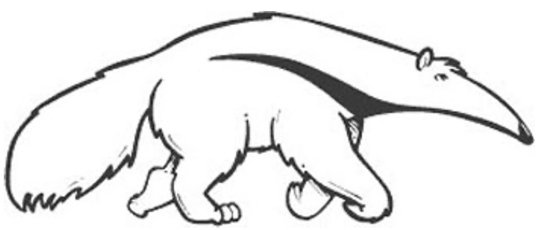
Algorithms for Causal Probabilistic Graphical Models

Class 1:
Introduction and Inference

Athens Summer School on AI
July 2024

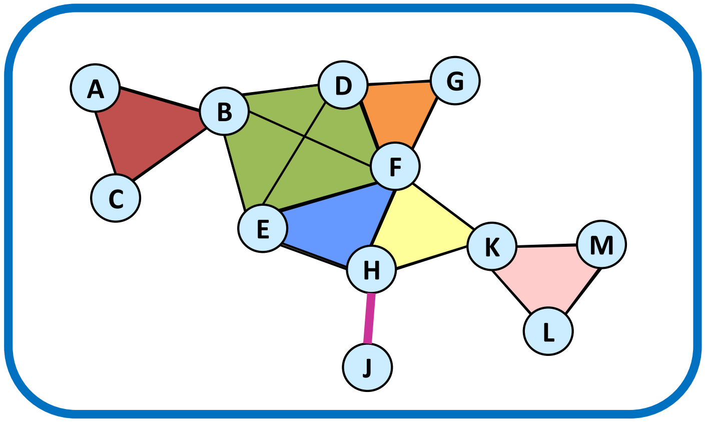
Prof. Rina Dechter
Prof. Alexander Ihler



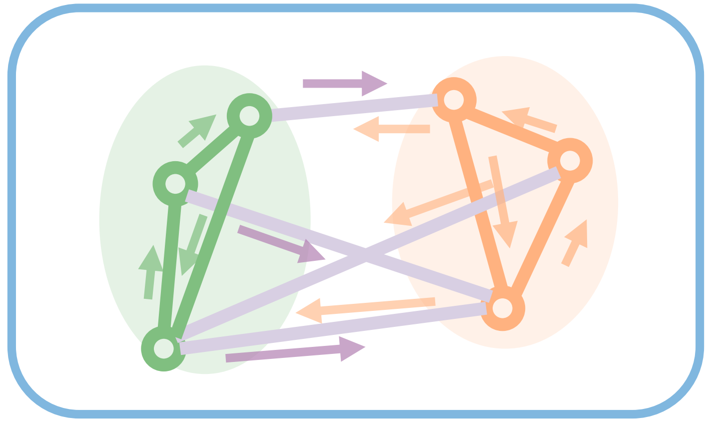
BREN:ICS INFORMATION AND COMPUTER SCIENCES
UNIVERSITY of CALIFORNIA IRVINE

Outline of Lectures

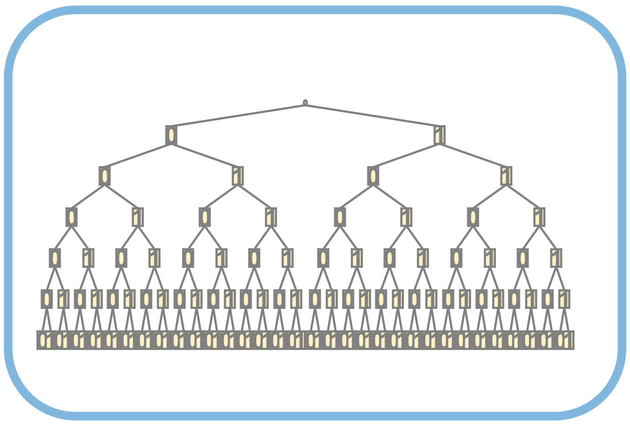
Class 1: Introduction & Inference



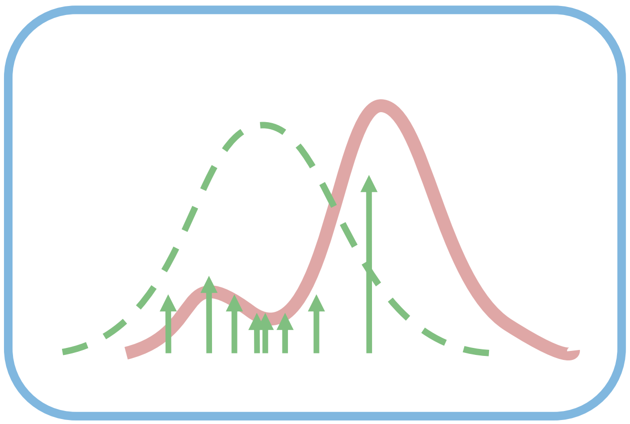
Class 2: Bounds & Variational Methods



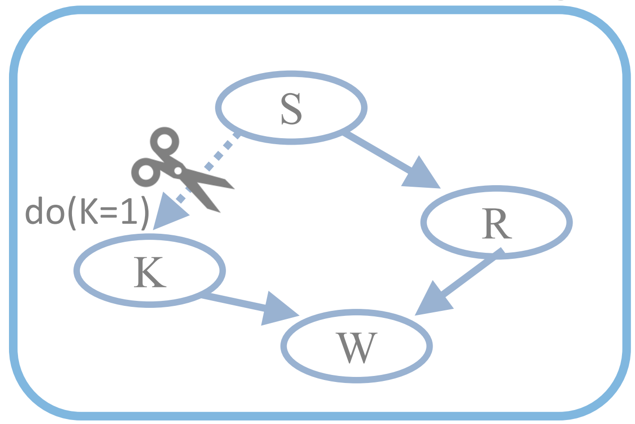
Class 3: Search Methods



Class 4: Monte Carlo Methods



Class 5: Causal Reasoning



Outline

- Graphical Models
- Inference Tasks
- Variable Elimination
- Tree Decomposition
- Variable Orderings
- Learning from Data

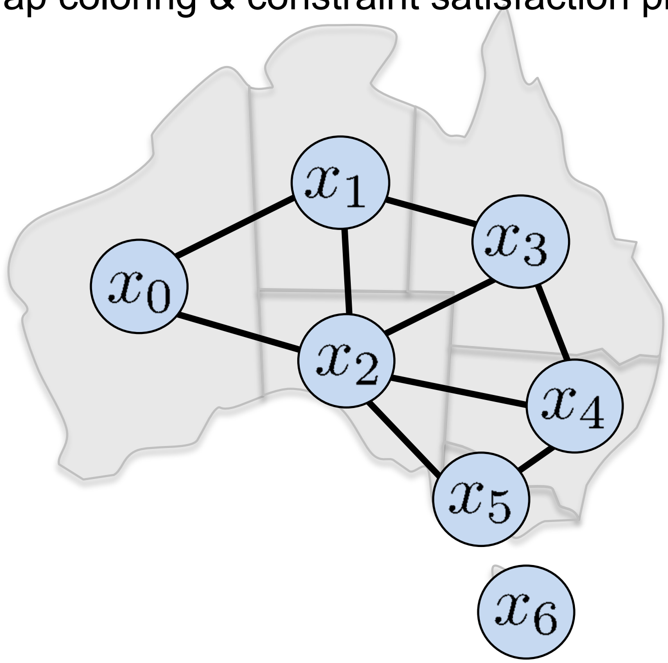
Graphical Models

Describe structure and interdependence in a model of the world

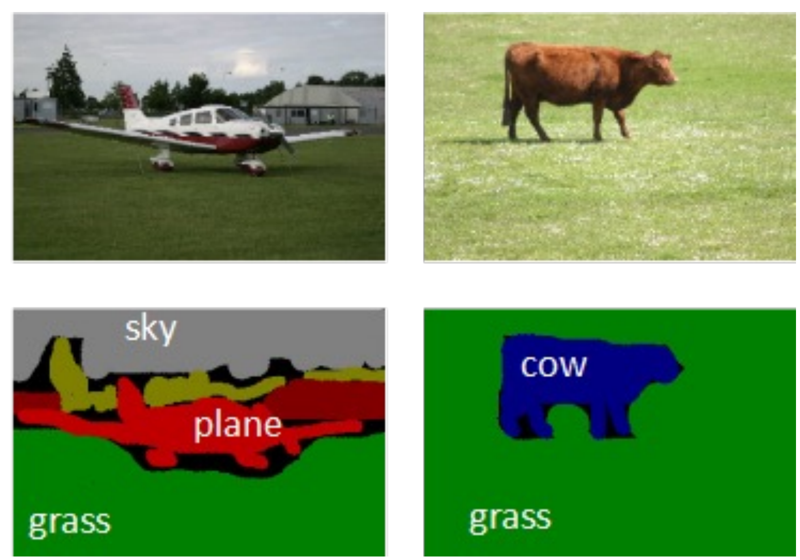
Examples:

- Markov Random Fields: correlations

Map coloring & constraint satisfaction problems



Semantic segmentation: fine-grain object recognition



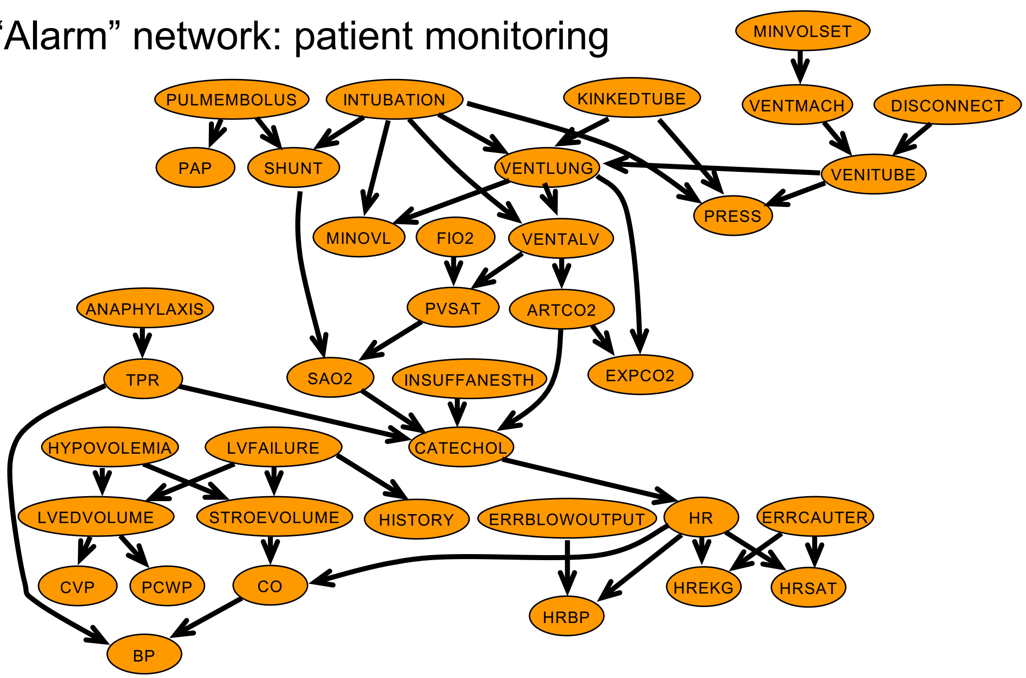
Graphical Models

Describe structure and interdependence in a model of the world

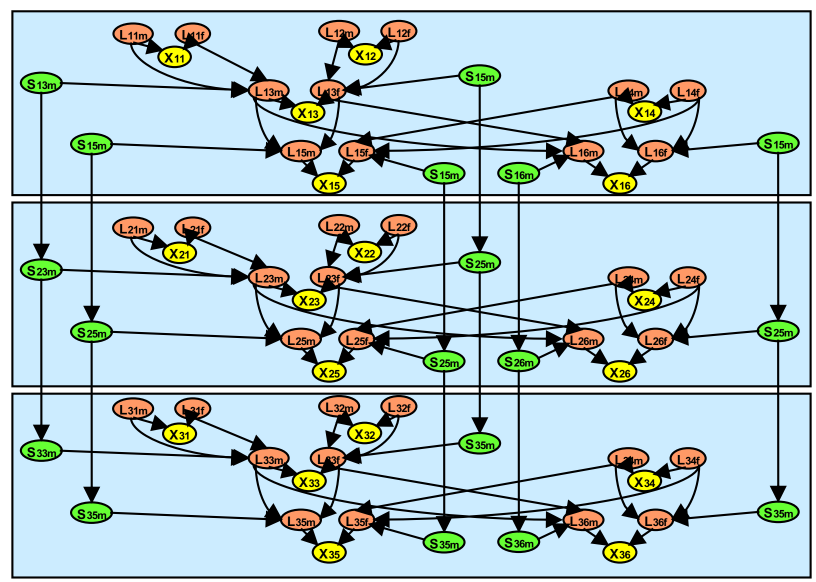
Examples:

- Markov Random Fields: correlations
- Bayesian Networks: conditional dependence

“Alarm” network: patient monitoring



Pedigree network: genetic inheritance



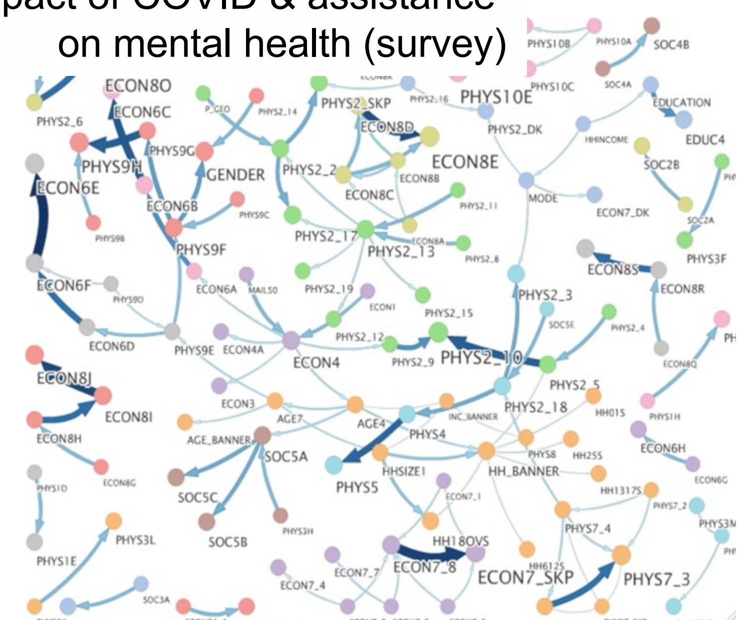
Graphical Models

Describe structure and interdependence in a model of the world

Examples:

- Markov Random Fields: correlations
- Bayesian Networks: conditional dependence
- Causal Networks: effect of intervention – what would happen if?

Impact of COVID & assistance on mental health (survey)

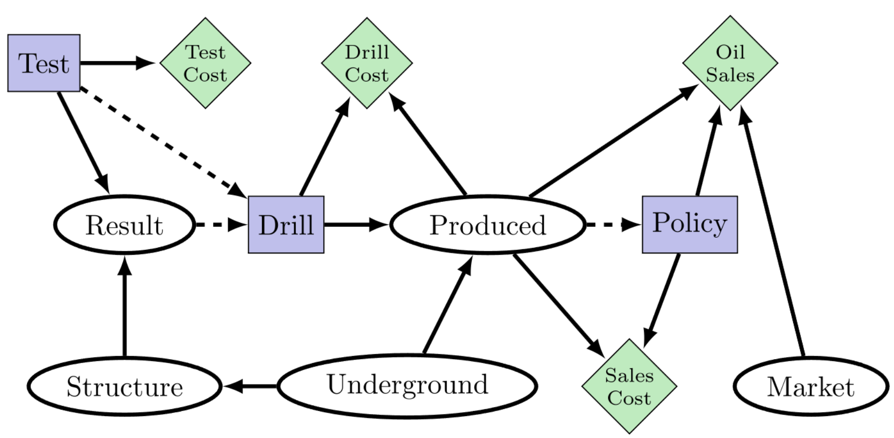


Graphical Models

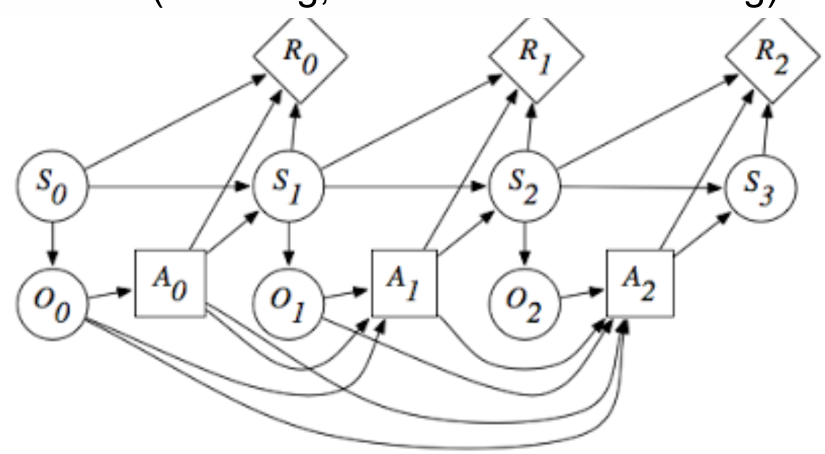
Describe structure and interdependence in a model of the world

Examples:
- Markov Random Fields: correlations
- Bayesian Networks: conditional dependence
- Causal Networks: effect of intervention – what would happen if?
- Influence Diagrams: actions and rewards – what should we do if?

“Oil Wildcatter” Decision Network



(Partially Observable) Markov Decision Process
(Planning, Reinforcement Learning)



Bayesian networks

Use **independence** and **conditional independence** to simplify a **joint probability**

- Joint probability, p(X=x,Y=y,Z=z)
  - The probability that event (x,y,z) happens.

