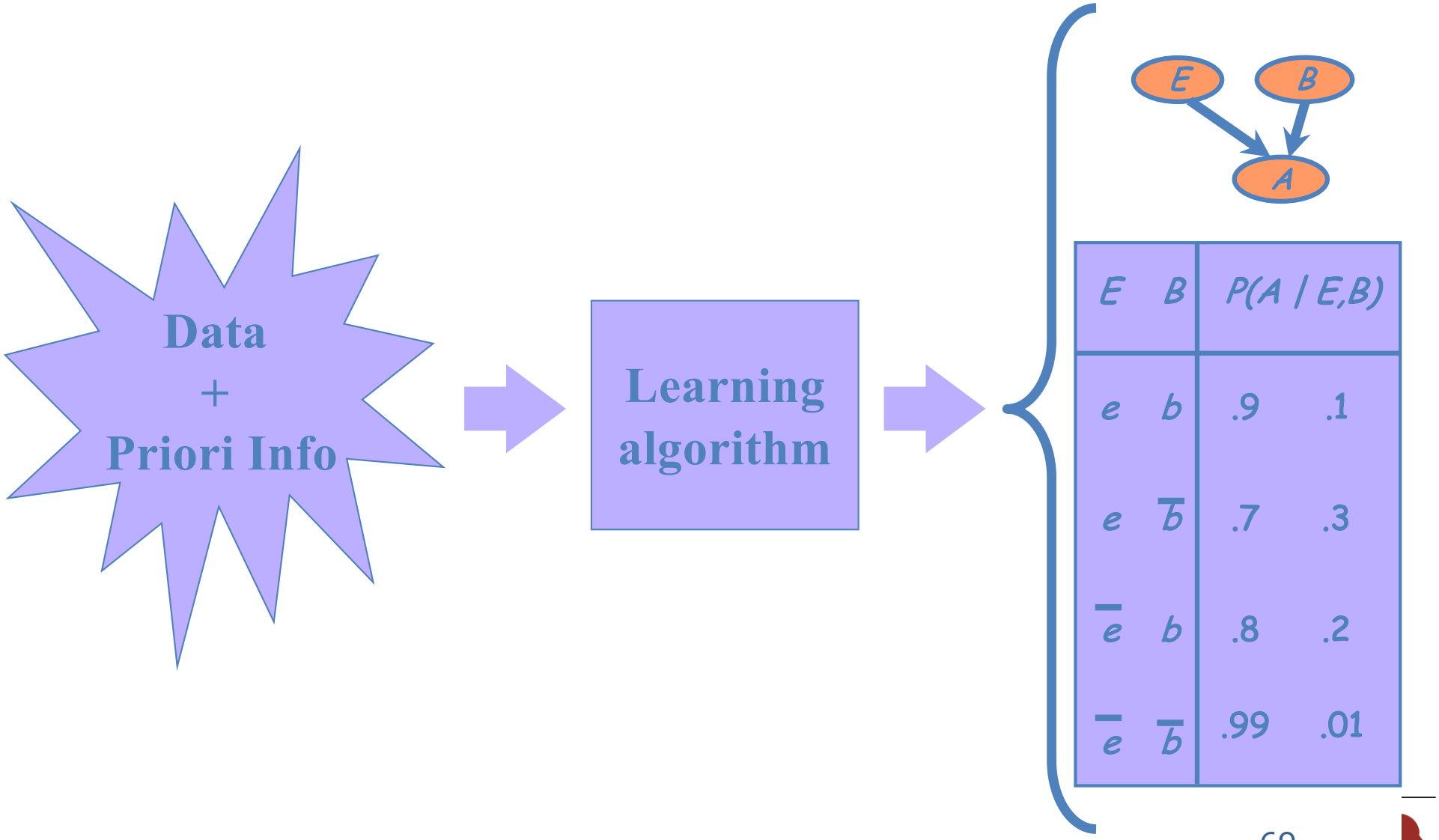
- Conditional independencies and graphical language capture structure of many real-world distributions
- **Graph structure provides much insight** into domain: "knowledge discovery"
- **Learned model can be used for many tasks**
- Automatically **dealing with missing data** and **hidden variables**

Learning With Bayesian Networks





Learning With Bayesian Networks



What does the data look like?

TECHNISCHE UNIVERSITÄT DARMSTADT

attributes/variables

complete data set

A1	A2	A3	A4	A5	A6	
true	true	false	true	false	false	**X1**
false	true	true	true	false	false	**X2**
...	...	...	...	...	...	⋮
true	false	false	false	true	true	**XM**

data cases

What does the data look like?

TECHNISCHE UNIVERSITÄT DARMSTADT

incomplete data set

A1	A2	A3	A4	A5	A6
true	true	?	true	false	false
?	true	?	?	false	false
...	...	...	...	...	...
true	false	?	false	true	?

- **Real-world data: states of some random variables are missing**
  - **E.g. medical diagnose: not all patient are subjects to all test**
  - **Parameter reduction, e.g. clustering, ...**

What does the data look like?

TECHNISCHE UNIVERSITÄT DARMSTADT

incomplete data set

A1	A2	A3	A4	A5	A6
true	true	?	true	false	false
?	true	?	?	false	false
...	...	...	...	...	...
true	false	?	false	true	?

- **Real-world data: states of some random variables are missing**
  - **E.g. medical diagnose: not all patient are subjects to all test**
  - **Parameter reduction, e.g. clustering, ...**

missing value

What does the data look like?

TECHNISCHE UNIVERSITÄT DARMSTADT

hidden/ latent

A1	A2	A3	A4	A5	A6
true	true	?	true	false	false
?	true	?	?	false	false
...	...	...	...	...	...
true	false	?	false	true	?

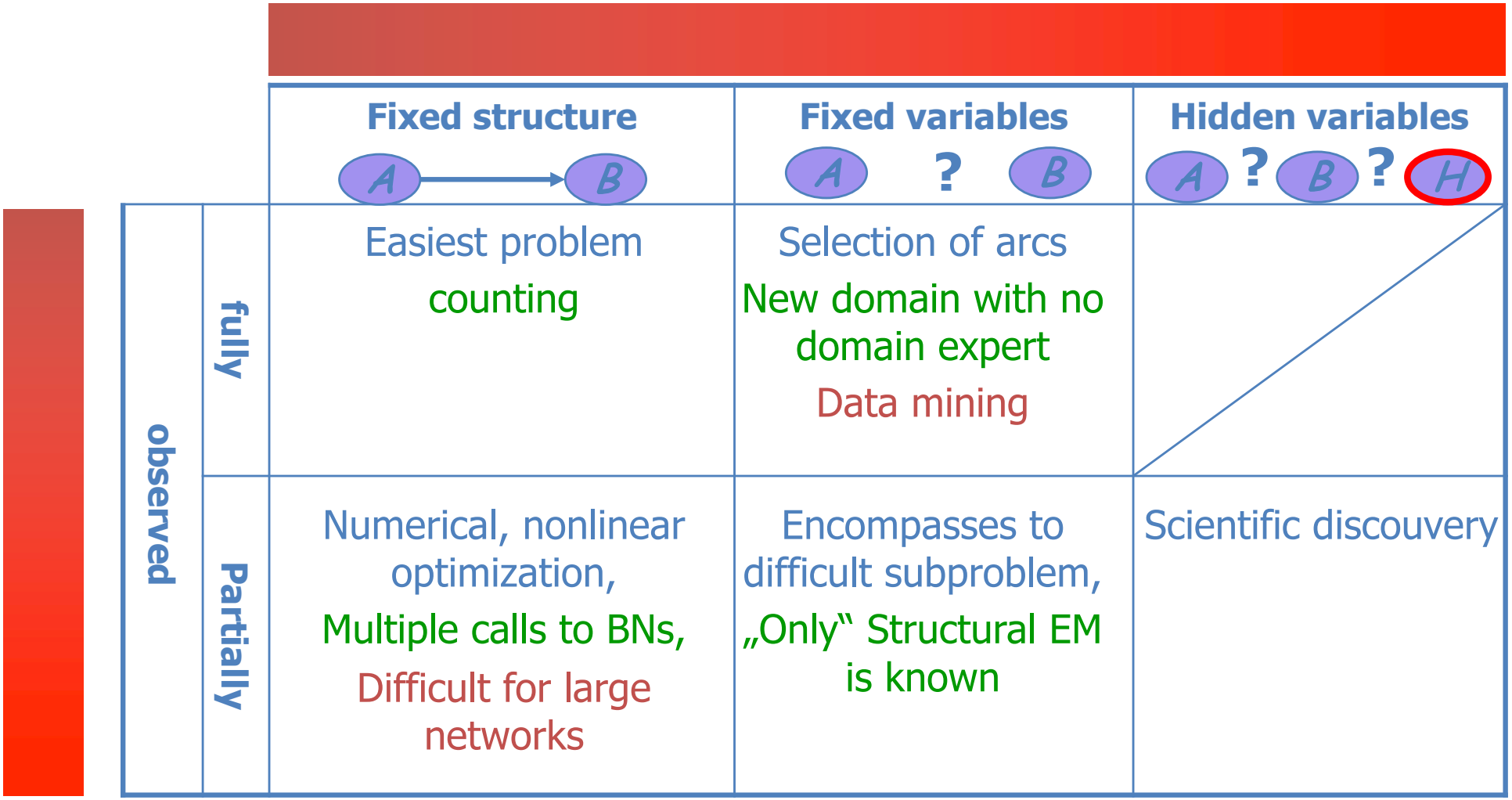
incomplete data set

- **Real-world data: states of some random variables are missing**
  - **E.g. medical diagnose: not all patient are subjects to all test**
  - **Parameter reduction, e.g. clustering, ...**

missing value

Learning With Bayesian Networks

TECHNISCHE UNIVERSITÄT DARMSTADT



Parameter Estimation and IID

TECHNISCHE UNIVERSITÄT DARMSTADT

- Let $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ set of data over m RVs
- $X_i \in \mathcal{X}$ is called a *data case*
- iid - assumption:
  - All data cases are independently sampled from identical distributions

Find:
Parameters Θ of CPDs which match the data best

Maximum Likelihood - Parameter Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- What does „best matching“ mean ?

Find paramteres Θ which have most likely produced the data

Maximum Likelihood - Parameter Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- What does „best matching“ mean ?
  1. MAP parameters $\Theta^* = \arg\max_\Theta P(\Theta|\mathcal{X})$

$$= \arg\max_\Theta P(\mathcal{X}|\Theta) \cdot \frac{\cancel{P(\Theta)}}{\cancel{P(\mathcal{X})}}$$

  2. **Data is equally likely for all parameters**
  3. **All parameters are apriori equally likely**

Maximum Likelihood - Parameter Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- What does „best matching“ mean ?

Find:

ML parameters

Taking the log does not affect the maximum

$$\Theta^* = \arg\max_\Theta P(\mathcal{X}|\Theta)$$

Likelihood $\mathcal{L}(\Theta|\mathcal{X})$ **of the params given the data**

$$\Theta^* = \arg\max_\Theta \log P(\mathcal{X}|\Theta)$$

Log-Likelihood $\mathcal{LL}(\Theta|\mathcal{X})$

Learning With Bayesian Networks

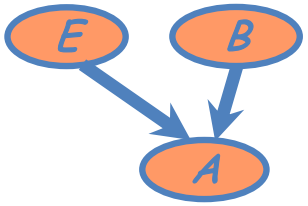
TECHNISCHE UNIVERSITÄT DARMSTADT

observed	Fixed structure 	Fixed variables 	Hidden variables 
fully	Easiest problem counting ?	Selection of arcs New domain with no domain expert Data mining	
Partially	Numerical, nonlinear optimization, Multiple calls to BNs, Difficult for large networks	Encompasses to difficult subproblem, „Only" Structural EM is known	Scientific discouvery

Known Structure, Complete Data

TECHNISCHE UNIVERSITÄT DARMSTADT

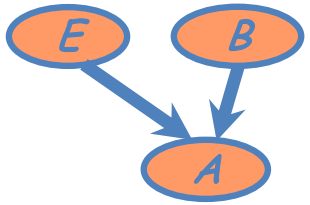
E, B, A
<Y,N,N>
<Y,N,Y>
<N,N,Y>
<N,Y,Y>
.
.
<N,Y,Y>



E	B	P(A | E,B)	
e	b	?	?
e	$\bar{b}$	?	?
$\bar{e}$	b	?	?
$\bar{e}$	$\bar{b}$	?	?