- Conditional probability
  - The chain rule of probability tells us

    p(X=x,Y=y,Z=z) = p(X=x)   p(Y=y | X=x)   p(Z=z | X=x,Y=y)

    (x,y,z all happen)   (x happens)   (y happens given x happened)   (z happens given x,y happened)

  - Can use any order, e.g. (Z,X,Y):

    p(X=x,Y=y,Z=z) = p(Z=z)   p(X=x | Z=z)   p(Y=y | X=x,Z=z)

Independence

- X, Y independent:
  - $p(X=x,Y=y) = p(X=x)\, p(Y=y)$ for all x,y
  - Shorthand: $p(X,Y) = P(X)\, P(Y)$
  - Equivalent: $p(X|Y) = p(X)$ or $p(Y|X) = p(Y)$ (if $p(Y), p(X) > 0$)
  - Intuition: knowing X has no information about Y (or vice versa)

Independent probability distributions:

A	P(A)
0	0.4
1	0.6

B	P(B)
0	0.7
1	0.3

C	P(C)
0	0.1
1	0.9

This reduces representation size!

Joint:

A	B	C	P(A,B,C)
0	0	0	.4 * .7 * .1
0	0	1	.4 * .7 * .9
0	1	0	.4 * .3 * .1
0	1	1	…
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Independence

- X, Y independent:
  - $p(X{=}x,Y{=}y) = p(X{=}x)\ p(Y{=}y)$ for all x,y
  - Shorthand: $p(X,Y) = P(X)\ P(Y)$
  - Equivalent: $p(X|Y) = p(X)$ or $p(Y|X) = p(Y)$ (if $p(Y), p(X) > 0$)
  - Intuition: knowing X has no information about Y (or vice versa)

Independent probability distributions:

A	P(A)
0	0.4
1	0.6

B	P(B)
0	0.7
1	0.3

C	P(C)
0	0.1
1	0.9

Joint:

A	B	C	P(A,B,C)
0	0	0	0.028
0	0	1	0.252
0	1	0	0.012
0	1	1	0.108
1	0	0	0.042
1	0	1	0.378
1	1	0	0.018
1	1	1	0.162

This reduces representation size!

Note: it is hard to "read" independence from the joint distribution.
We can "test" for it, however.

Conditional Independence

- X, Y independent given Z
  - $p(X=x, Y=y | Z=z) = p(X=x | Z=z)\, p(Y=y | Z=z)$ for all x,y,z
  - Equivalent: $p(X | Y,Z) = p(X | Z)$ or $p(Y | X,Z) = p(Y | Z)$ (if all > 0)
  - Intuition: X has no additional info about Y beyond Z's

- Example

X = height	p(height | reading, age) = p(height | age)
Y = reading ability	p(reading | height, age) = p(reading | age)
Z = age	

Height and reading ability are dependent (not independent), but are conditionally independent given age

Conditional Independence

- X, Y independent given Z
  - p(X=x,Y=y|Z=z) = p(X=x|Z=z) p(Y=y|Z=z) for all x,y,z
  - Equivalent: p(X|Y,Z) = p(X|Z) or p(Y|X,Z) = p(Y|Z)
  - Intuition: X has no additional info about Y beyond Z's

- Example: Dentist

$(T \perp\!\!\!\perp D \mid C)?$

Is T conditionally independent of C given D?

Again, hard to "read" from the joint probabilities; only from the conditional probabilities.

Like independence, reduces representation size!

Joint prob:

T	D	C	P(T,D,C)
0	0	0	0.576
0	0	1	0.008
0	1	0	0.144
0	1	1	0.072
1	0	0	0.064
1	0	1	0.012
1	1	0	0.016
1	1	1	0.108

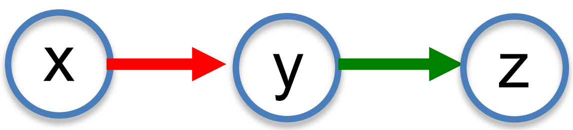
Conditional prob:

T	D	C	P(T|D,C)
0	0	0	0.90
0	0	1	0.40
0	1	0	0.90
0	1	1	0.40
1	0	0	0.10
1	0	1	0.60
1	1	0	0.10
1	1	1	0.60

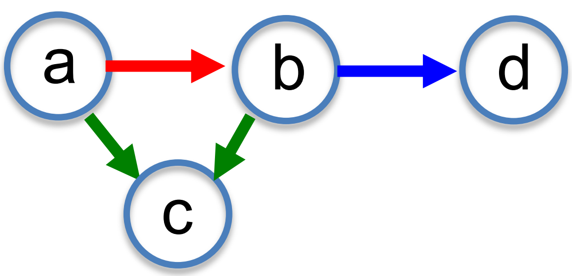
Bayesian networks

- Directed graphical model
- Nodes associated with variables
- "Draw" independence in conditional probability expansion
  - Parents in graph are the RHS of conditional

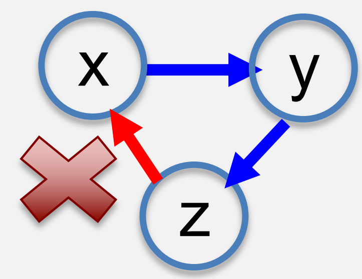
- Ex: $p(x, y, z) = p(x)\ p(y \mid x)\ p(z \mid y)$



- Ex $p(a, b, c, d) = p(a)\ p(b \mid a)\ p(c \mid a, b)\ p(d \mid b)$



Graph must be ***acyclic***



Corresponds to an order over the variables (chain rule)

Example adapted from Pearl

Example

- Consider the following 5 binary variables:
  - B = a burglary occurs at your house
  - E = an earthquake occurs at your house
  - A = the alarm goes off
  - W = Watson calls to report the alarm
  - H = Mrs. Hudson calls to report the alarm

  - What is P(B | H=1, W=1) ? (for example)

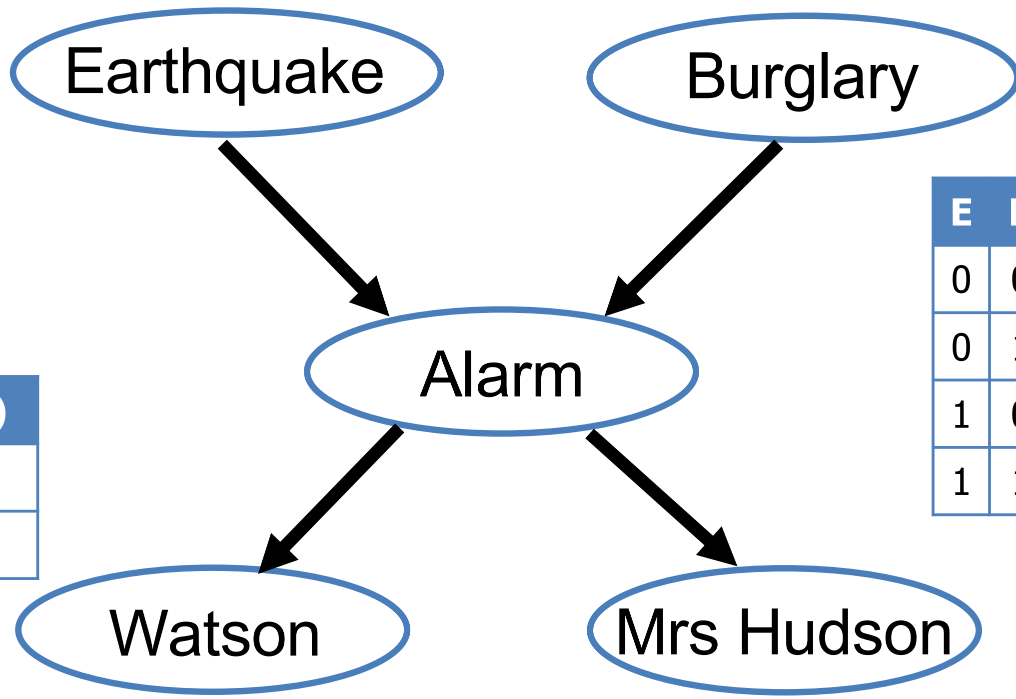
  - We can use the full joint distribution to answer this question
    - Requires $2^5 = 32$ probabilities

    - Can we use prior domain knowledge to come up with a Bayesian network that requires fewer probabilities?

Constructing a Bayesian network

- Given $p(W, H, A, E, B) = p(E)\ p(B)\ p(A|E, B)\ p(W|A)\ p(H|A)$
- Define probabilities: 1 + 1 + 4 + 2 + 2
- Where do these come from?
  - Expert knowledge; estimate from data; some combination

P(E)
0.002



P(B)
0.001

E	B	P(A|E,B)
0	0	0.001
0	1	0.29
1	0	0.94
1	1	0.95

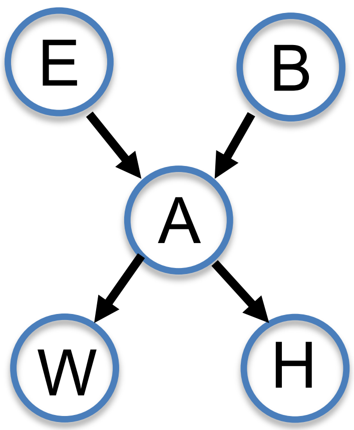
"CPT" = conditional probablility table

A	P(W|A)
0	0.05
1	0.90

A	P(H|A)
0	0.01
1	0.70

Constructing a Bayesian network

- Joint distribution



Full joint distribution:
2^5 = 32 probabilities

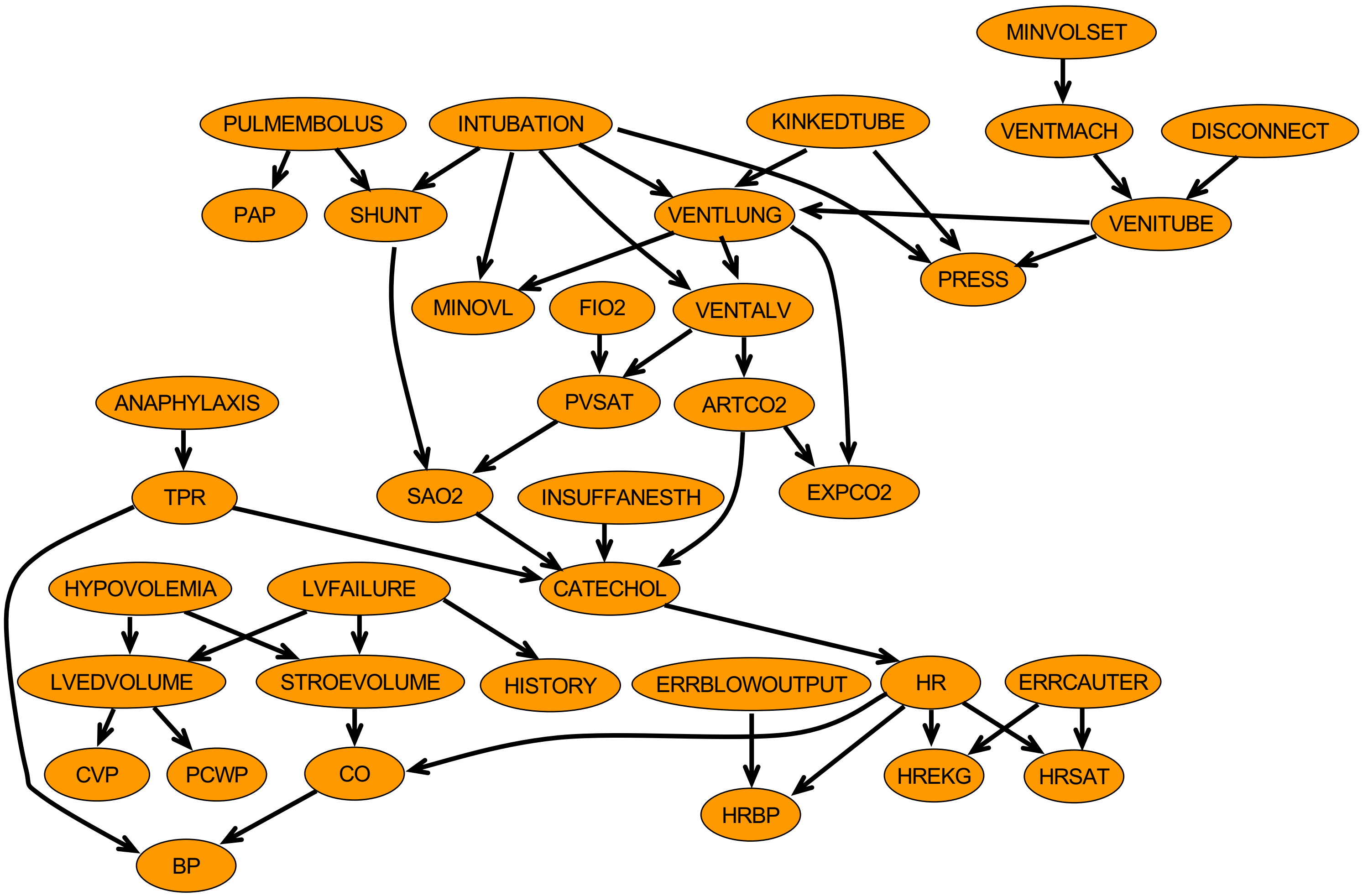
Structured distribution:
specify 10 parameters

E	B	A	W	H	P(…)
0	0	0	0	0	.93674
0	0	0	0	1	.00133
0	0	0	1	0	.00005
0	0	0	1	1	.00000
0	0	1	0	0	.00003
0	0	1	0	1	.00002
0	0	1	1	0	.00003
0	0	1	1	1	.00000
0	1	0	0	0	.04930
0	1	0	0	1	.00007
0	1	0	1	0	.00000
0	1	0	1	1	.00000
0	1	1	0	0	.00027
0	1	1	0	1	.00016
0	1	1	1	0	.00025
0	1	1	1	1	.00000

E	B	A	W	H	P(…)
1	0	0	0	0	.00946
1	0	0	0	1	.00001
1	0	0	1	0	.00000
1	0	0	1	1	.00000
1	0	1	0	0	.00007
1	0	1	0	1	.00004
1	0	1	1	0	.00007
1	0	1	1	1	.00000
1	1	0	0	0	.00050
1	1	0	0	1	.00000
1	1	0	1	0	.00000
1	1	0	1	1	.00000
1	1	1	0	0	.00063
1	1	1	0	1	.00037
1	1	1	1	0	.00059
1	1	1	1	1	.00000

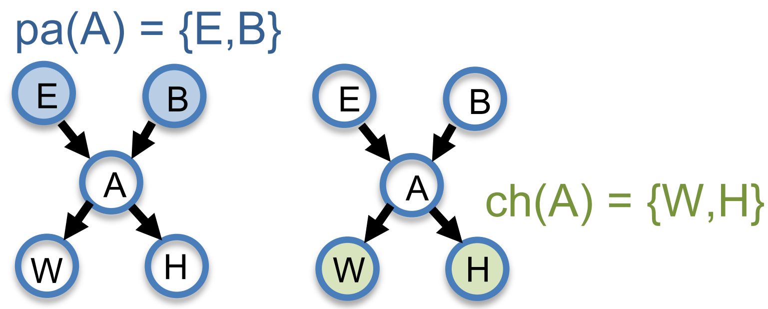
Alarm network [Beinlich et al., 1989]

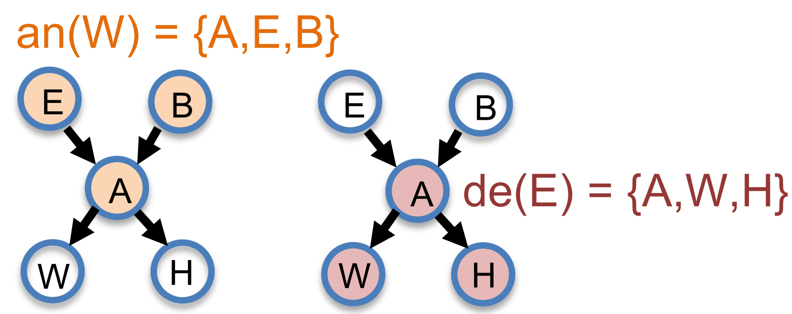
The “alarm” network: 37 variables, 509 parameters (rather than $2^{37} = 10^{11}$!)



Some terminology

- Parents & Children
  - Parents   pa(A) = {E,B}
  - Children ch(A) = {W,H}
- Ancestors & Descendants
  - Ancestors an(W) = {A,E,B}
  - Descendants de(E) = {A,W,H}
- Roots & Leaves
- Paths
  - Directed paths, undirected paths





Graphical models

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains (we'll assume discrete)

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*

Example:

$A \in \{0,1\}$

$B \in \{0,1\}$

$C \in \{0,1\}$

$f_{AB}(A,B), \quad f_{BC}(B,C)$

The *combination operator* defines an overall function from the individual factors,

e.g., "*" : $F(A,B,C) = f_{AB}(A,B) \cdot f_{BC}(B,C)$

Notation:

Discrete X_i : values called "states"

"Tuple" or "configuration": states taken by a set of variables

"Scope" of f: set of variables that are arguments to a factor f

often index factors by their scope, e.g., $f_\alpha(X_\alpha), \quad X_\alpha \subseteq X$

Canonical forms

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*

Typically either multiplication or summation; mostly equivalent:

$$f_\alpha(X_\alpha) \geq 0$$

$$F(X) = \prod_\alpha f_\alpha(X_\alpha)$$

log / exp

$$\theta_\alpha(X_\alpha) = \log f_\alpha(X_\alpha) \in \mathbb{R}$$

$$\theta(X) = \log F(x) = \sum_\alpha \theta_\alpha(X_\alpha)$$

Product of nonnegative factors
(probabilities, 0/1, etc.)

Sum of factors
(costs, utilities, etc.)