Learning algorithm

- **Network structure is specified**
  - **Only estimation of parameters**
- **No missing data values**



E	B	P(A | E,B)	
e	b	.9	.1
e	$\bar{b}$	.7	.3
$\bar{e}$	b	.8	.2
$\bar{e}$	$\bar{b}$	.99	.01

ML Parameter Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

A1	A2	A3	A4	A5	A6
true	true	false	true	false	false
false	true	true	true	false	false
...	...	...	...	...	...
true	false	false	false	true	true

$$\mathcal{LL}(\Theta|\mathcal{X}) = \log P(X_1, X_2, \ldots, X_n|\Theta)$$

(iid) $$= \log \prod_{i=1}^{n} P(X_i|\Theta)$$

$\log \prod$

$$= \sum_{i=1}^{n} \log P(X_i|\Theta) = \sum_{i=1}^{n} \log P(x_i^1, x_i^2, \ldots, x_i^m|\Theta)$$

$= \sum \log$

$$= \sum_{i=1}^{n} \log \left(\prod_{j=1}^{m} P(x_i^j | \mathrm{pa}(x_i^j), \Theta) \right)$$ **(BN semantics)**

$\log \prod$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \log P(x_i^j | \mathrm{pa}(x_i^j), \Theta)$$

$= \sum \log$

$$= \sum_{j=1}^{m} \sum_{i=1}^{n} \log P(x_i^j | \mathrm{pa}(x_i^j), \Theta_j)$$

$$= \sum_{j=1}^{m} \mathcal{LL}(\Theta_j|\mathcal{X})$$

Only local parameters of family of Aj involved

Each factor individually !!

ML Parameter Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

A1	A2	A3	A4	A5	A6
true	true	false	true	false	false
false	true	true	true	false	false
...	...	...	...	...	...
true	false	false	false	true	true

$$\mathcal{LL}(\Theta|\mathcal{X}) = \log P(X_1, X_2, \ldots, X_n|\Theta)$$

(iid) $$= \log \prod_{i=1}^{n} P(X_i|\Theta)$$

$$\log \prod \quad \sum^{n} \quad \sum^{n} \quad |\Theta)$$

$$= \sum$$

Decomposability of the likelihood

ics)

$$\log \prod$$

$$= \sum \log$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \log P(x_i^j | \mathrm{pa}(x_i^j), \Theta)$$

$$= \sum_{j=1}^{m} \sum_{i=1}^{n} \log P(x_i^j | \mathrm{pa}(x_i^j), \Theta_j)$$

$$= \sum_{j=1}^{m} \mathcal{LL}(\Theta_j|\mathcal{X})$$

Only local parameters of family of Aj involved

Each factor individually !!

Decomposability of Likelihood

TECHNISCHE UNIVERSITÄT DARMSTADT

If data set is **complete/fully observed** (i.e. no "?")

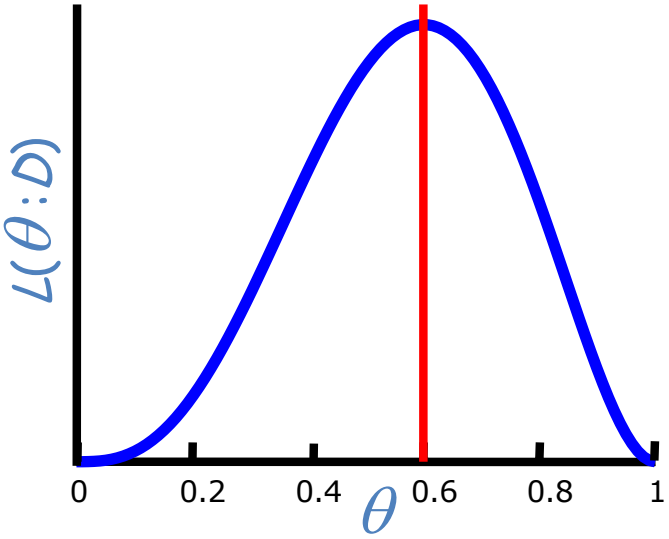
- we can maximize each local likelihood function **independently**, and
- then **combine** the solutions to get an MLE solution
- This **decomposition** of the global problem to independent, local sub-problems allows us to come up with efficient solutions to the MLE problem

Likelihood Function: Multinomials

TECHNISCHE UNIVERSITÄT DARMSTADT

$$L(\theta : D) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

- The likelihood for the sequence H, T, T, H, H is



$$L(\theta : D) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

Preference towards heads

General case:

$$L(\Theta : D) = \prod_{k=1}^{K} \theta_k^{N_k}$$

Count of k^{th} outcome in D

Probability of k^{th} outcome

Likelihood for Binominals (2 states only)

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Compute partial derivative**

$$\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v|\mathcal{X}) = \frac{\partial}{\partial \theta_i} \left(N_1 \log \theta_1 + N_2 \log(1-\theta_1)\right)$$

$$= \frac{N_1}{\theta_1} + \frac{N_2}{1-\theta_1}$$

$\theta_1 + \theta_2 = 1$

- **Set partial derivative zero**

$$\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v|\mathcal{X}) = 0 \Leftrightarrow \frac{N_1}{\theta_1} + \frac{N_2}{1-\theta_1} = 0$$

=> MLE is $\theta_1^* = \frac{N_1}{N_1 + N_2}$

Likelihood for Binominals (2 states only)

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Compute partial derivative**

$$\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v|\mathcal{X}) = \frac{\partial}{\partial \theta_i} \left(N_1 \log \theta_1 + N_2 \log(1-\theta_1)\right)$$

$$= \frac{N_1}{\theta_1} + \frac{N_2}{1-\theta_1}$$

$\theta_1 + \theta_2 = 1$

- **Set partial derivative zero**

$$\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v|\mathcal{X}) = 0 \Leftrightarrow \frac{N_1}{\theta_1} + \frac{N_2}{1-\theta_1} = 0$$

In general, for multinomials (>2 states), the MLE is

$$\theta_i^* = \frac{N_i}{\sum_j N_j}$$

Likelihood for Conditional Multinominals

TECHNISCHE UNIVERSITÄT DARMSTADT

- $P(V = k | \mathrm{pa}(V) = \mathbf{pa})$ multinomial for each joint state pa of the parents of V:

$$P(k|1,1), P(k|1,2), P(k|2,1), P(k|2,2)$$

- $$\mathcal{LL}(\Theta_v | \mathcal{X})$$
$$= \sum_{\mathbf{pa}} \sum_{k=1}^{K} \log \theta_{k|\mathbf{pa}}^{N_{k,\mathbf{pa}}} = \sum_{\mathbf{pa}} \sum_{k=1}^{K} N_{k,\mathbf{pa}} \cdot \theta_{k|\mathbf{pa}}$$

$$\theta^*_{k|\mathbf{pa}} = \frac{N_{k,\mathbf{pa}}}{N_{\mathbf{pa}}}$$

Learning With Bayesian Networks

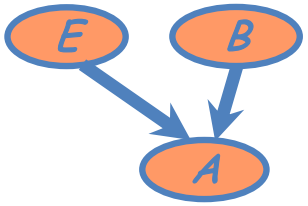
TECHNISCHE UNIVERSITÄT DARMSTADT

observed	Fixed structure (A → B)	Fixed variables (A ? B)	Hidden variables (A ? B ? H)
fully	Easiest problem counting 	Selection of arcs New domain with no domain expert Data mining	
Partially	Numerical, nonlinear optimization, Multiple calls to BNs, Difficult for large networks ?	Encompasses to difficult subproblem, „Only" Structural EM is known	Scientific discouvery

Known Structure, Incomplete Data

TECHNISCHE UNIVERSITÄT DARMSTADT

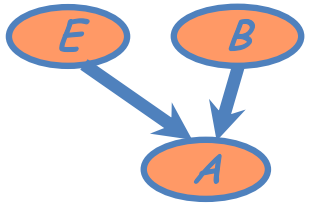
E, B, A
<Y,?,N>
<Y,N,?>
<N,N,Y>
<N,Y,Y>
.
.
<?,Y,Y>



E	B	P(A | E,B)	
e	b	?	?
e	$\bar{b}$	?	?
$\bar{e}$	b	?	?
$\bar{e}$	$\bar{b}$	?	?

Learning algorithm

- **Network structure is specified**
- **Data contains missing values**
  - **Need to consider assignments to missing values**



E	B	P(A | E,B)	
e	b	.9	.1
e	$\bar{b}$	.7	.3
$\bar{e}$	b	.8	.2
$\bar{e}$	$\bar{b}$	.99	.01

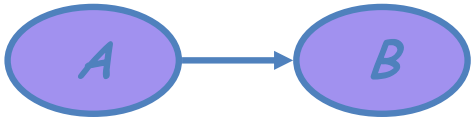
EM Idea

TECHNISCHE UNIVERSITÄT DARMSTADT

- If **data is complete**, ML parameter estimation is easy:
  - **simple counting** (1 iteration)
- But what if there are missing values, i.e., we are facing **incomplete data**?

1. **Complete data** (Imputation)
   - most probable?, average?, ... value
2. **Count**
3. **Iterate**

EM Idea: complete the data



$$\theta_{A=\text{true}} = \frac{1}{2}$$
$$\theta_{B=\text{true}|A=\text{true}} = \frac{1}{2}$$
$$\theta_{B=\text{true}|A=\text{false}} = \frac{1}{2}$$

incomplete data

A	B
true	true
true	?
false	true
true	false
false	?

complete

$$P(B = \text{true}|A = \text{true}) = 0.5$$
$$P(B = \text{true}|A = \text{false}) = 0.5$$

complete data **expected counts**

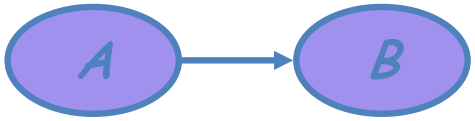
A	B	N
true	true	1.5
true	false	1.5
false	true	1.5
false	false	0.5

maximize

iterate

$$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.5+0.5} = 0.6$$
$$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$$
$$\theta_{B=\text{true}|A=\text{false}} = \frac{1.5}{1.5+0.5} = 0.75$$

EM Idea: complete the data



$$\theta_{A=\text{true}} = 0.6$$
$$\theta_{B=\text{true}|A=\text{true}} = 0.5$$
$$\theta_{B=\text{true}|A=\text{false}} = 0.875$$

complete

$$P(B = \text{true}|A = \text{true}) = 0.5$$
$$P(B = \text{true}|A = \text{false}) = 0.875$$

incomplete data

A	B
true	true
true	?
false	true
true	false
false	?

complete data **expected counts**

A	B	N
true	true	1.5
true	false	1.5
false	true	1.875
false	false	0.125

maximize

iterate

$$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.875+0.125} = 0.6$$
$$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$$
$$\theta_{B=\text{true}|A=\text{false}} = \frac{1.875}{1.875+0.125} = 0.9375$$

EM for Multinominals

TECHNISCHE UNIVERSITÄT DARMSTADT

Random variable V with 1,…,K values

$$P(V = k) = \theta_k \qquad \sum_{k=1}^{K} \theta_k = 1$$

$$\mathcal{Q}(\Theta_v, \Theta') = \sum_{k=1}^{K} \log \theta_k^{EN_k} = \sum_{k=1}^{K} \log EN_k \cdot \theta_k$$

where EN_k are the expected counts of state k in the data, i.e.

$$EN_k = \sum_{i=1}^{m} P(k|X_i)$$

„MLE“:

$$\frac{EN_i}{\sum_k EN_k}$$

EM for Conditional Multinominals

TECHNISCHE UNIVERSITÄT DARMSTADT

- $P(V = k | \mathrm{pa}(V) = \mathrm{pa})$ multinomial for each joint state pa of the parents of V:

$$P(k|1,1), P(k|1,2), P(k|2,1), P(k|2,2)$$

$$\mathcal{Q}(\Theta_v, \Theta')$$

$$= \sum_{\mathrm{pa}} \sum_{k=1}^{K} \log \theta_{k|\mathrm{pa}}^{EN_{k,\mathrm{pa}}} = \sum_{\mathrm{pa}} \sum_{k=1}^{K} EN_{k,\mathrm{pa}} \cdot \theta_{k|\mathrm{pa}}$$

- „MLE"

$$\theta^*_{k|\mathrm{pa}} = \frac{EN_{k,\mathrm{pa}}}{EN_{\mathrm{pa}}}$$

Learning Parameters: incomplete data

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Initialize parameters
2. Compute **pseudo counts** for each variable