Graphical visualization

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

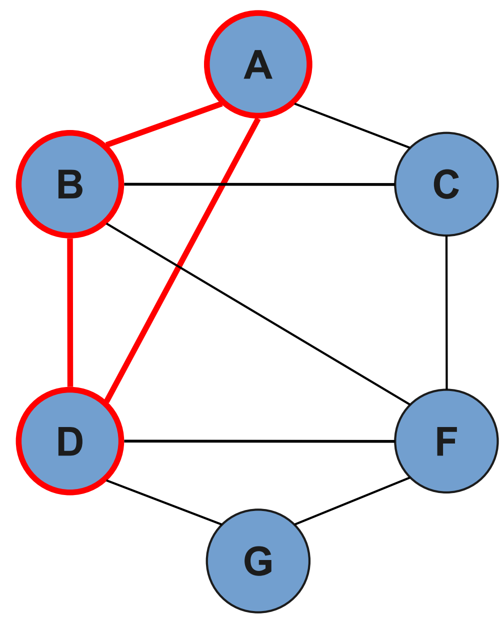
$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

Primal graph:

variables ⟷ nodes

factors ⟷ cliques

$$p(A,B,C,D,F,G) = f_1(A,B,D) \cdot f_2(D,F,G) \cdot f_3(B,C,F) \cdot f_4(A,C)$$



Outline

- Graphical Models
- **Inference Tasks**
- Variable Elimination
- Tree Decomposition
- Variable Orderings
- Learning from Data

Inference

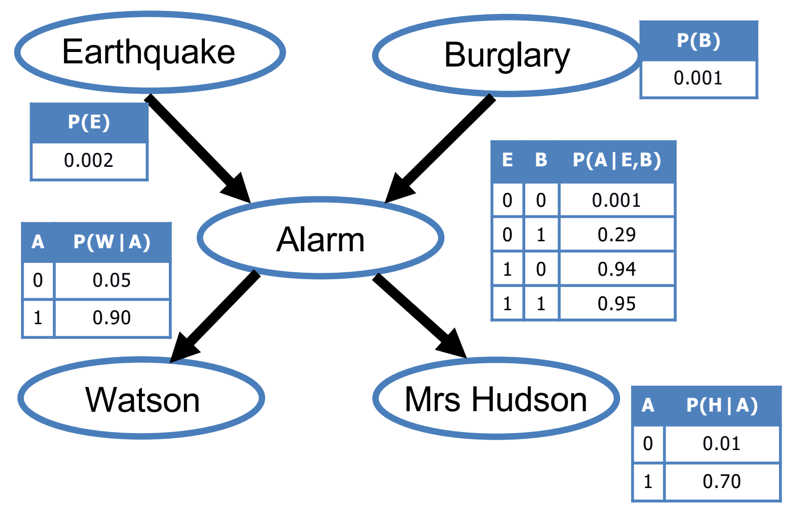
Enable us to answer **queries** about our model

- Some probabilities are directly accessible
- Some are only **implicit**, and require computation

p(B=1) = .001
Explicitly in model parameters

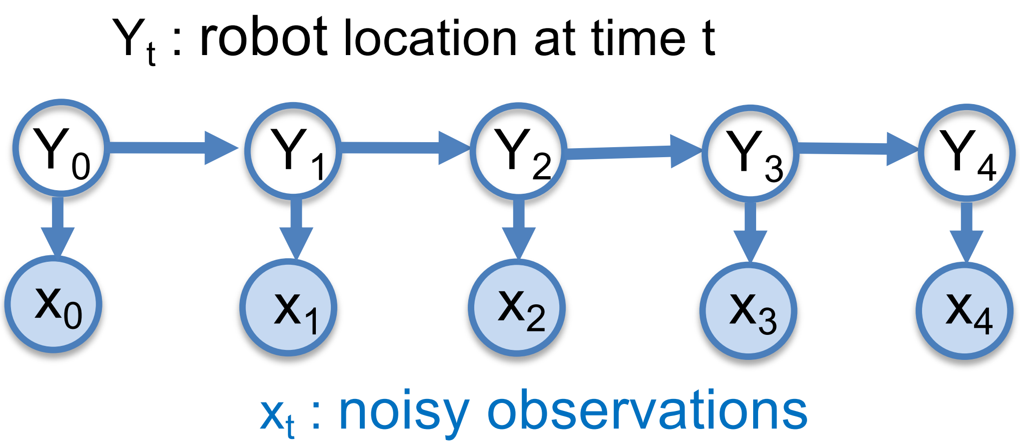
p(A=1) = ?
Implicit only:
p(A=1|E=0,B=0) p(E=0) p(B=0) +
p(A=1|E=1,B=0) p(E=1) p(B=0) + …

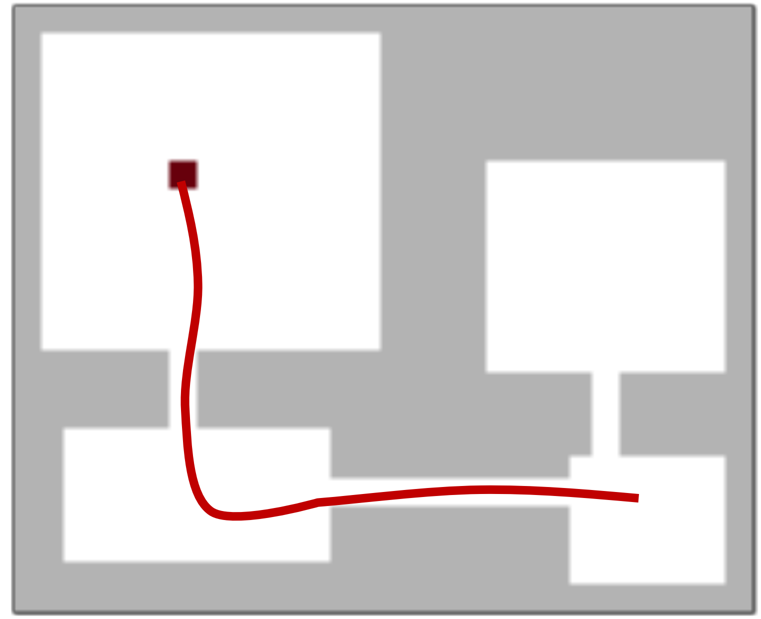
p(W=1) = ?
Implicit:
p(W=1|A=0) p(A=0) + p(W=1|A=1) p(A=1)
p(A) =? (may need to compute recursively!)



Types of queries

Ex: Robot position over time





Summation Query (marginal probabilitiies, probability of evidence):

$$p(Y_t|x_0 \dots x_t) \propto \sum_{y_{t-1}} \dots \sum_{y_0} p(Y_t, x_t, y_{t-1}, \dots, x_0)$$

What do my model and observations tell me about my uncertainty?

Maximization Query (MAP: maximum a posteriori estmation):

$$\mathbf{y}^* = \arg \max_{y_0, \dots, y_t} p(y_t, x_t, y_{t-1}, \dots, x_0)$$

What is the most probable value of the unobserved variables?

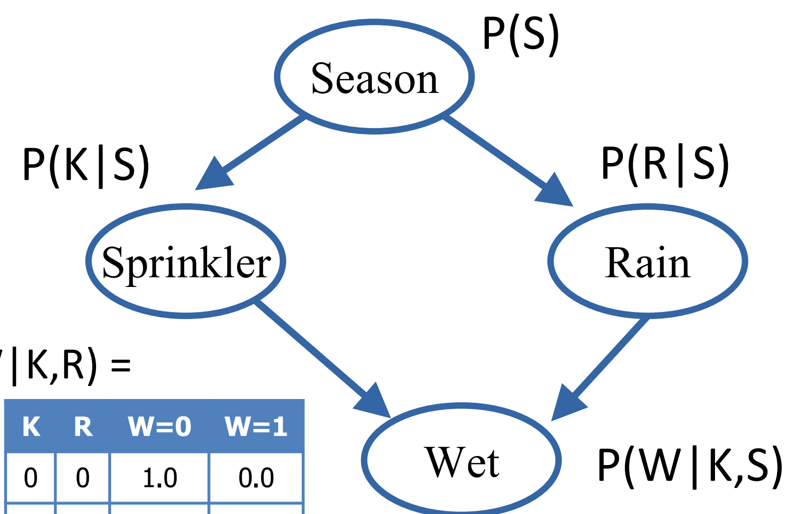
Causal Bayesian networks

- Typical BNs capture conditional independence
- May not correspond to causation; but if so:

Causal Effect Query (“Intervention”):

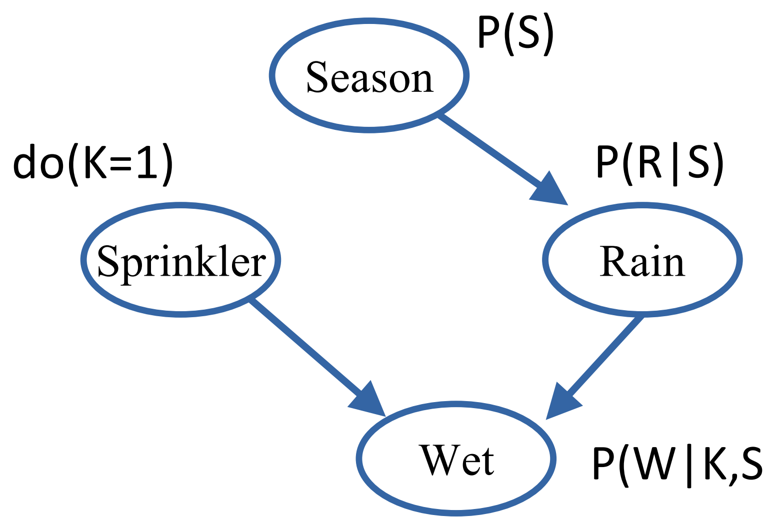
$$p(W|\text{do}(K=1))$$

What is the probability when we intervene to turn on the sprinkler?



P(W|K,R) =

K	R	W=0	W=1
0	0	1.0	0.0
0	1	0.2	0.8
1	0	0.1	0.9
1	1	0.01	0.99



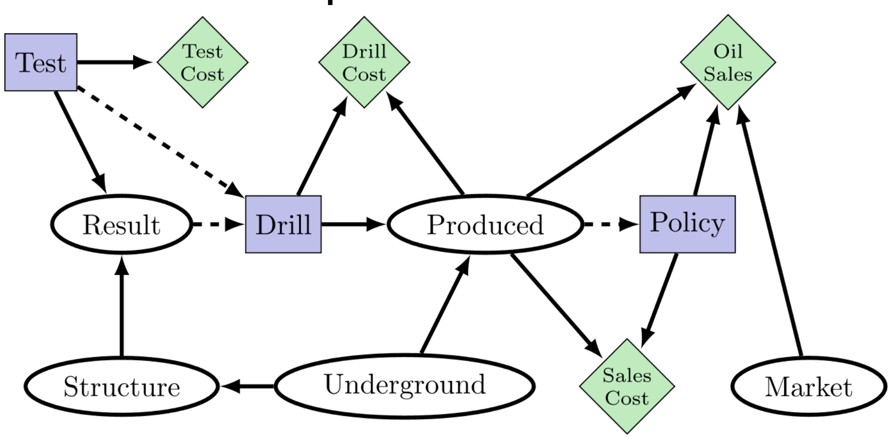
Influence diagrams

Random variables, plus **actions** (policy) and **utilities** (outcome values)

Maximum Expected Utility Query:

What actions should I take in a given situation?
What is the expected value of my policy over the actions?

The "oil wildcatter" problem:



e.g., [Raiffa 1968; Shachter 1986]

Chance variables: $X = x_1, \ldots, x_n$

Decision variables: $D = d_1, \ldots d_m$

CPDs for chance variables: $P_i = P(x_i | x_{\text{pa}_i}),$

Reward components: $r = \{r_1, \ldots, r_j\}$

Utility function: $u(X) = \sum_i r_i(X)$

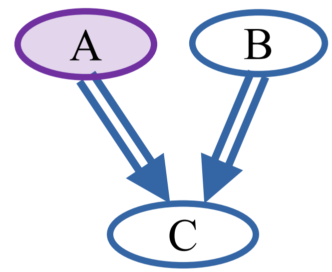
Structural Causal Models

Deterministic mechanisms involving (random) underlying causes

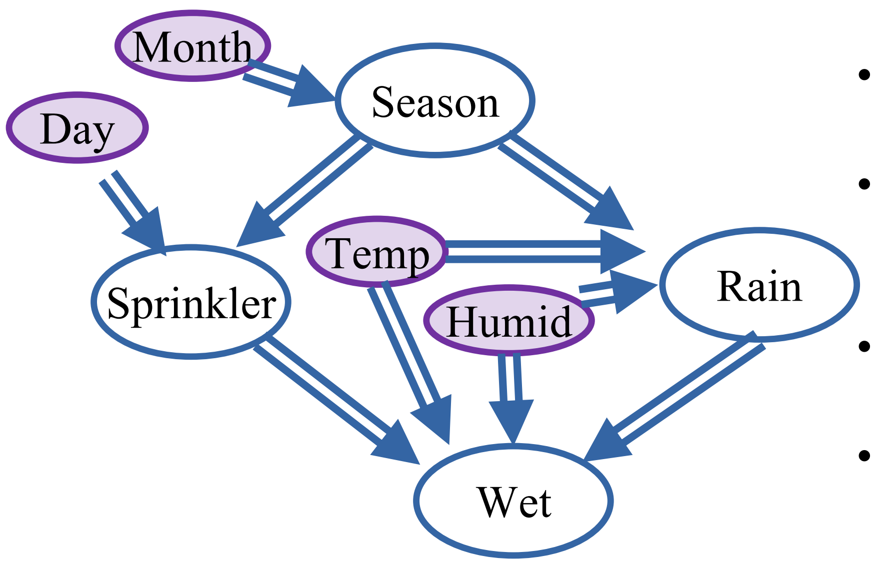
Unobservable random variables: {A}

Observable variables: {B,C}

Deterministic mechanism: ⟹ C = f(A,B)



Ex: Sprinkler



- p(S): season a function of (unobserved) month
- p(K|S): sprinkler on due to watering schedule: randomness in K due to (unobserved) day of week
- p(R|S) caused by humidity and temperature
- p(W|R,K) also caused by humidity and temperature (effects of evaporation, etc.)

Structural Causal Models

Deterministic mechanisms involving (random) underlying causes

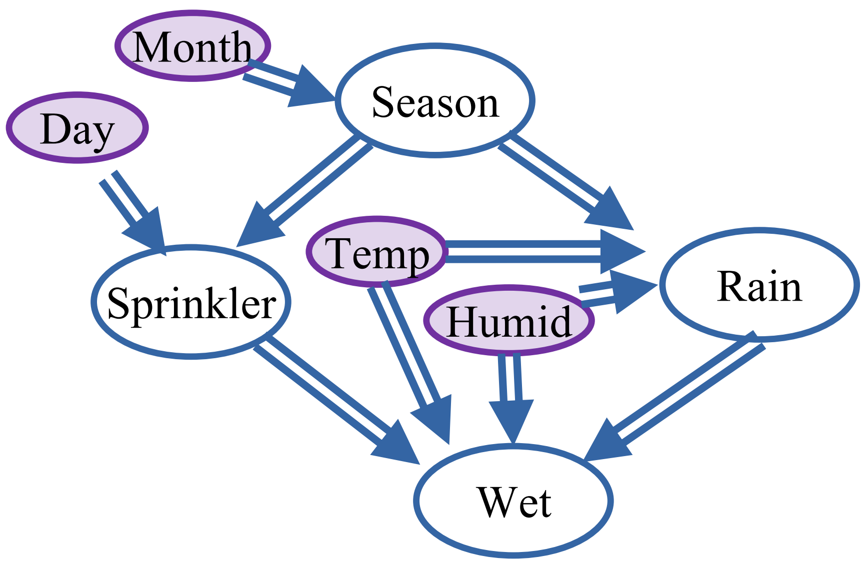
Counterfactual Query:

Probability of an event in contradiction with the observations

*What **would have happened** if the sprinkler had been turned off?*

Requires that we transfer information about random outcomes that happened, to a different setting

Ex: Sprinkler



Observe the sprinkler is on & grass is wet: (K=1,W=1)

What is the probability it would still be wet if we had turned the sprinkler off?

Observing K=1 tells us it is more likely to be summer;
Observing K=1,W=1 tells us it is not too hot & dry.

Then, apply this knowledge to compute the counterfactual:

$$p(W_{K=0} \mid K = 1, W = 1)$$

Why graphical models?