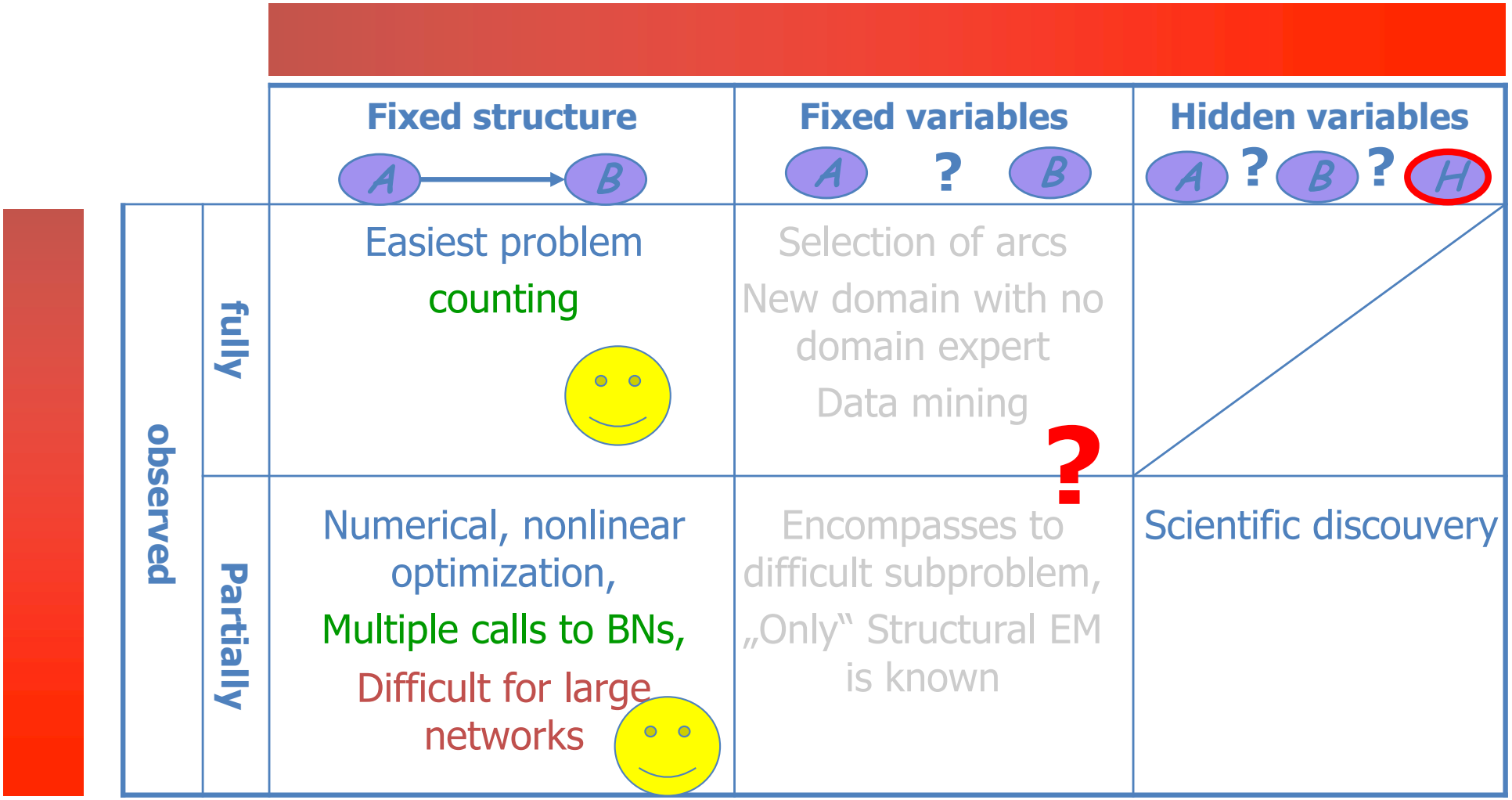
$$\theta^*_{k|\mathrm{pa}} = \frac{\sum_{i=1}^{m} P(k, \mathrm{pa}|X_i)}{\sum_{i=1}^{m} P(\mathrm{pa}|X_i)}$$

inference

3. **Set parameters to the (completed) ML estimates**
4. If not converged, iterate to 2

Learning With Bayesian Networks

TECHNISCHE UNIVERSITÄT DARMSTADT



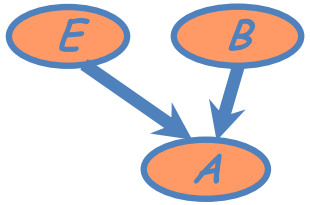
Unknown Structure, (In)complete Data

TECHNISCHE UNIVERSITÄT DARMSTADT

E, B, A
<Y,N,N>
<Y,N,Y>
<N,N,Y>
<N,Y,Y>
.
.
<N,Y,Y>

- **Network structure is not specified**
  - **Learnerr needs to select arcs & estimate parameters**
- **Data does not contain missing values**

E	B	P(A | E,B)	
e	b	?	?
e	$\bar{b}$	?	?
$\bar{e}$	b	?	?
$\bar{e}$	$\bar{b}$	?	?



E	B	P(A | E,B)	
e	b	.9	.1
e	$\bar{b}$	.7	.3
$\bar{e}$	b	.8	.2
$\bar{e}$	$\bar{b}$	.99	.01

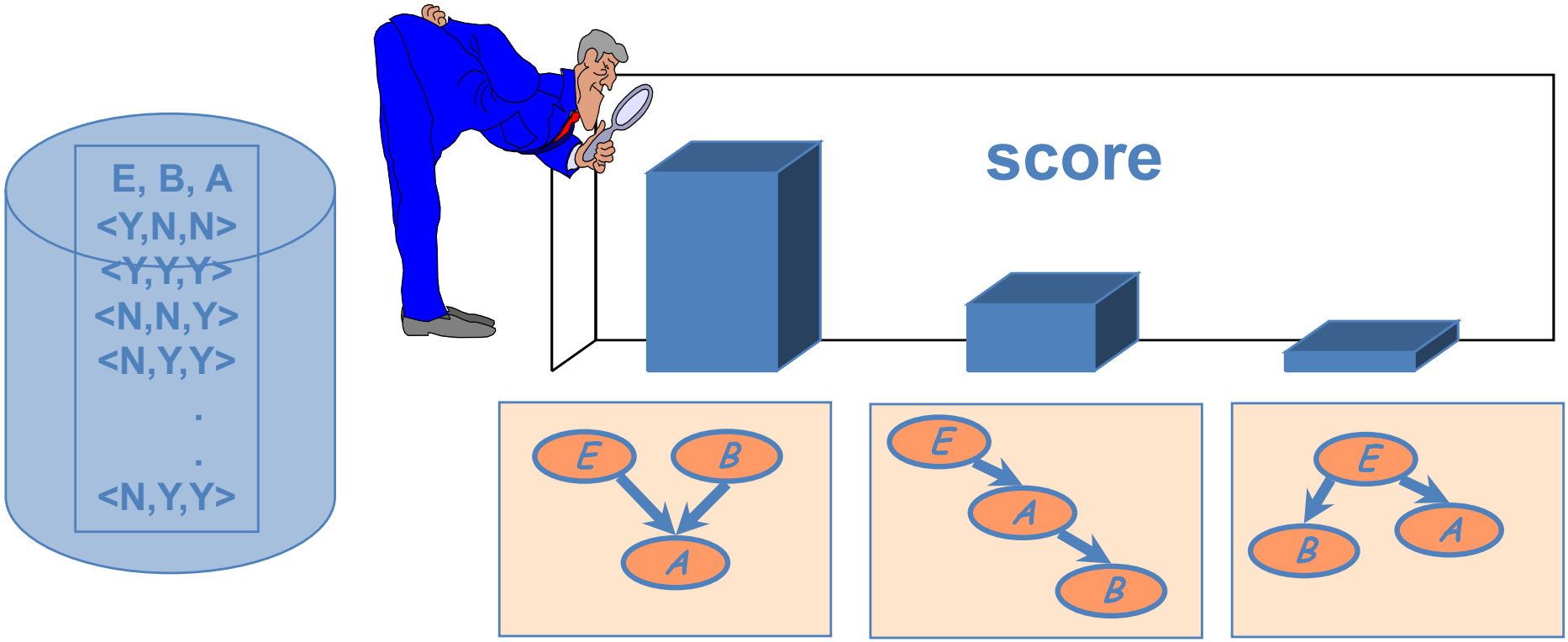
E, B, A
<Y,?,N>
<Y,N,?>
<N,N,Y>
<N,Y,Y>
.
.
<?,Y,Y>

- **Network structure is not specified**
- **Data contains missing values**
  - **Need to consider assignments to missing values**

Score-based Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

Define scoring function that evaluates how well a structure matches the data



Search for a structure that maximizes the score

Likelihood Score for Structure

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\ell(G:D) = \log L(G:D) = M \sum_i \left(I(X_i; Pa_i^G) - H(X_i) \right)$$

Mutual information between X_i and its parents

Entropy X_i

- Larger dependence of X_i on $Pa_i \Rightarrow$ higher score

- Adding arcs always helps
  - $I(X; Y) \leq I(X; \{Y,Z\})$
  - Max score attained by fully connected network
  - Overfitting: A bad idea…

Bayesian Information Criterion (BIC)

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\log P(D \mid G) = \ell(G : D) - \frac{\log M}{2} \dim(G) + O(1)$$

$$= \underbrace{M \sum_i \left(I(X_i; Pa_i^G) - H(X_i) \right)}_{\text{Fit dependencies in empirical distribution}} - \underbrace{\frac{\log M}{2} \dim(G)}_{\text{Complexity penalty}} + O(1)$$

- As M (amount of data) grows,
  - Increasing pressure to fit dependencies in distribution
  - Complexity term avoids fitting noise
- Asymptotic equivalence to MDL score
- Bayesian score/BIC is **consistent**
  - Observed data eventually overrides prior

Heuristic Search

TECHNISCHE UNIVERSITÄT DARMSTADT

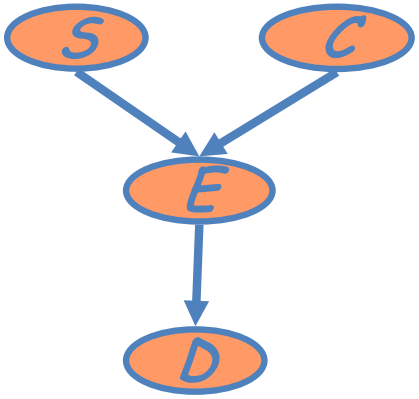
- Define a search space:
  - search states are possible structures
  - operators make small changes to structure
- Traverse space looking for high-scoring structures
- Search techniques:
  - Greedy hill-climbing
  - Best first search
  - Simulated Annealing
  - ...

Theorem: Finding maximal scoring structure with at most k parents per node is NP-hard for $k > 1$

Typically: Local Search

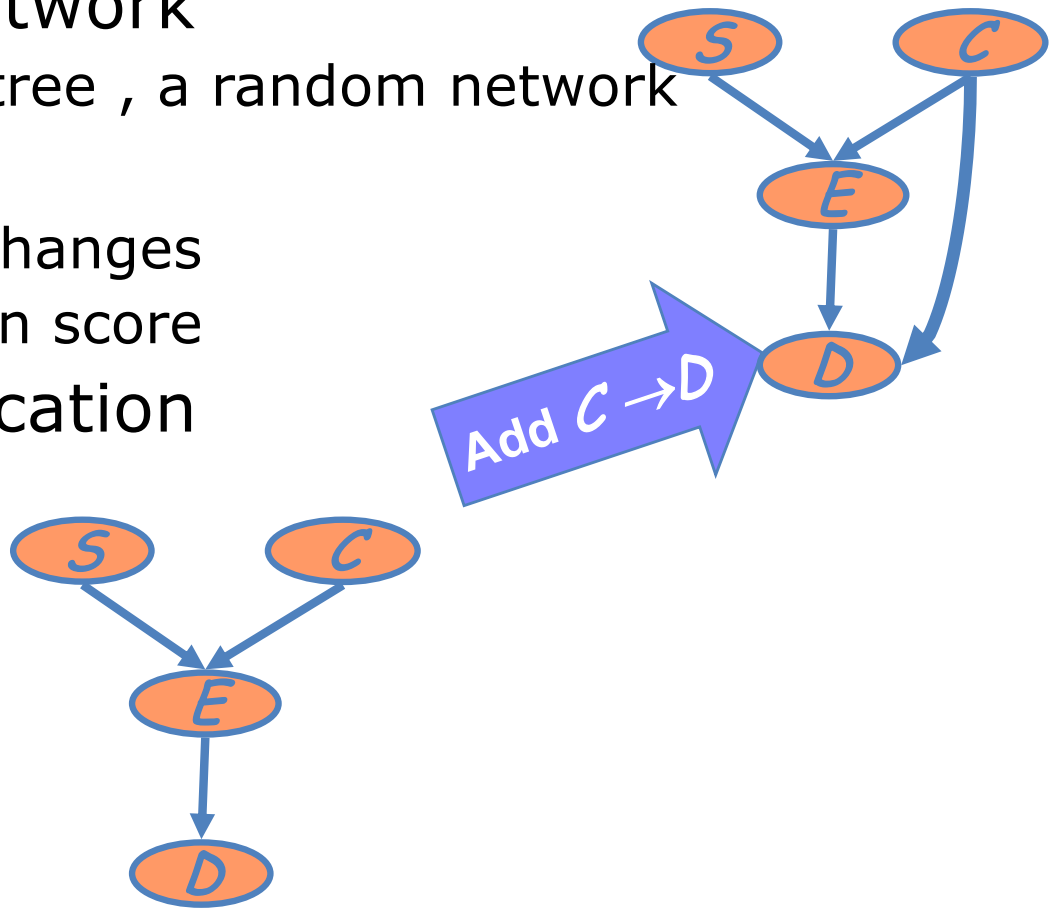
TECHNISCHE UNIVERSITÄT DARMSTADT

- Start with a given network
  - empty network, best tree , a random network
- At each iteration
  - Evaluate all possible changes
  - Apply change based on score
- Stop when no modification improves score