Combine domain knowledge with learning and data

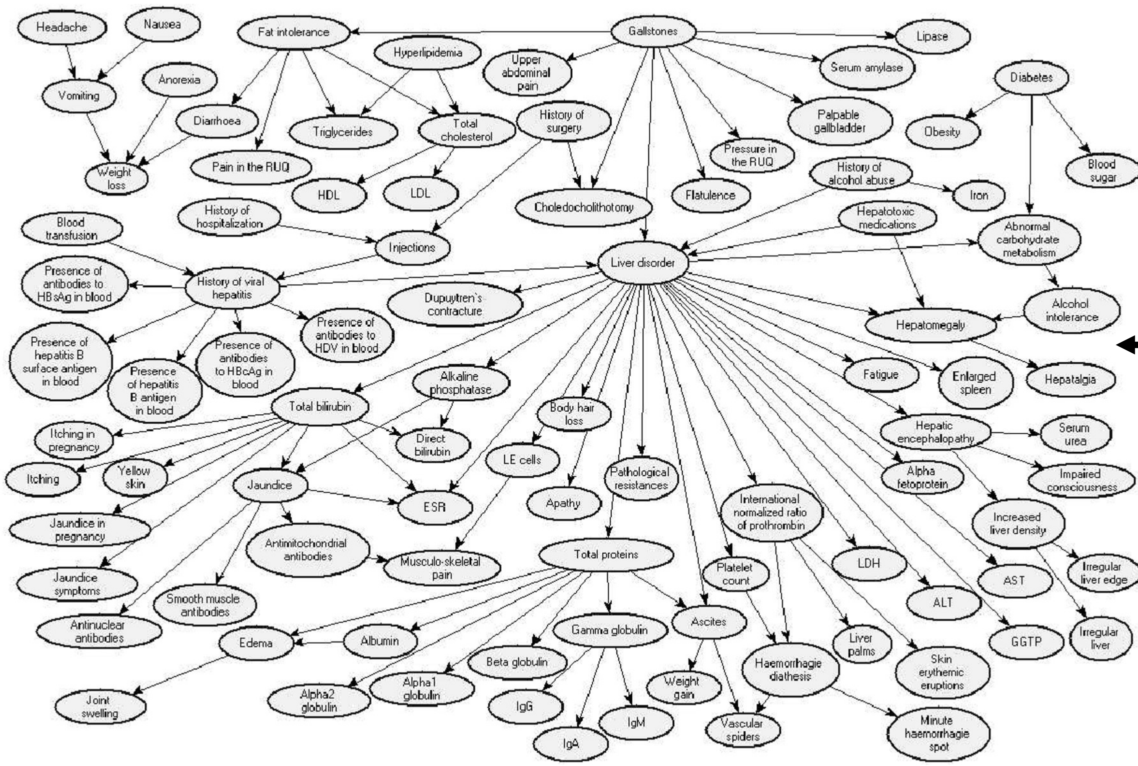
- Domain knowledge
  - Problem structure: potential causation or interactions
  - Model parameters: known dependency mechanisms, probabilities

- Learning and data
  - Identify (in)dependence from data
  - Estimate model parameters to explain observations

- Scalable and Composable
  - Models over large systems may be composed of smaller parts
  - Efficient representation allows learning from relatively few data

Ex: Medical diagnosis

Diagnosing liver disease [Onisko et al., 1999]



Queries:

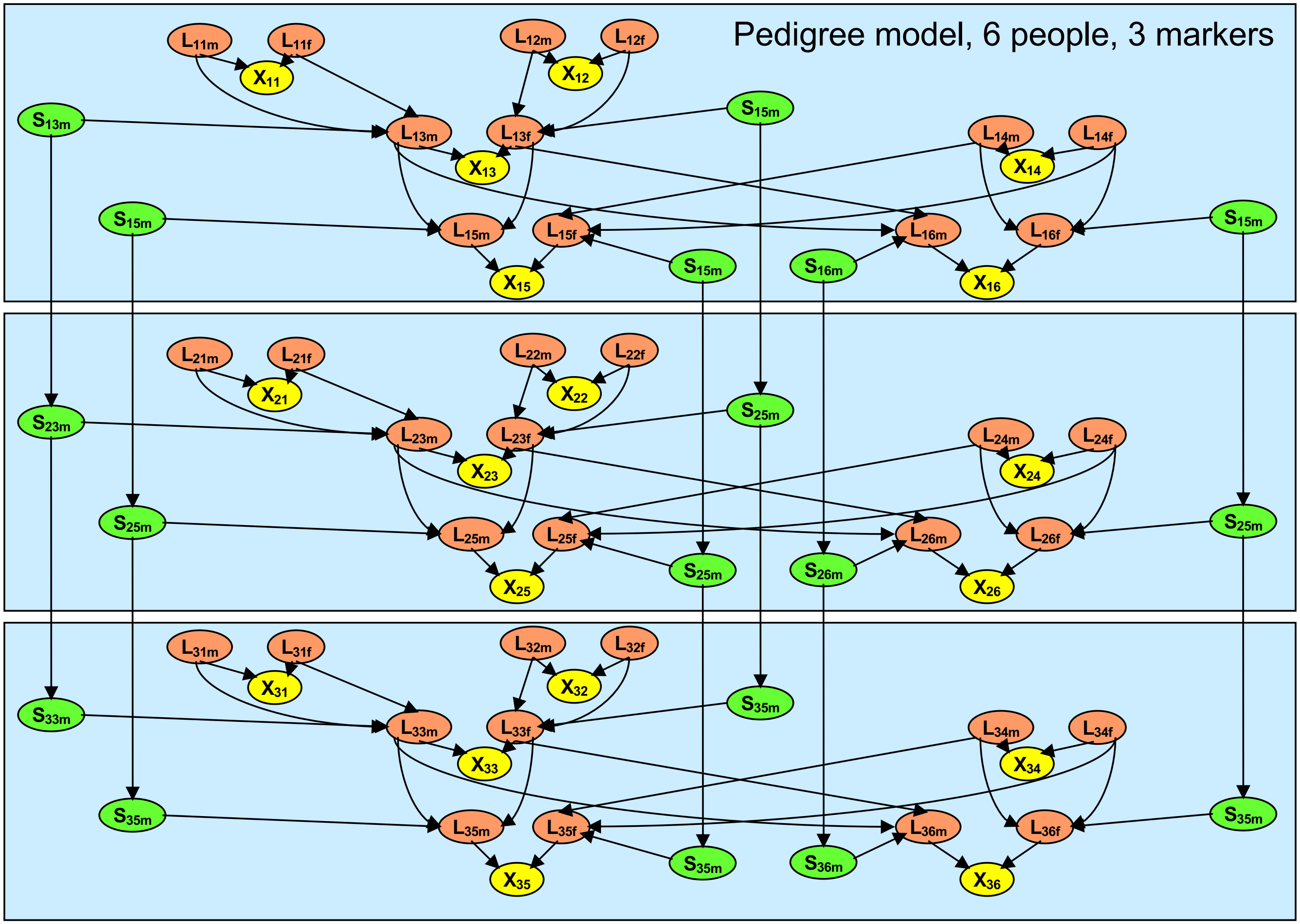
- Prediction
- Diagnosis, explanation
- Situation assessment
- Planning, decision making
- Counterfactual reasoning

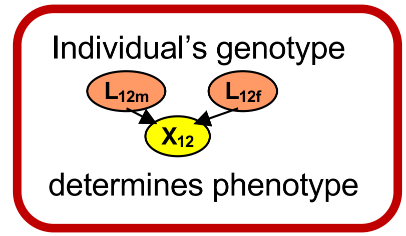
Automated Reasoning:

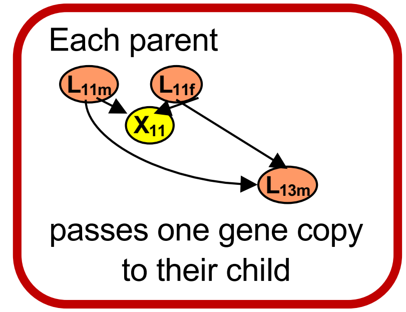
- Develop methods to answer these questions
- Learning models from experts and data

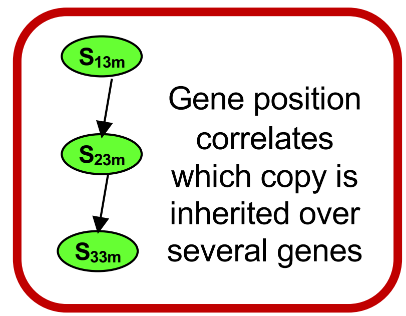
Ex: Model composability

Large models may be defined by many repeated, interrelated structures





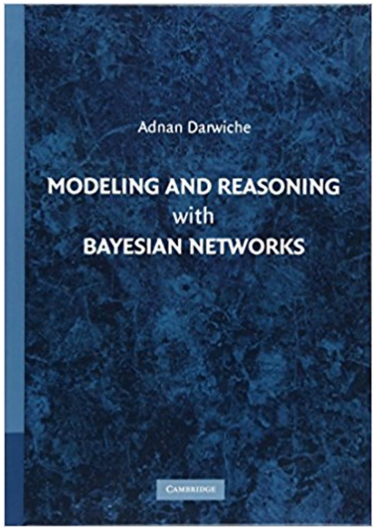


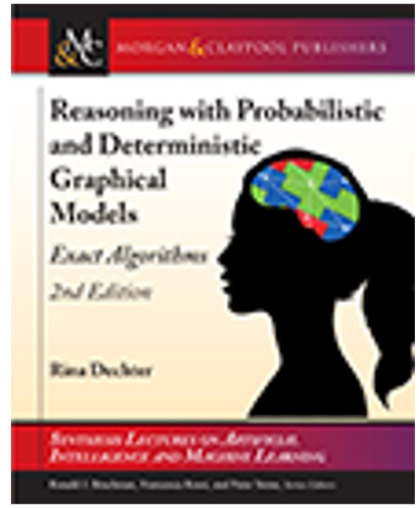


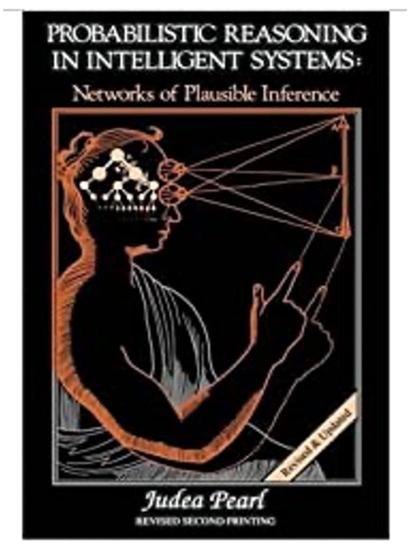
Domains for graphical models

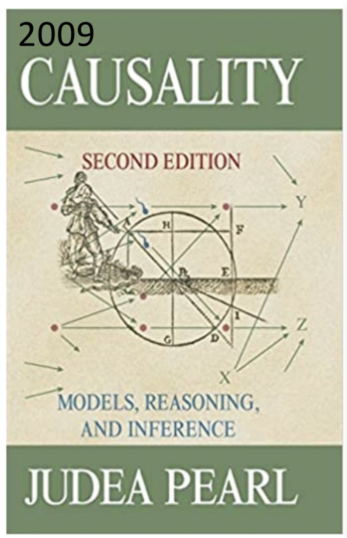
- Natural Language processing
  - Information extraction, semantic parsing, translation, topic models, …
- Computer vision
  - Object recognition, scene analysis, segmentation, tracking, …
- Computational biology
  - Pedigree analysis, protein folding and binding, sequence matching, …
- Networks
  - Webpage link analysis, social networks, communications, citations, ….
- Robotics
  - Planning & decision making

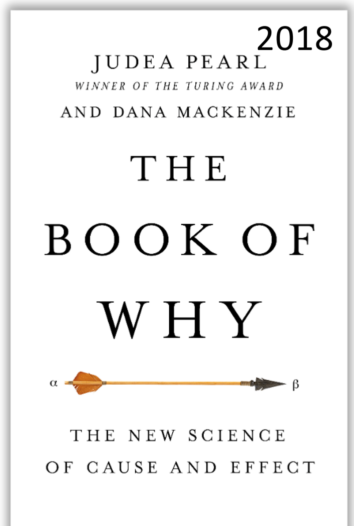
Books on Graphical Models & Causality

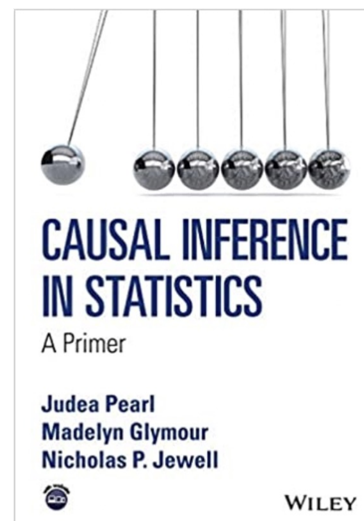








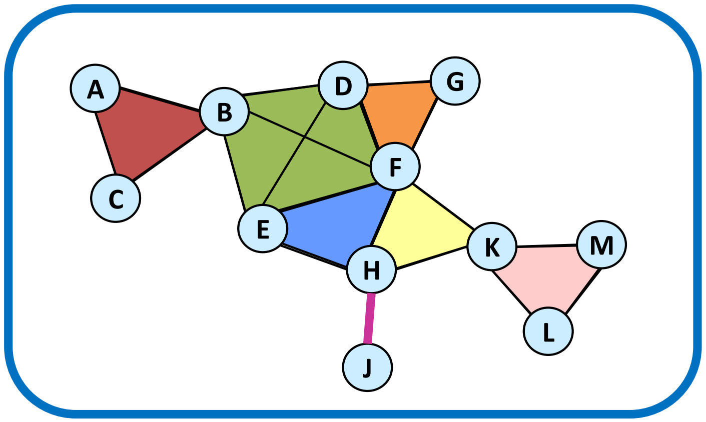




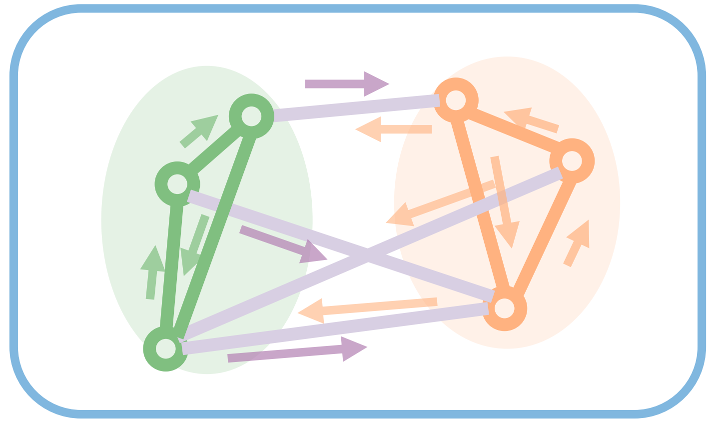
(intervention & counterfactuals)

Outline of Lectures

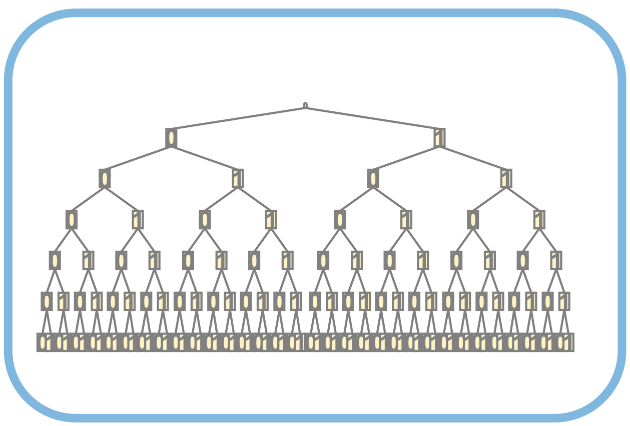
Class 1: Introduction & Inference



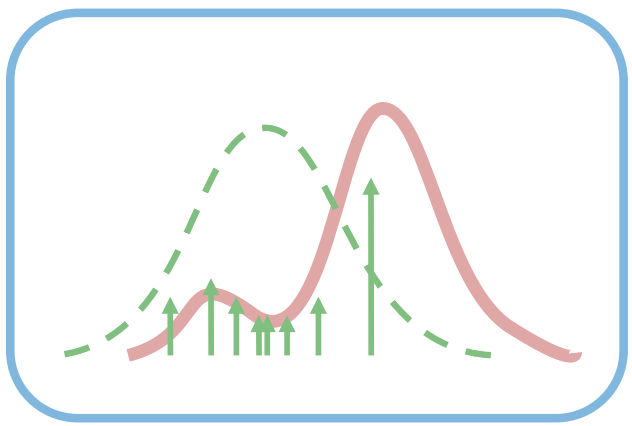
Class 2: Bounds & Variational Methods



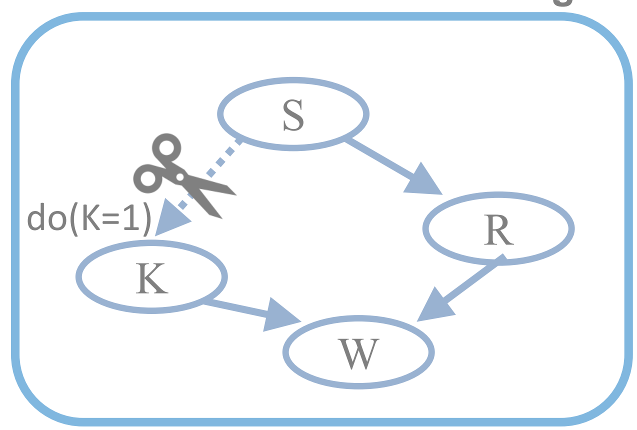
Class 3: Search Methods



Class 4: Monte Carlo Methods



Class 5: Causal Reasoning



Outline

- Graphical Models
- Inference Tasks
- **Variable Elimination**
- Tree Decomposition
- Variable Orderings
- Learning from Data

Inference

- Take information (model, observations)
  - Implicitly defines a joint / conditional joint distribution
- Inference: what does that information mean?
  - How do other probabilities change? (Conditional) marginals
  - How likely was our observation? Probabiliity of evidence
  - Predict the value of the other variables? MPE / MAP estimates
- Tasks

"summation"

$$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha) \qquad p(X_i = x_i) = Z^{-1} \sum_{x \sim x_i} \prod_\alpha f_\alpha(x_\alpha)$$

"maximization"

$$x^* = \arg\max_x \prod_\alpha f_\alpha(x_\alpha) \qquad f^* = f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$$

A simple example

- Suppose we have two factors: $f(X) = f_{12}(X_1, X_2)\, f_{23}(X_2, X_3)$

- To compute the partition function (sum):

$$Z = \sum_{x_1, x_2, x_3} f(x_1, x_2, x_3) = f(0,0,0) + f(0,0,1) + f(0,0,2) + f(0,1,0) + \dots$$
$$+ f(1,0,0) + f(1,0,1) + f(1,0,2) + f(1,1,0) + \dots$$

  – Use the factorization of f(x):

$$Z = f_{12}(0,0)\, f_{23}(0,0) + f_{12}(0,0)\, f_{23}(0,1) + f_{12}(0,0)\, f_{23}(0,2) + f_{12}(0,1)\, f_{23}(1,0) + \dots$$
$$+ f_{12}(1,0)\, f_{23}(0,0) + f_{12}(1,0)\, f_{23}(0,1) + f_{12}(1,0)\, f_{23}(0,2) + f_{12}(1,1)\, f_{23}(1,0) + \dots$$

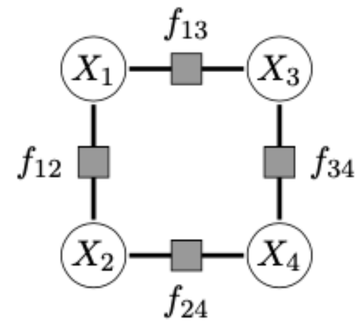
  and apply the distributive rule:

$$= f_{12}(0,0) \Big(f_{23}(0,0) + f_{23}(0,1) + f_{23}(0,2) \Big) + f_{12}(0,1) \Big(f_{23}(1,0) + \dots$$
$$+ f_{12}(1,0) \Big(f_{23}(0,0) + f_{23}(0,1) + f_{23}(0,2) \Big) + f_{12}(1,1) \Big(f_{23}(1,0) + \dots$$

We can pre-compute and re-use these terms in the sum!

$$\lambda(x_2) = \Big(\sum_{x_3} f_{23}(x_2, x_3) \Big) \qquad Z = \sum_{x_1, x_2} f_{12}(x_1, x_2)\, \lambda(x_2)$$

Variable Elimination



Product of factors:

$$p(X_1, X_2, X_3, X_4) = \frac{1}{Z} f_{12}(X_1, X_2) f_{13}(X_1, X_3) f_{24}(X_2, X_4) f_{34}(X_3, X_4).$$

Compute:

$$Z = \sum_{x_4} \sum_{x_3} \sum_{x_2} \sum_{x_1} f_{34}(x_3, x_4) f_{24}(x_2, x_4) f_{12}(x_1, x_2) f_{13}(x_1, x_3),$$

Collect terms involving x_1, then x_2, and so on:

$$Z = \sum_{x_4} \sum_{x_3} f_{34}(x_3, x_4) \sum_{x_2} f_{24}(x_2, x_4) \sum_{x_1} f_{12}(x_1, x_2) f_{13}(x_1, x_3),$$

"Bucket elimination":

$$\lambda_1(x_2, x_3) = \sum_{x_1} f_{12}(x_1, x_2) f_{13}(x_1, x_3),$$

Collect all factors with x_1 in a "bucket"

$$\lambda_2(x_3, x_4) = \sum_{x_2} f_{24}(x_2, x_4) \lambda_1(x_2, x_3),$$

Collect all remaining factors with x_2

$$\lambda_3(x_4) = \sum_{x_3} f_{34}(x_3, x_4) \lambda_2(x_3, x_4),$$

Place intermediate calculations in bucket of their earliest argument

$$Z = \sum_{x_4} \lambda_3(x_4),$$

Combination of factors

A	B	f(A,B)
b	b	0.4
b	g	0.1
g	b	0
g	g	0.5

●

B	C	f(B,C)
b	b	0.2
b	g	0
g	b	0
g	g	0.8

A	B	C	f(A,B,C)
b	b	b	0.1
b	b	g	0
b	g	b	0
b	g	g	0.08
g	b	b	0
g	b	g	0
g	g	b	0
g	g	g	0.4

= 0.1 x 0.8

Elimination in a factor

A	B	f(A,B)
b	b	4
b	g	6
b	r	1
g	b	2
g	g	6
g	r	3
r	b	1
r	g	1
r	r	6

Elim = sum

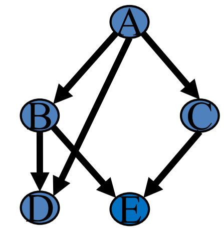
Elim(**f**,**B**)

A	g(A)
b	11
g	11
r	8

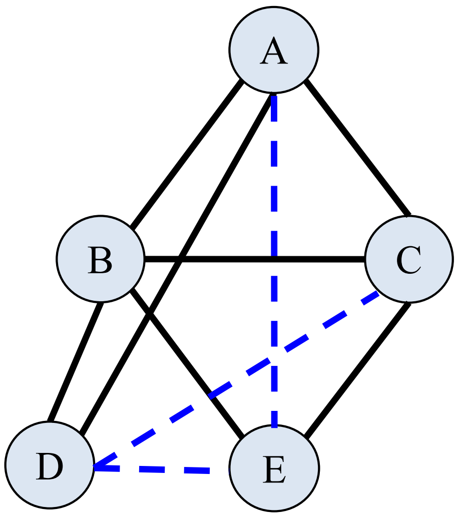
Elim(**g**,**A**)

$h_{\varnothing}$
30

Belief updating



- p(X | Evidence) = ?



“primal” graph

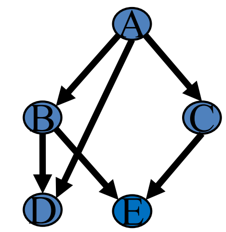
$$p(A|E=0)$$
$$\propto p(A, E=0)$$
$$= \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|b,A)\, p(e|b,c)\, \mathbb{1}[e=0]$$

$$p(A) \sum_{e} \sum_{d} \sum_{c} p(c|A)\, \mathbb{1}[e=0] \underbrace{\sum_{b} p(b|A)\, p(d|b,A)\, p(e|b,c)}_{\lambda_{B\to C}(a,d,c,e)}$$

Variable Elimination

Bucket elimination

Algorithm *BE-bel* [Dechter 1996]



$$p(A|E=0) = \alpha \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|A,b)\, p(e|b,c)\, \mathbb{1}[e=0]$$

$\sum_b \prod$ ← Elimination & combination operators

bucket B: $p(b|A)\ p(d|b,A)\ p(e|b,c)$

bucket C: $p(c|A)\quad \lambda_{B\to C}(A,d,c,e)$

bucket D: $\lambda_{C\to D}(A,d,e)$

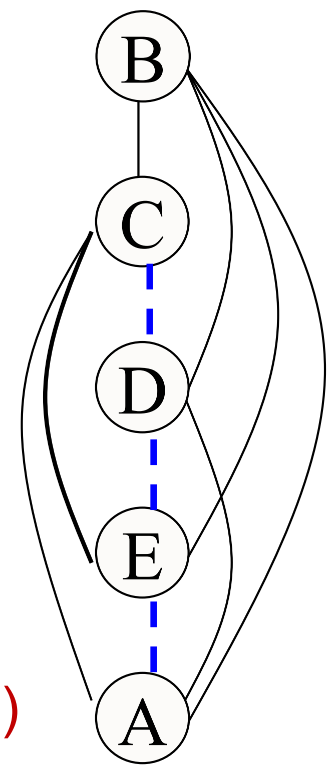
bucket E: $\mathbb{1}[E=0]\ \lambda_{D\to E}(A,e)$

bucket A: $p(A)\quad \lambda_{E\to A}(A)$

$p(E=0)$

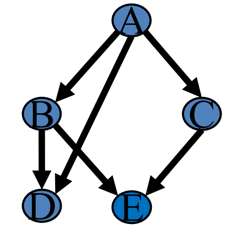
$$p(A|E=0) = p(A,E=0)\,/\,p(E=0)$$

W*=4
"induced width"
(max clique size)



Bucket elimination

Algorithm *BE-bel* [Dechter 1996]



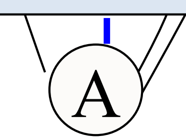
$$p(A|E=0) = \alpha \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|A,b)\, p(e|b,c)\, \mathbb{1}[e=0]$$

$\sum_b \prod$ ← Elimination & combination operators

Time and space exponential in the induced-width / treewidth

bucket A: $p(A)$ $\lambda_{E \to A}(A)$

induced width (max clique size)



$p(E=0)$

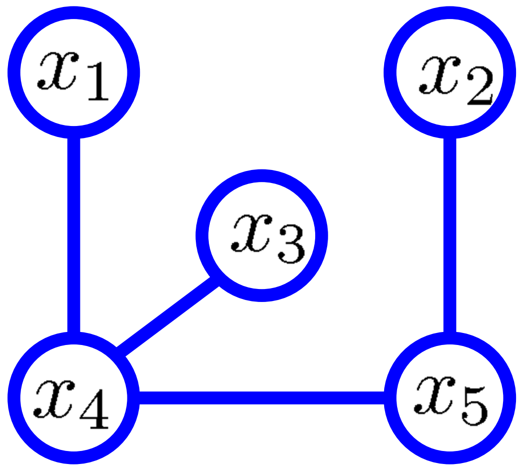
$p(A|E=0) = p(A, E=0) \,/\, p(E=0)$

Variable elimination in trees

(Use distributive rule to calculate efficiently:)

$$\max_{x_1 \ldots x_5} f_{14}\ f_{25}\ f_{34}\ f_{45}$$

$$= \max_{x_2 \ldots x_5} f_{25}\ f_{34}\ f_{45} \Big[\max_{x_1} f_{14} \Big]$$



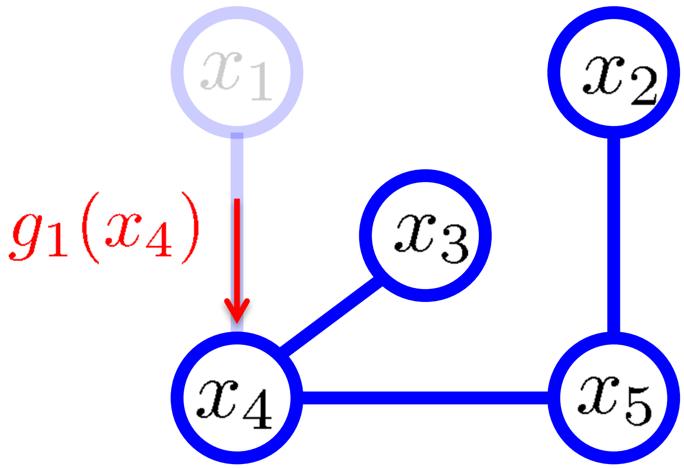
Variable elimination in trees

(Use distributive rule to calculate efficiently:)

$$\max_{x_1 \dots x_5} f_{14}\ f_{25}\ f_{34}\ f_{45}$$

$$= \max_{x_2 \dots x_5} f_{25}\ f_{34}\ f_{45} \Big[\max_{x_1} f_{14} \Big]$$

$$= \max_{x_3 \dots x_5} f_{34}\ f_{45}\ g_1(x_4)\ \Big[\max_{x_2} f_{25} \Big]$$



Variable elimination in trees

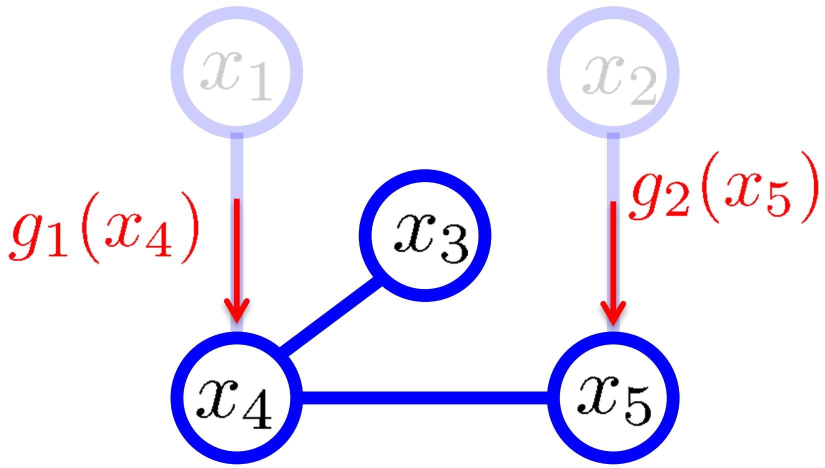
(Use distributive rule to calculate efficiently:)

$$\max_{x_1 \ldots x_5} f_{14}\ f_{25}\ f_{34}\ f_{45}$$

$$= \max_{x_2 \ldots x_5} f_{25}\ f_{34}\ f_{45} \Big[\max_{x_1} f_{14}\Big]$$

$$= \max_{x_3 \ldots x_5} f_{34}\ f_{45}\ g_1(x_4)\ \Big[\max_{x_2} f_{25}\Big]$$

$$= \max_{x_4 \ldots x_5} f_{45}\ g_1(x_4)\ g_2(x_5)\ \Big[\max_{x_3} f_{34}\Big]$$



Variable elimination in trees

(Use distributive rule to calculate efficiently:)

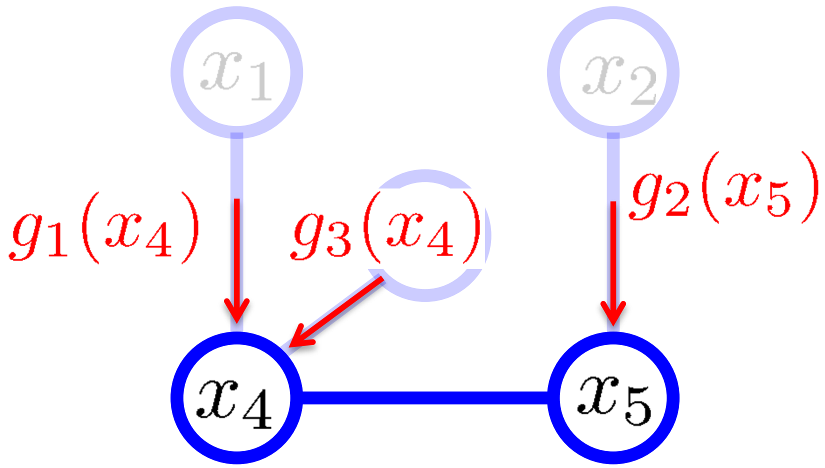
$$\max_{x_1 \ldots x_5} f_{14}\; f_{25}\; f_{34}\; f_{45}$$

$$= \max_{x_2 \ldots x_5} f_{25}\; f_{34}\; f_{45} \Big[\max_{x_1} f_{14}\Big]$$

$$= \max_{x_3 \ldots x_5} f_{34}\; f_{45}\; g_1(x_4)\; \Big[\max_{x_2} f_{25}\Big]$$

$$= \max_{x_4 \ldots x_5} f_{45}\; g_1(x_4)\; g_2(x_5)\; \Big[\max_{x_3} f_{34}\Big]$$

$$= \max_{x_5} \; g_2(x_5)\; \Big[\max_{x_4} f_{45}\; g_1(x_4)\; g_3(x_4)\Big]$$



Variable elimination in trees

(Use distributive rule to calculate efficiently:)

$$\max_{x_1 \ldots x_5} f_{14}\ f_{25}\ f_{34}\ f_{45}$$

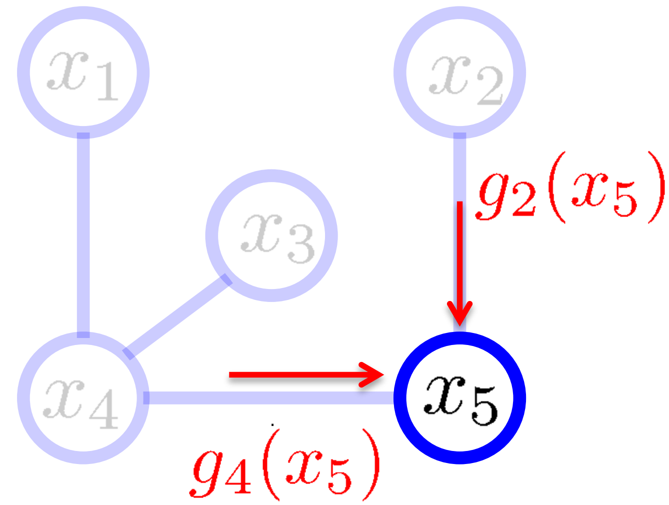
$$= \max_{x_2 \ldots x_5} f_{25}\ f_{34}\ f_{45} \Big[\max_{x_1} f_{14} \Big]$$

$$= \max_{x_3 \ldots x_5} f_{34}\ f_{45}\ g_1(x_4)\ \Big[\max_{x_2} f_{25} \Big]$$

$$= \max_{x_4 \ldots x_5} f_{45}\ g_1(x_4)\ g_2(x_5)\ \Big[\max_{x_3} f_{34} \Big]$$

$$= \max_{x_5}\ g_2(x_5)\ \Big[\max_{x_4} f_{45}\ g_1(x_4)\ g_3(x_4) \Big]$$

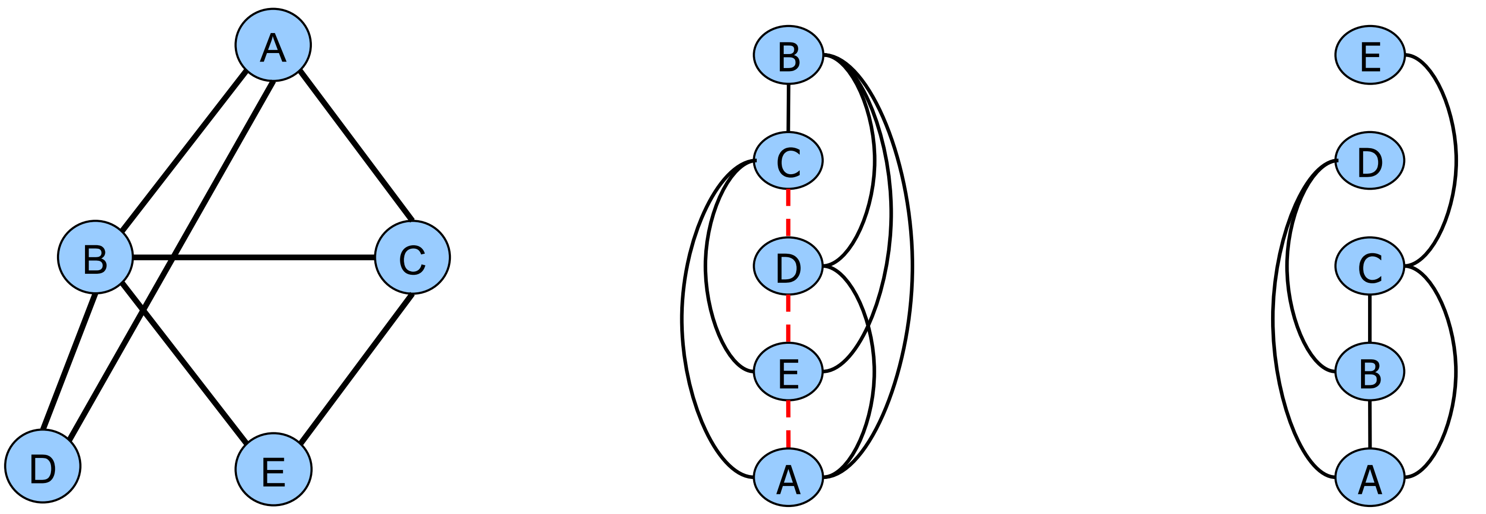
$$= \max_{x_5}\ g_2(x_5)\ g_4(x_5)$$



For trees:
Efficient elimination order (leaves to root);
computational complexity same as model size

Induced Width

- Width is the max number of parents in the ordered graph
- Induced-width is the width of the induced ordered graph: recursively connecting parents going from last node to first.
- Induced-width w*(d) is the max induced-width over all nodes in ordering d
- Induced-width of a graph, w* is the min w*(d) over all orderings d



primal graph $w^*(d_1) = 4$ $w^*(d_2) = 2$

Complexity of Bucket Elimination

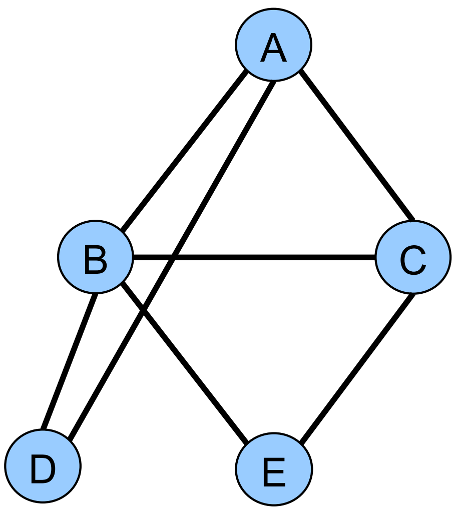
Bucket-Elimination is **time** and **space**

$$O(\,r\exp(w_d^*)\,)$$

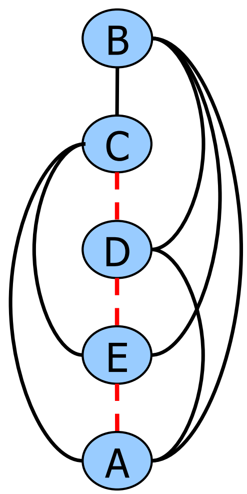
w_d^* : the induced width of the primal graph along ordering d

r = number of functions

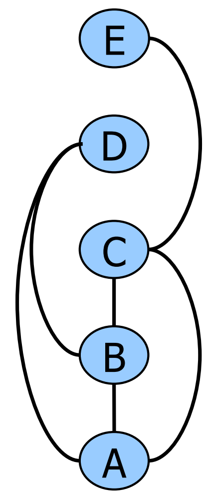
The effect of the ordering:



primal graph



$w^*(d_1) = 4$

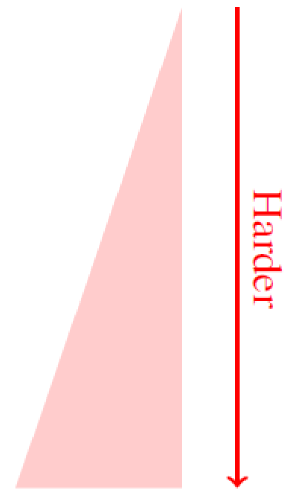


$w^*(d_2) = 2$

Finding smallest induced-width is hard!