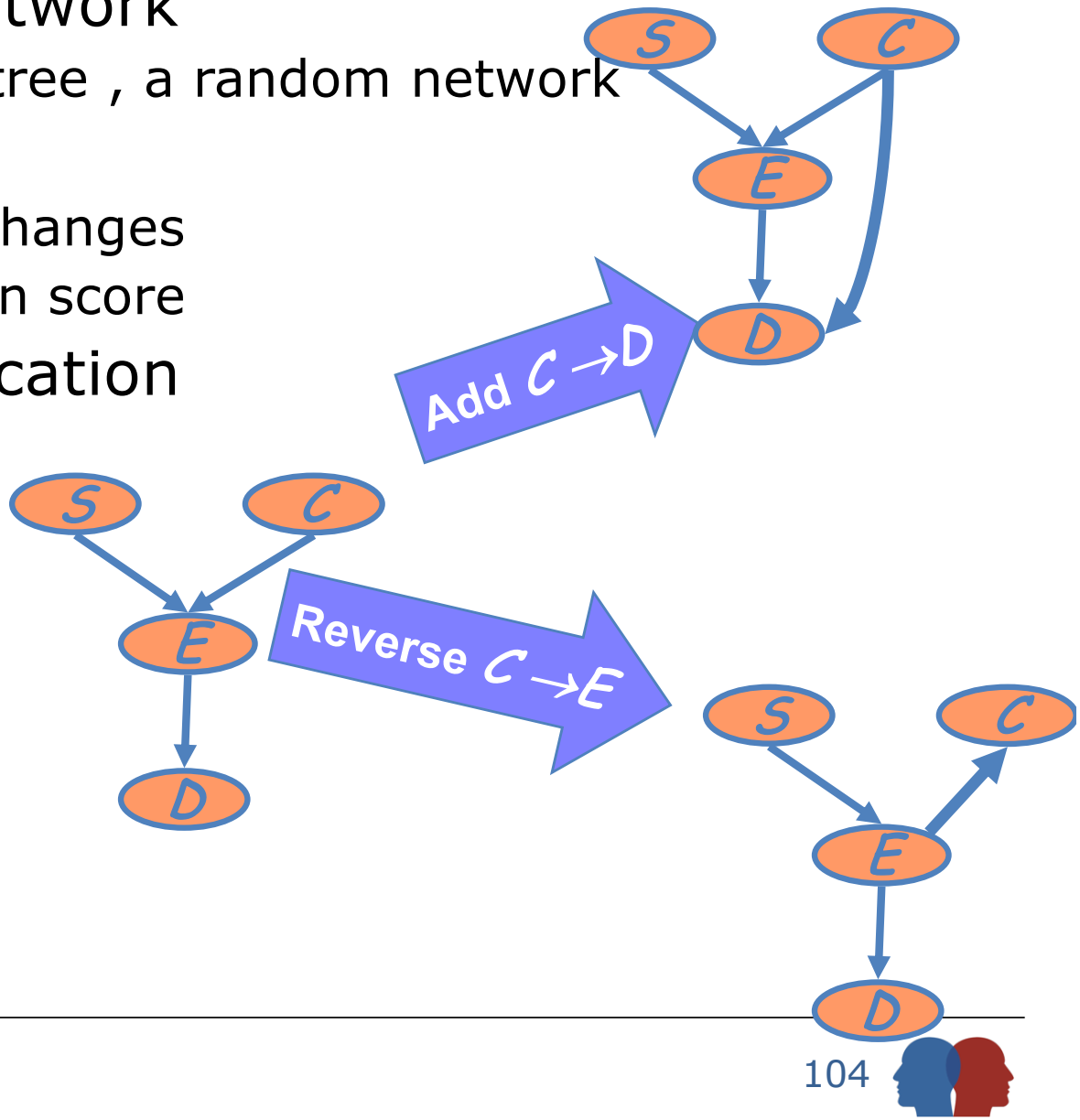
Typically: Local Search

- Start with a given network
  - empty network, best tree , a random network
- At each iteration
  - Evaluate all possible changes
  - Apply change based on score
- Stop when no modification improves score



Typically: Local Search

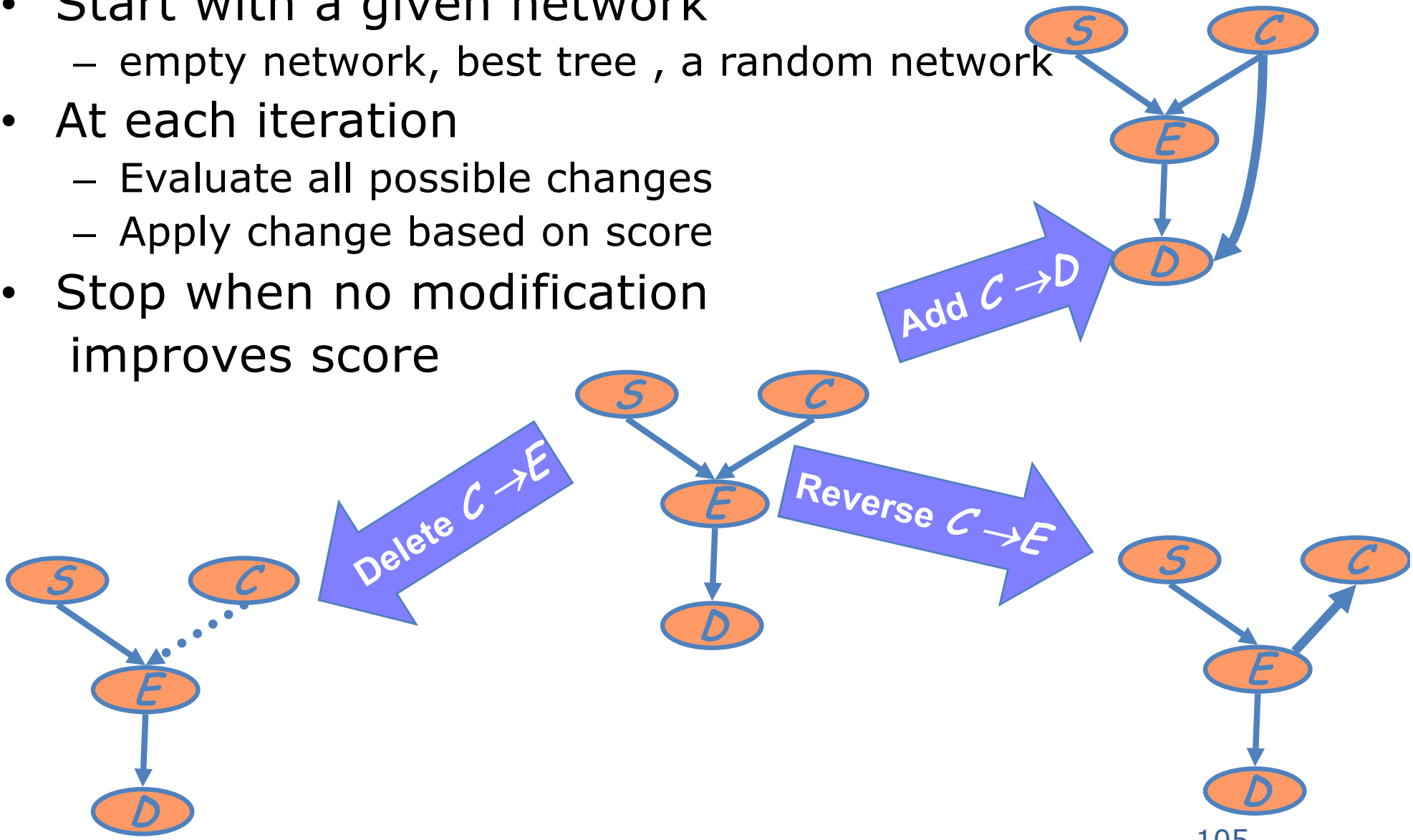
- Start with a given network
  - empty network, best tree , a random network
- At each iteration
  - Evaluate all possible changes
  - Apply change based on score
- Stop when no modification improves score