Types of queries

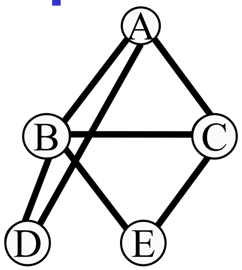
Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference: (e.g., causal effects)	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP): (optimal prediction)	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU): (e.g., decisions & planning)	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$



- **NP-hard**: exponentially many terms
- Difficulty (in part) due to restricted elimination orderings
- We will focus on **approximation** algorithms
  - **Anytime**: very fast & very approximate $\rightarrow$ Slower & more accurate

Finding MPE/MAP

Algorithm BE-mpe (Dechter 1996, Bertele and Briochi,



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

$$= \max_b \; p(b|a) \cdot p(d|b,a) \cdot p(e|b,c)$$

$\max_x \; \prod$

bucket B: $p(b|a) \;\; p(d|b,a) \;\; p(e|b,c)$

bucket C: $p(c|a) \quad \lambda_{B\to C}(a,d,c,e)$

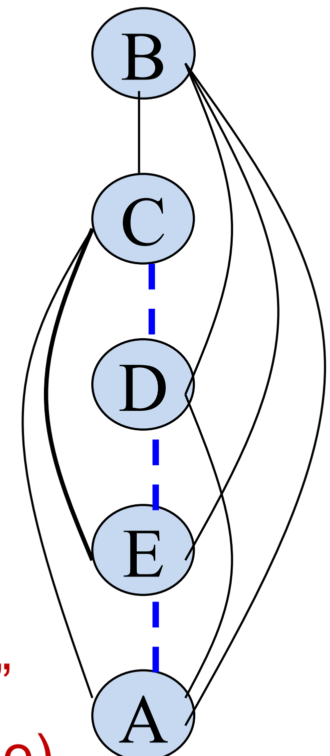
bucket D: $\lambda_{C\to D}(a,d,e)$

bucket E: $\mathbb{1}[e=0] \quad \lambda_{D\to E}(a,e)$

bucket A: $p(a) \quad \lambda_{E\to A}(a)$

OPT

W*=4
"induced width"
(max clique size)



Generating the optimal assignment

- Given BE messages, select optimum config in reverse order

$$\mathbf{b}^* = \arg\max_b \; p(b|a^*)\, p(d^*|b,a^*)\, p(e^*|b,c^*)$$

$$\mathbf{c}^* = \arg\max_c \; p(c|a^*)\, \lambda_{B\to C}(a^*,c,d^*,e^*)$$

$$\mathbf{d}^* = \arg\max_d \; \lambda_{C\to D}(a^*,d,e^*)$$

$$\mathbf{e}^* = \arg\max_e \; \mathbb{1}[e=0]\, \lambda_{D\to E}(a^*,e)$$

$$\mathbf{a}^* = \arg\max_a \; p(a)\cdot \lambda_{E\to A}(a)$$

B: $p(b|a)\;\; p(d|b,a)\;\; p(e|b,c)$

C: $p(c|a)\quad \lambda_{B\to C}(a,c,d,e)$

D: $\lambda_{C\to D}(a,d,e)$

E: $\mathbb{1}[e=0]\quad \lambda_{D\to E}(a,e)$

A: $p(a)\;\; \lambda_{E\to A}(a)$

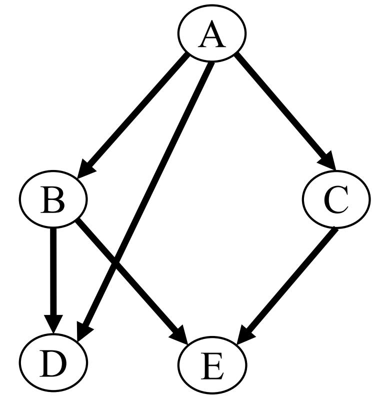
OPT = optimal value

Return optimal configuration (a*,b*,c*,d*,e*)

Outline

- Graphical Models
- Inference Tasks
- Variable Elimination
- **Tree Decompositions**
- Variable Orderings
- Learning from Data

Bucket Tree Elimination



$$p(A|E=0) = \alpha \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|A,b)\, p(e|b,c)\, \mathbb{1}[e=0]$$

E: $p(E|B,C)\mathbb{1}[E=0]$

D: $p(D|A,B)$

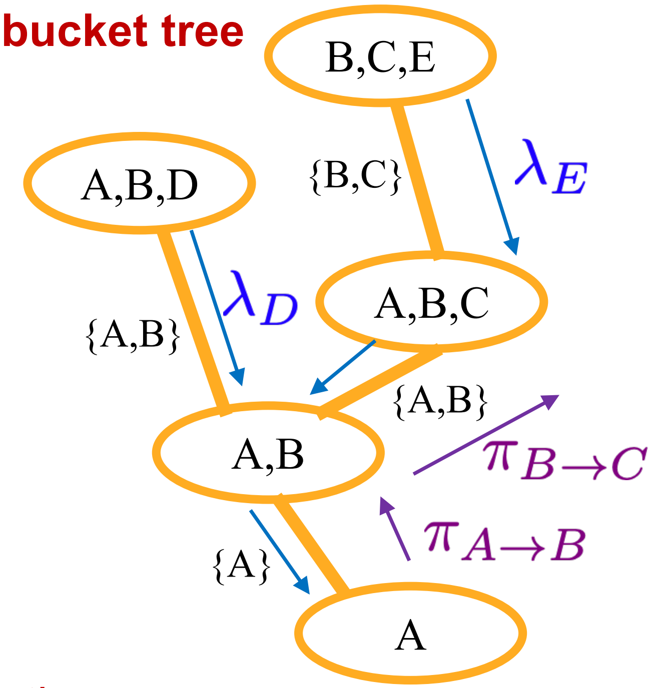
C: $p(C|A)$ $\quad \lambda_E(B,C)$

B: $\lambda_D(A,B)$ $\quad p(B|A)$ $\quad \lambda_C(A,B)$

A: $p(A)$ $\quad \lambda_B(A)$

Z = p(E=0)

View elimination as messages on a **bucket tree**



Can pass messages both ways:

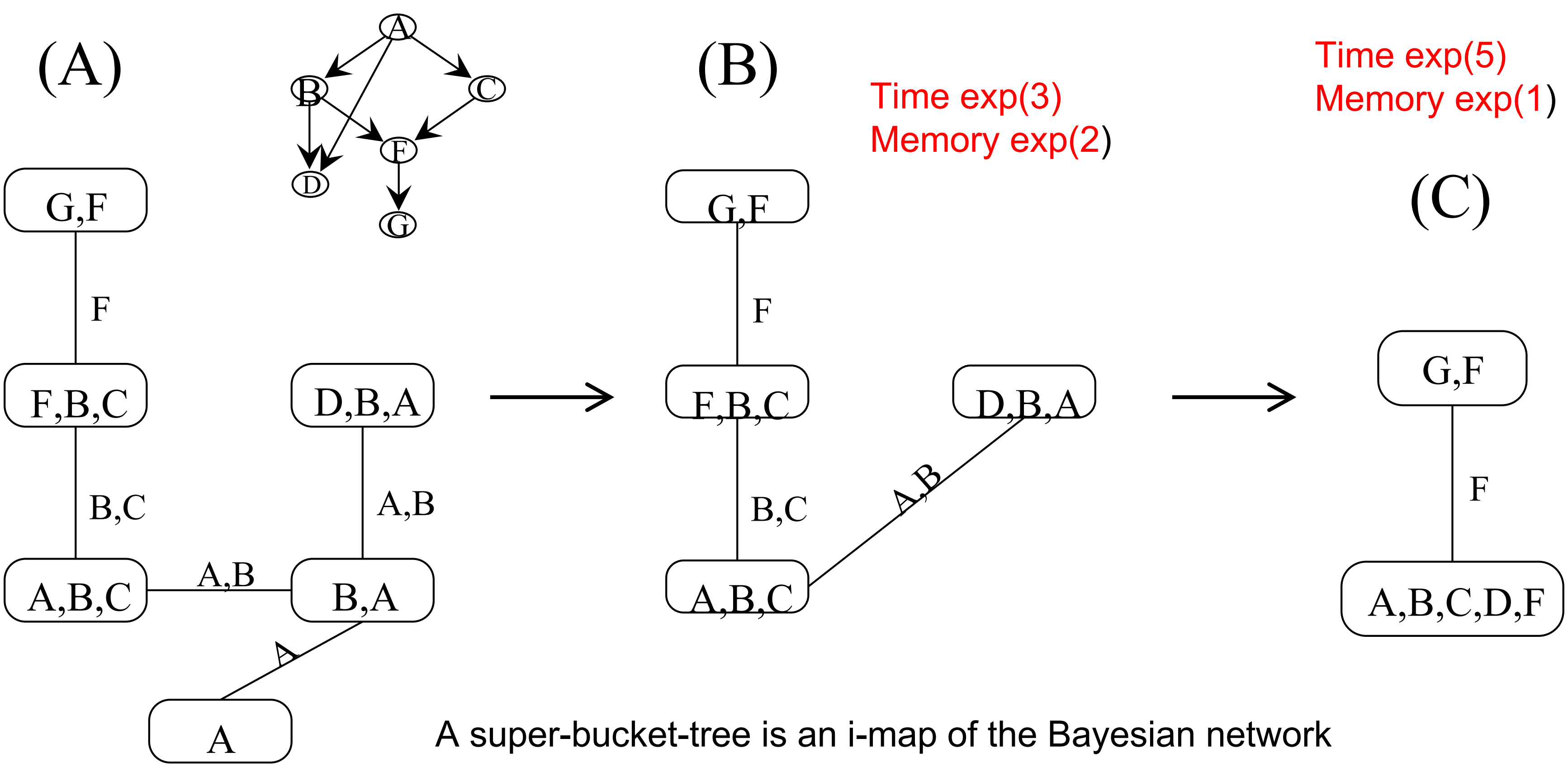
$$p(C|E=0) \propto p(C|A)\ \lambda_E(B,C)\ \pi_{B\to C}(A,B)$$

$$\pi_{B\to C}(A,B) = \lambda_D(A,B)\ p(B|A)\ \pi_{A\to B}(A)$$

$$\pi_{A\to B}(A) = p(A)$$

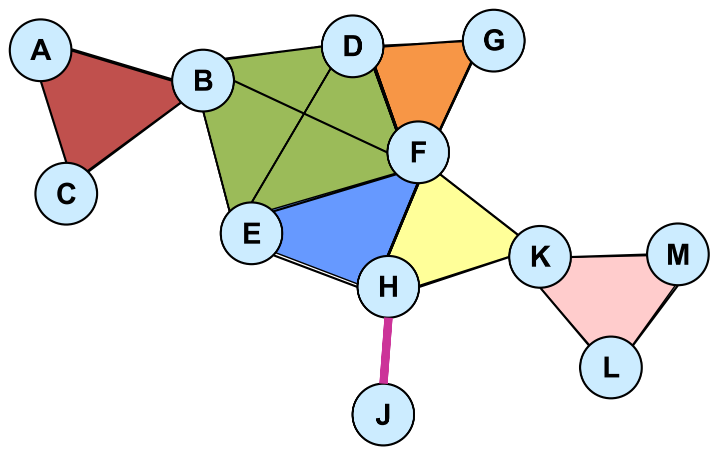
From Buckets to Clusters

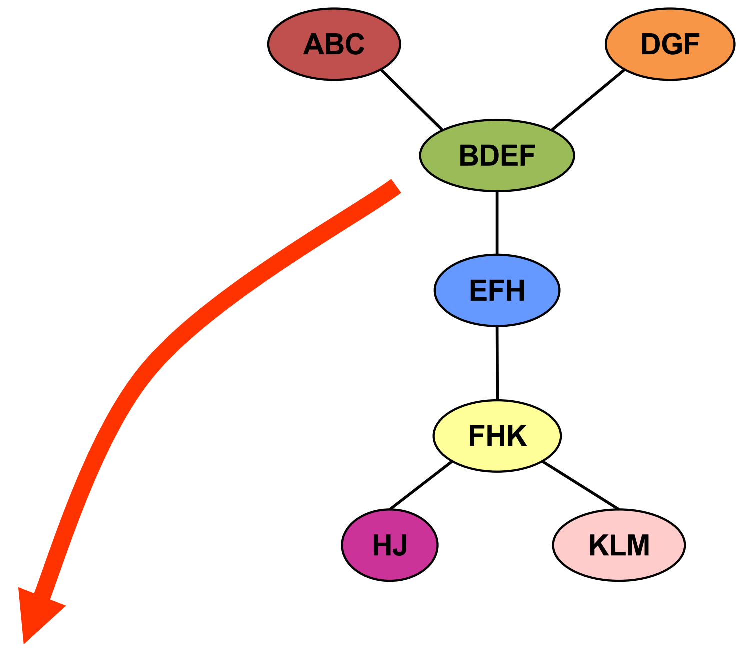
- Merge non-maximal buckets into maximal clusters.
- Connect clusters into a tree: each cluster to one with which it shares a largest subset of variables.
- Separators are variable- intersection on adjacent clusters.



A super-bucket-tree is an i-map of the Bayesian network

The General Tree-Decomposition





Inference algorithm:

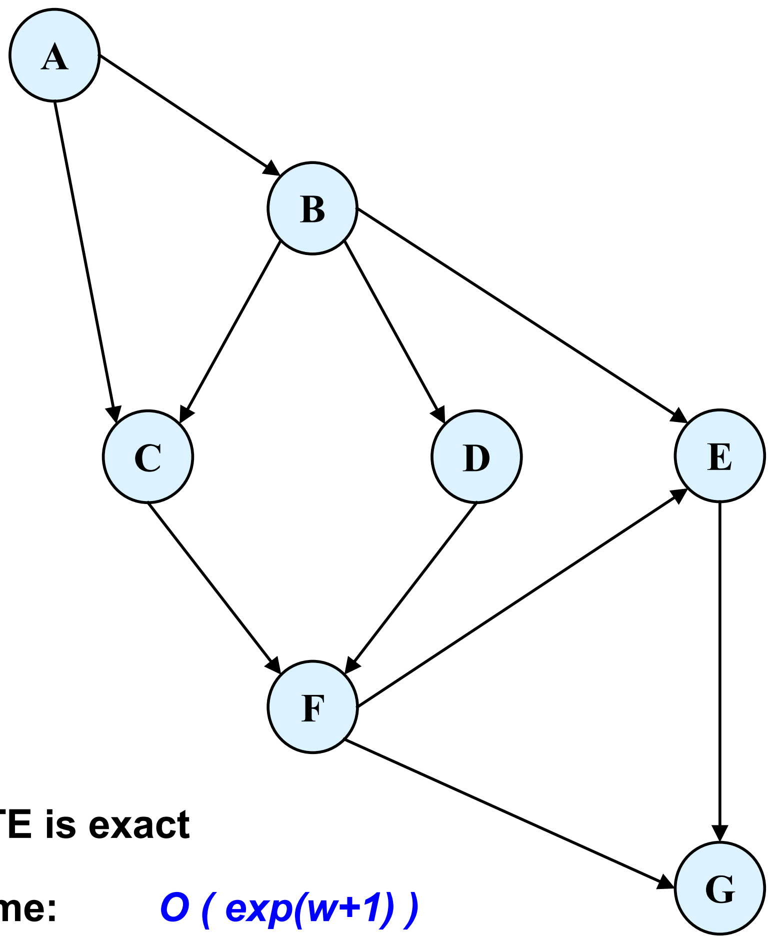
Time: exp(tree-width)

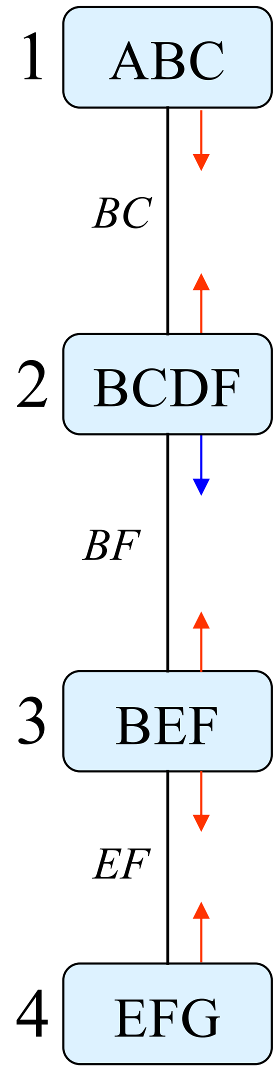
Space: exp(tree-width)

treewidth = 4 - 1 = 3

treewidth = (maximum cluster size) - 1

Cluster-Tree Elimination (CTE), or Join-tree message-passing





$$h_{(1,2)}(b,c) = \sum_{a} p(a) \cdot p(b \mid a) \cdot p(c \mid a,b)$$

$$h_{(2,1)}(b,c) = \sum_{d,f} p(d \mid b) \cdot p(f \mid c,d) \cdot h_{(3,2)}(b,f)$$

$$h_{(2,3)}(b,f) = \sum_{c,d} p(d \mid b) \cdot p(f \mid c,d) \cdot h_{(1,2)}(b,c)$$

$$h_{(3,2)}(b,f) = \sum_{e} p(e \mid b,f) \cdot h_{(4,3)}(e,f)$$

$$h_{(3,4)}(e,f) = \sum_{b} p(e \mid b,f) \cdot h_{(2,3)}(b,f)$$

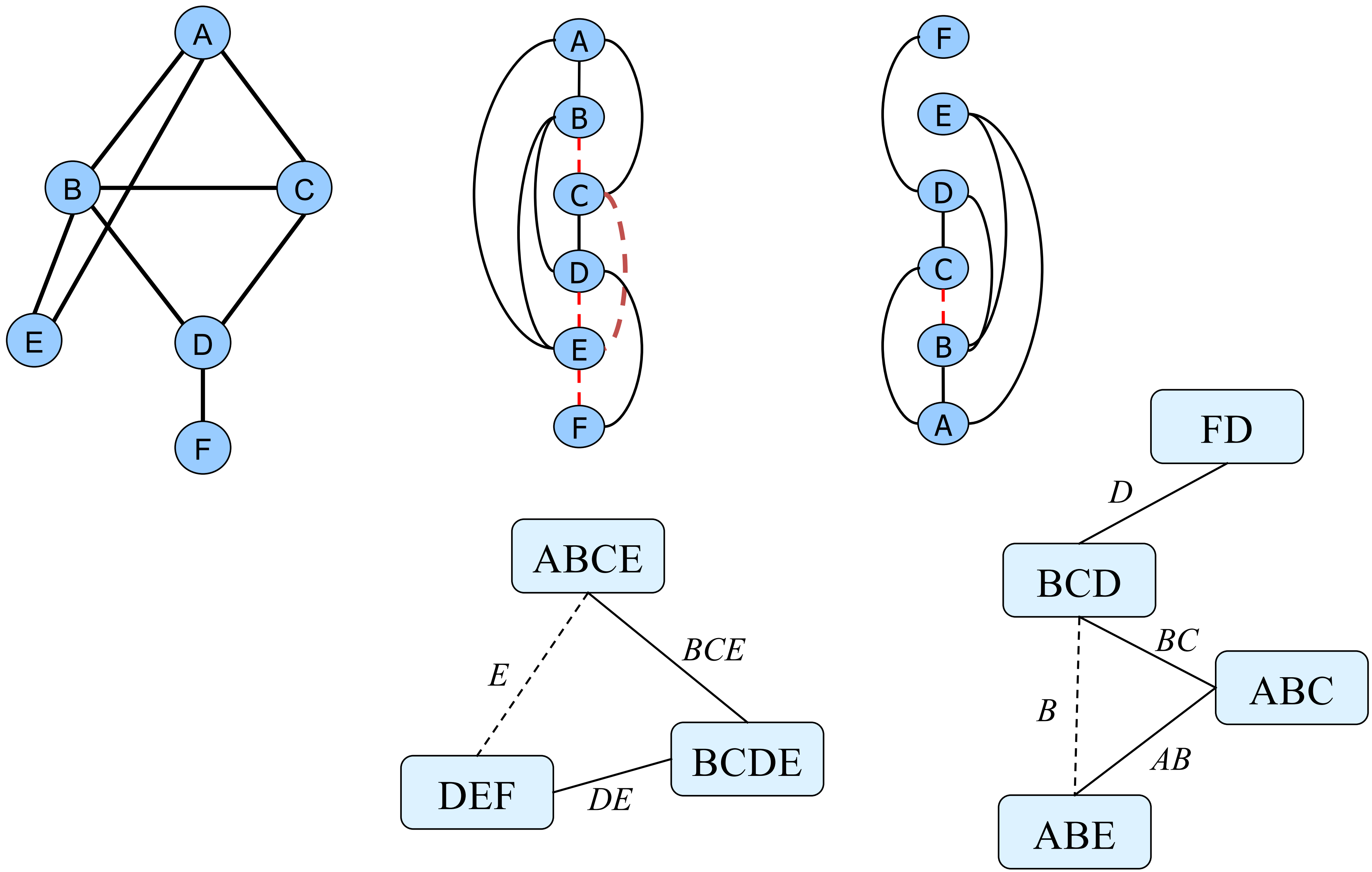
$$h_{(4,3)}(e,f) = p(G = g_e \mid e,f)$$

CTE is exact

Time: ***O (exp(w+1))***
Space: ***O (exp(sep))***

For each cluster P(X|e) is computed, also P(e)

Examples of (Join)-Trees Construction



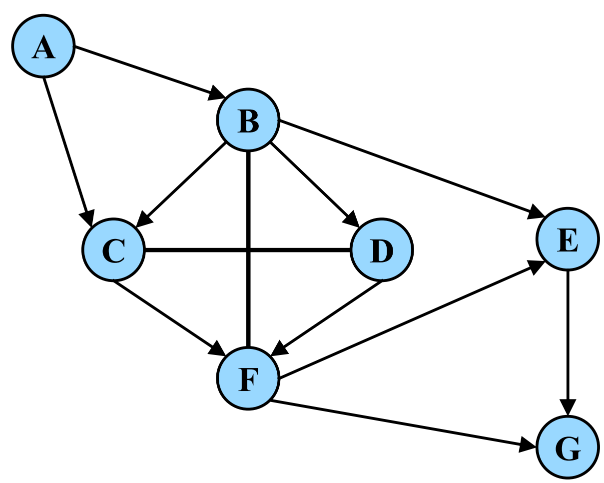
Tree and Hypertree Decompositions

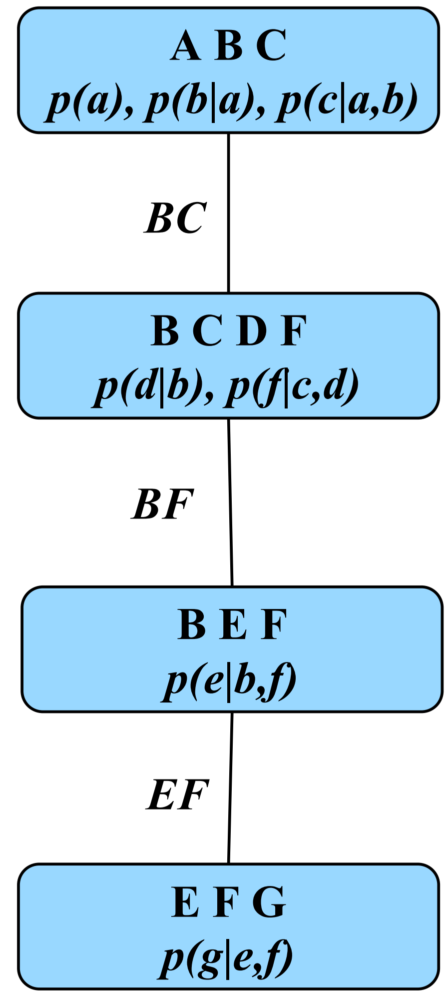
A *tree decomposition* for a belief network $BN = < X, D, G, P >$ is a triple $< T, \chi, \psi >$, where $T = (V, E)$ is a tree and χ and ψ are labeling functions, associating with each vertex $v \in V$ two sets, $\chi(v) \subseteq X$ and $\psi(v) \subseteq P$ satisfying :

1. For each function $p_i \in P$ there is exactly one vertex such that $p_i \in \psi(v)$ and $scope(p_i) \subseteq \chi(v)$
2. For each variable $X_i \in X$ the set $\{v \in V | X_i \in \chi(v)\}$ forms a connected subtree (running intersection property)

Connectedness, or
Running intersection property

Treewidth (w) = 3
Hyper tree-width (hw) =2





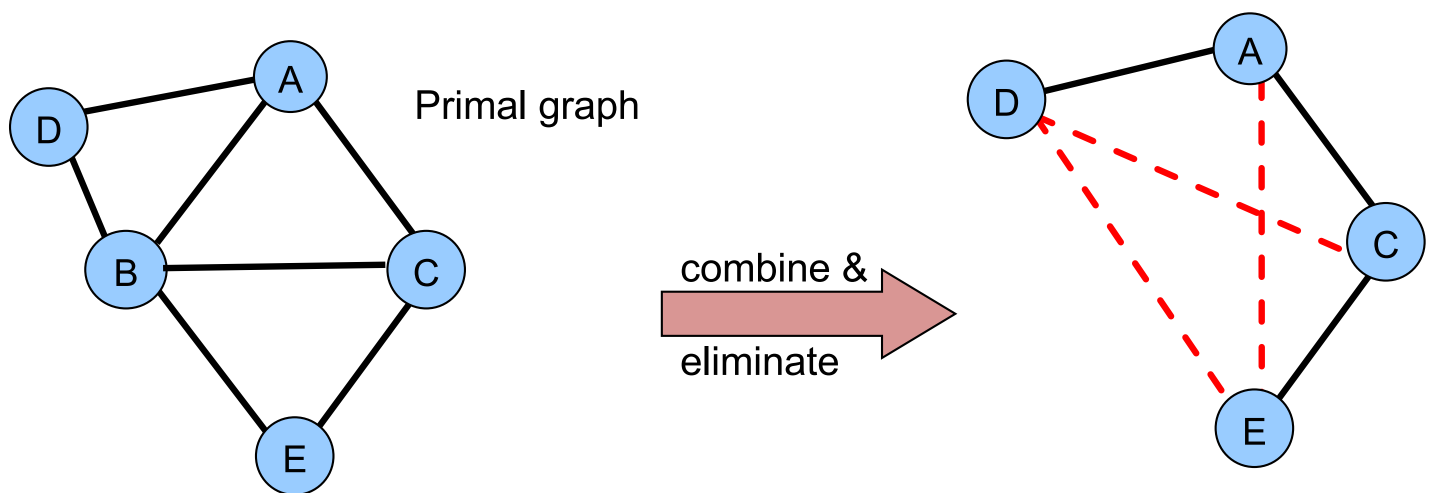
Tree decomposition

Outline

- Graphical Models
- Inference Tasks
- Variable Elimination
- Tree Decompositions
- **Variable Orderings**
- Learning from Data

Variable ordering heuristics

- What makes a good order?
  - Low induced width
  - Elimination creates a function over neighbors
- Finding the best order is hard (NP-complete!)
  - But we can do well with simple heuristics
    - Min-induced-width, Min-Fill, …
  - Anytime algorithms
    - Search-based [Gogate & Dechter 2003]
    - Stochastic (CVO) [Kask, Gelfand & Dechter 2010]

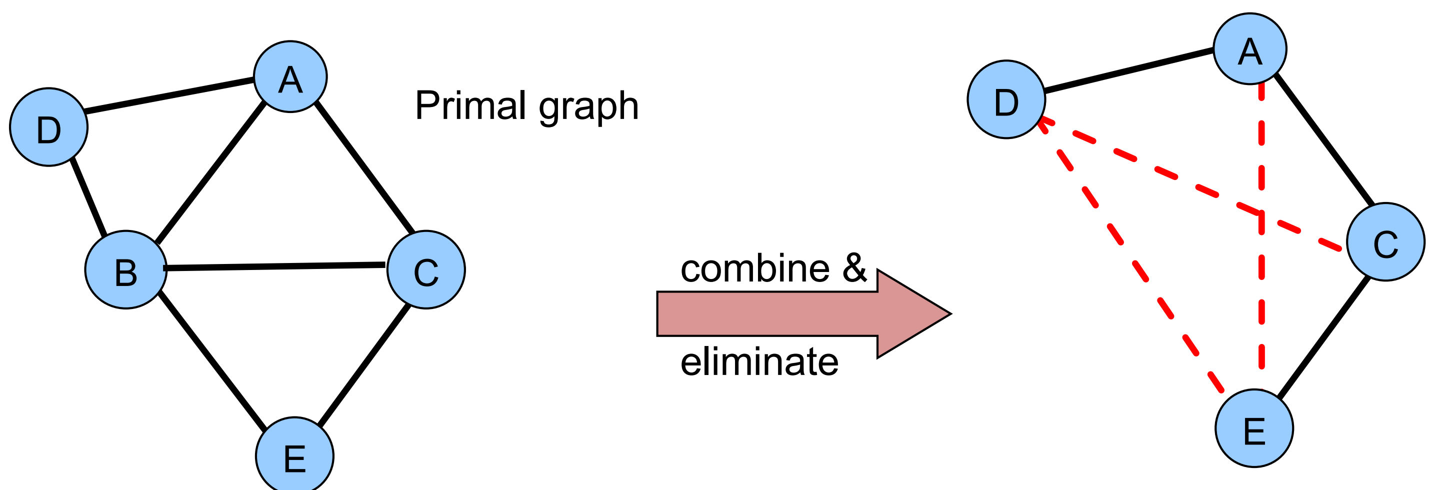


Variable ordering heuristics

- Min (induced) width heuristic

1. for i=1 to n (# of variables)
2. Select a node X_i with smallest degree as next eliminated
3. Connect Xi's neighbors:
4. E = E + { (X_i, X_k) : (X_i, X_j) and (X_i, X_k) in E }
5. Remove X_i from the graph: V = V – $\{X_i\}$
6. end

("Weighted" version: weight edges by domain size)