Typically: Local Search

TECHNISCHE UNIVERSITÄT DARMSTADT

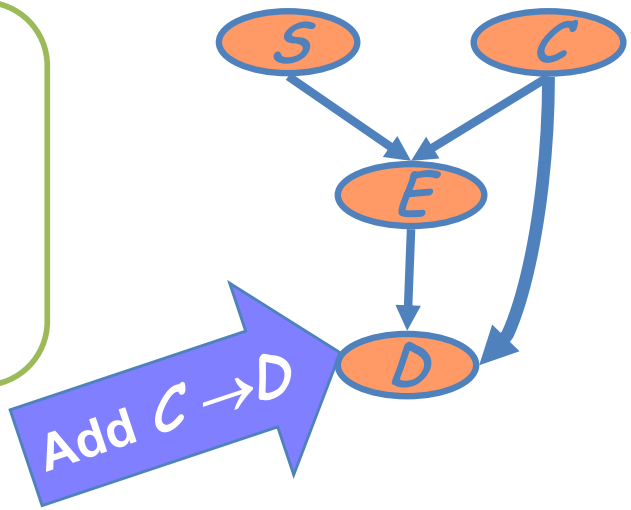
- Start with a given network
  - empty network, best tree , a random network
- At each iteration
  - Evaluate all possible changes
  - Apply change based on score
- Stop when no modification improves score

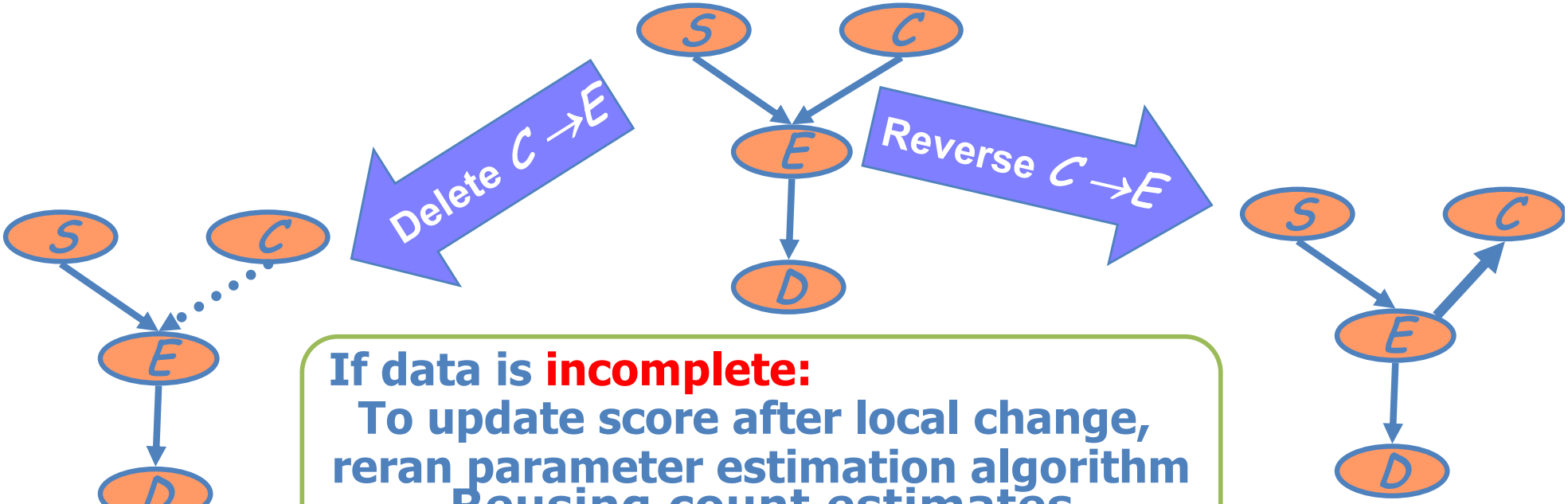


Typically: Local Search

TECHNISCHE UNIVERSITÄT DARMSTADT

If data is complete:
To update score after local change, only re-score (counting) families that changed





If data is incomplete:
To update score after local change, reran parameter estimation algorithm Reusing count estimates

Local Search in Practice

TECHNISCHE UNIVERSITÄT DARMSTADT

- Local search can get stuck in:
  - **Local Maxima:**
    - All one-edge changes reduce the score
  - **Plateaux:**
    - Some one-edge changes leave the score unchanged

- So, standard heuristics can escape both
  - Random restarts
  - TABU search
  - Simulated annealing

Lessons learnt

- Bayesian networks are DAGs with associated conditional probability tables
- Inference in general intractable
- Most basic inference approach: variable elimination
- Parameter estimation is easy when everything is observed
- When there are partially observed datacases we have us nonlinear optimization together with inference
- Model selection typically via heuristic search

St. Paul's Cathredal, London, UK