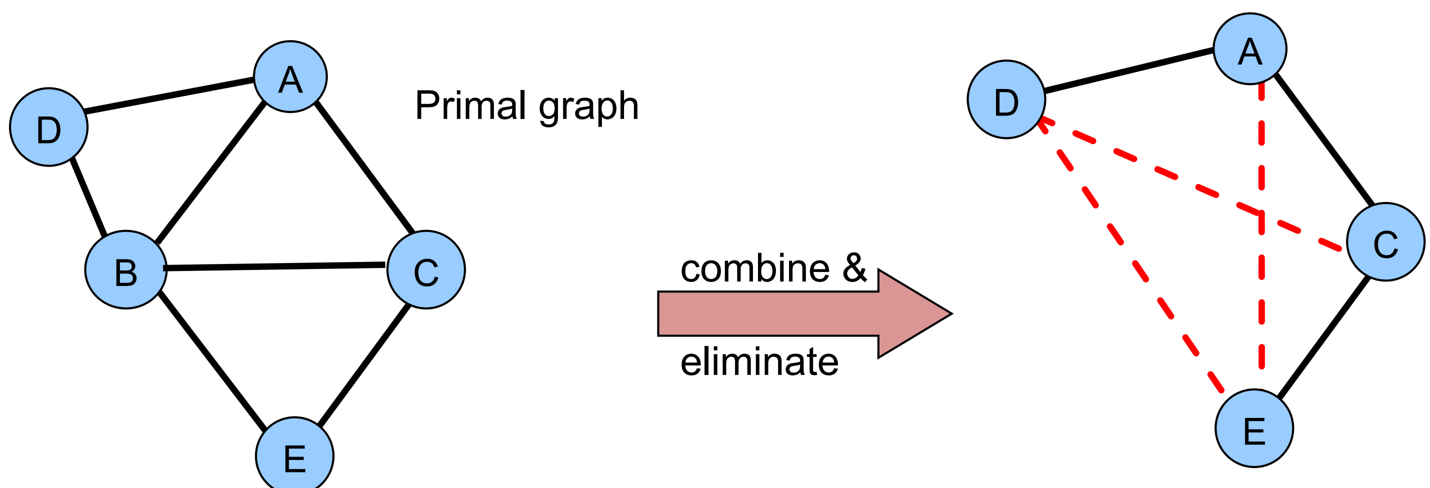


Variable ordering heuristics

- Min fill heuristic

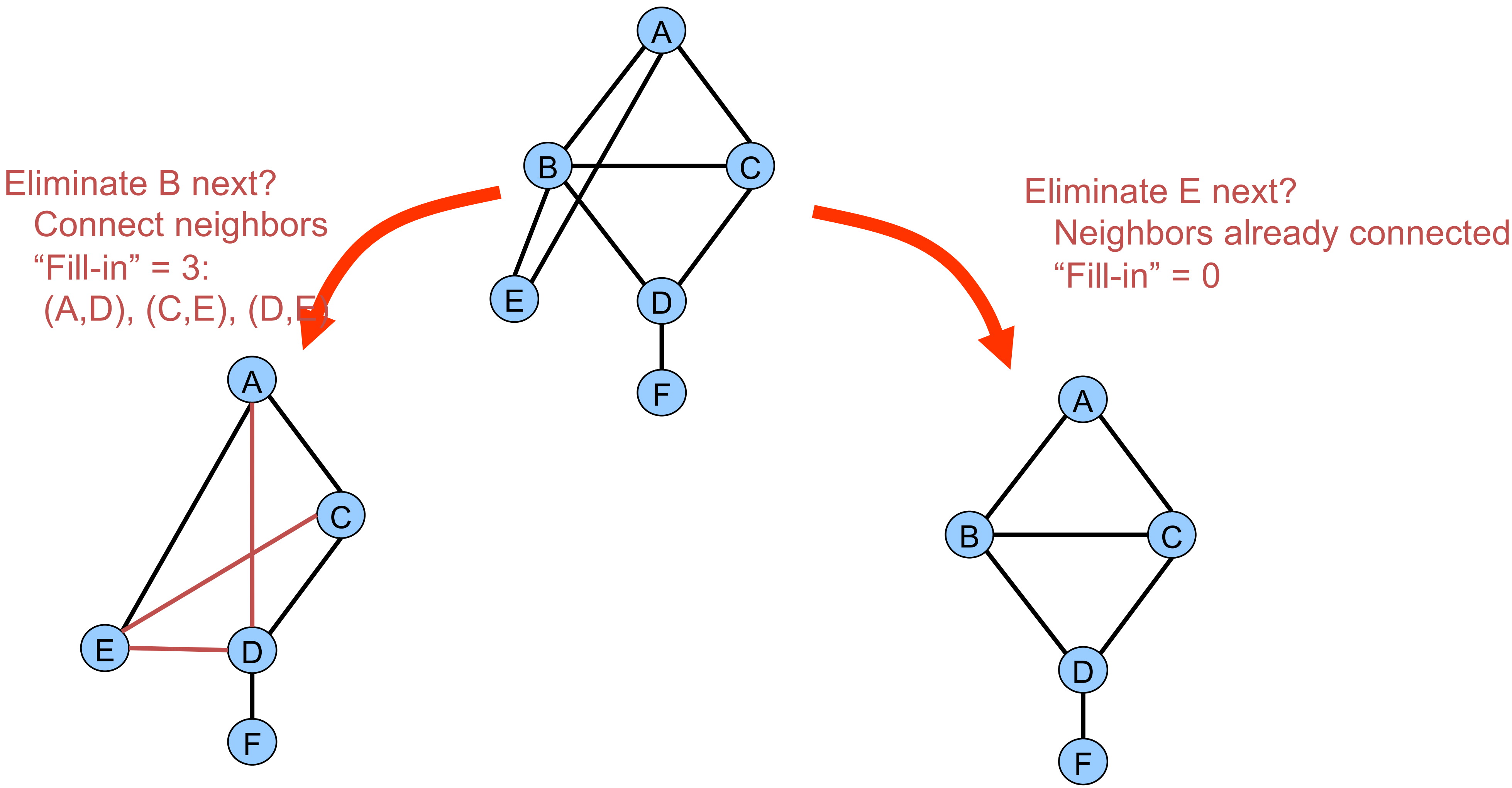
1. for i=1 to n (# of variables)
2. Select a node X_i with smallest "fill edges" as next eliminated
3. Connect Xi's neighbors:
4. E = E + { (X_j, X_k) : (X_i, X_j) and (X_i, X_k) in E }
5. Remove X_i from the graph: V = V – $\{X_i\}$
6. end

("Weighted" version: weight edges by domain size)



Min-Fill Heuristic

- Select the variable that creates the fewest “fill-in” edges

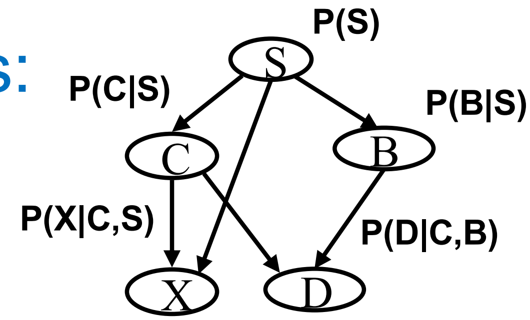


Outline

- Graphical Models
- Inference Tasks
- Variable Elimination
- Tree Decompositions
- Variable Orderings
- **Learning from Data**

Learning Bayesian networks

- Known graph? learn parameters:



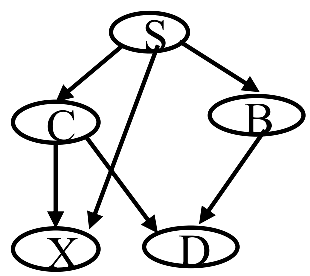
Complete data
- parameter estimation (ML, MAP, …)

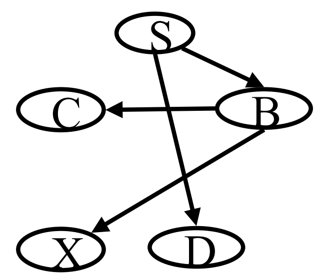
Incomplete data
- parameter optimization (gradient, EM, …)

- Unknown graph? learn graph and parameters

Complete data
- search over graphs (score-based, constraint-based)

Incomplete data
- iterative optimization & search (structural EM, etc.)

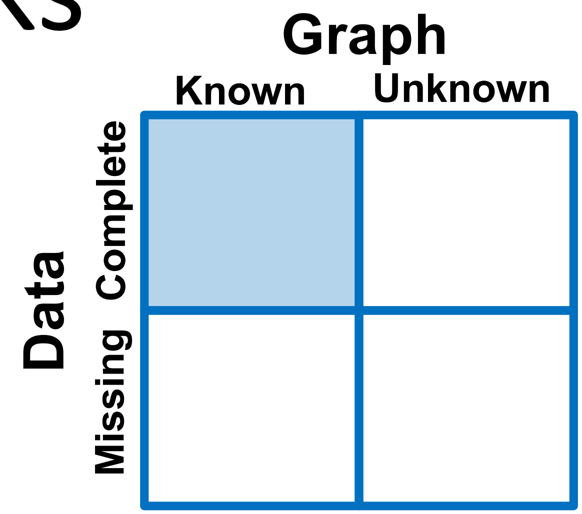




…

Learning Bayesian networks

- Maximum Likelihood estimation
  - Select model that makes the data most probable
- For discrete X_i & no shared parameters
  - ML estimates are empirical probabilities

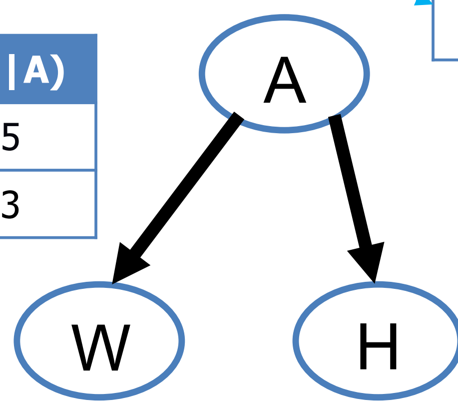


Observed Data

A	W	H
0	0	0
0	0	0
0	1	0
0	1	1
0	0	0
1	0	1
1	1	0
1	1	1

P(A)
3/8

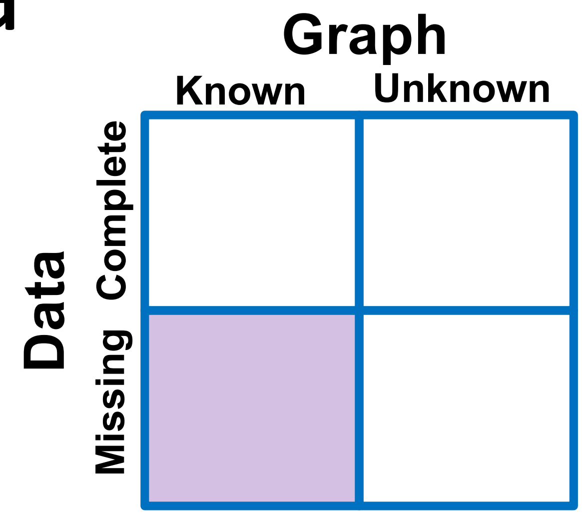
A	P(W|A)
0	2/5
1	2/3



A	P(H|A)
0	1/5
1	2/3

Learning with missing data

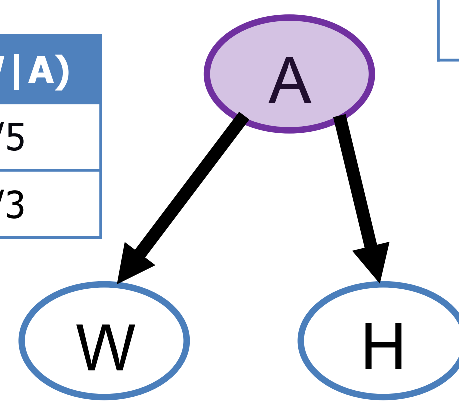
- Latent / hidden variables
  - Value is never observed
  - No unique model (e.g., symmetry)
  - No closed form solution; iterative ML
- More general missing values?
  - May depend on the reason for missingness!



Observed Data

A	W	H
?	0	0
?	0	0
?	1	0
?	1	1
?	0	0
?	0	1
?	1	0
?	1	1

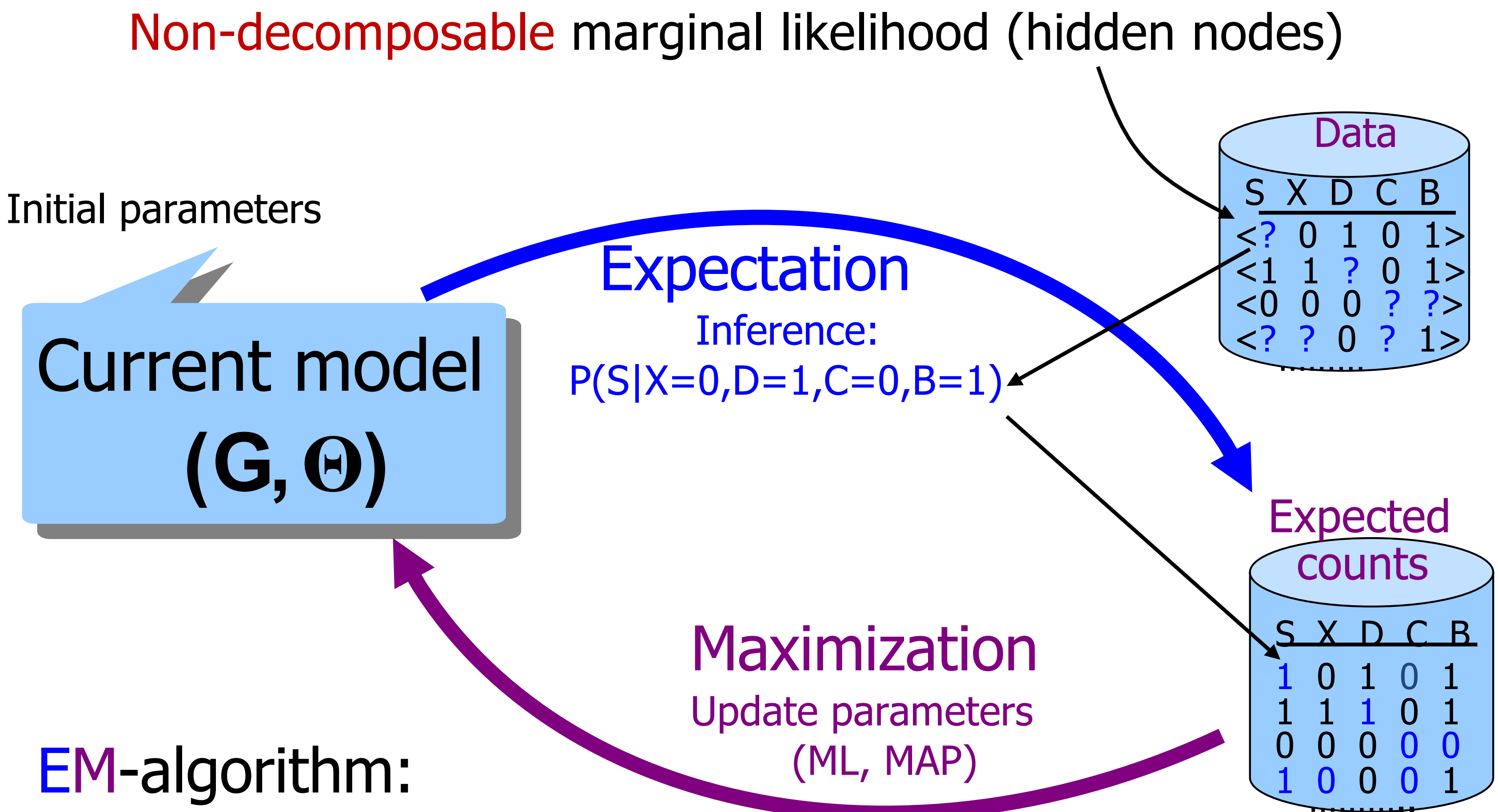
A	P(W|A)
0	2/5
1	2/3



P(A)
3/8

A	P(H|A)
0	1/5
1	2/3

Learning with missing data



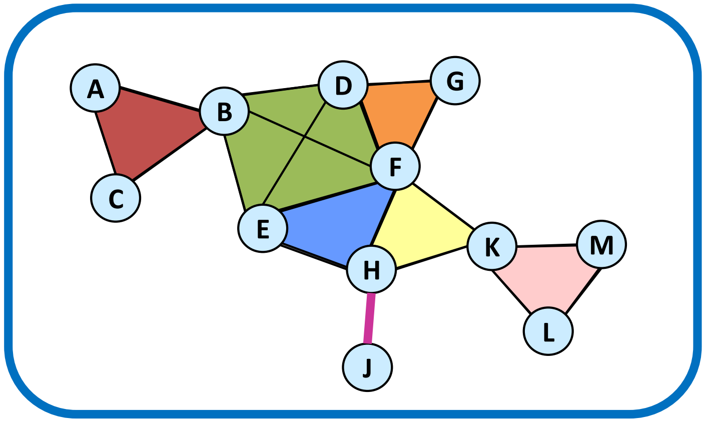
EM-algorithm:
iterate until convergence

Summary

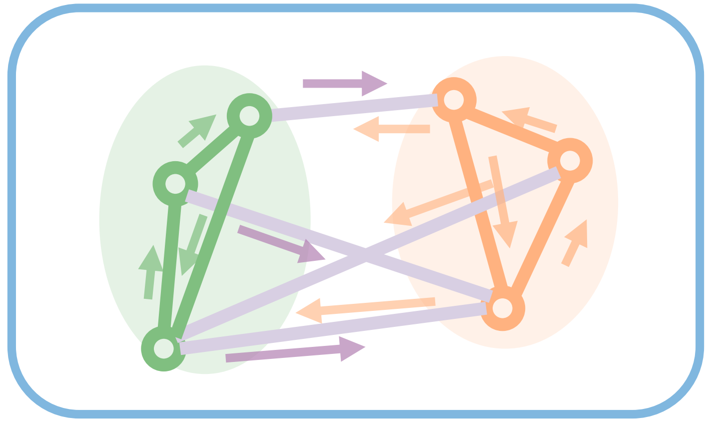
- Graphical Models
- Inference Tasks
- Variable Elimination
- Tree Decomposition
- Variable Orderings
- Learning from Data

Outline of Lectures

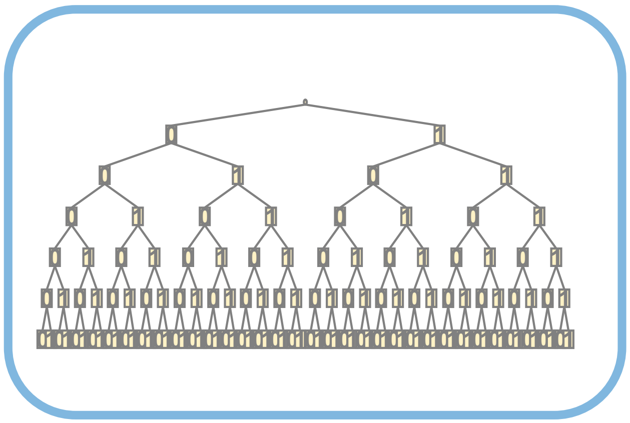
Class 1: Introduction & Inference



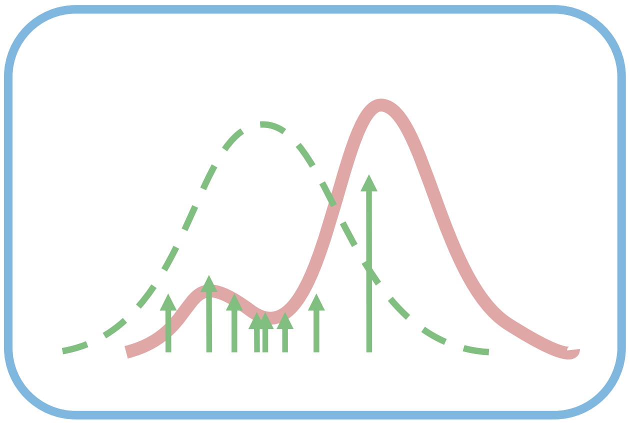
Class 2: Bounds & Variational Methods



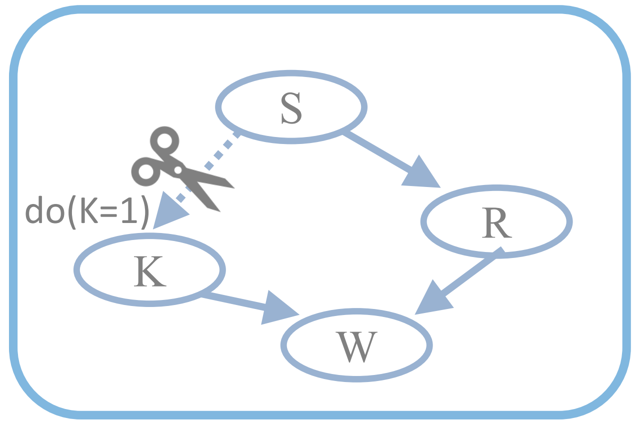
Class 3: Search Methods



Class 4: Monte Carlo Methods



Class 5: Causal Reasoning



Summary

- Formalisms for describing probabilistic & causal models
  - Bayesian networks, Influence Diagrams, Structural Causal Models

- Reasoning & Inference Queries

- Exact inference
  - Bucket elimination: time & memory exponential in the induced width.
  - Tree decomposition: organize computation into message-passing
  - Finding optimal induced width is hard, but greedy schemes work well
  - Inference task may restrict elimination orderings & increase width

- Learning from Data
  - Maximum likelihood learning
  - Easy case: match empirical statistics of the data