Introduction to

Artificial Intelligence

Lecture 12 – Bayesian Network Inference

CS/CNS/EE 154

Andreas Krause

Problems with high-dim. distributions

- Suppose we have *n* propositional symbols
- How many parameters do we need to specify $P(X_1=x_1,\ldots,X_n=x_n)$?

X_1	X_2	…	X_{n-1}	X_n	P(X)
0	0	…	0	0	.01
0	0	…	1	0	.001
0	0	…	1	1	.213
…	…	…	…	…	
1	1	…	1	1	.0003

2^n-1 parameters! ☹

Marginal distributions

- Suppose we have joint distribution $P(X_1,\ldots,X_n)$
- Then

$$P(X_i = x_i) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} P(x_1,\ldots,x_n)$$

Need, because want to compute

$$P(X_1 = T \mid X_3 = F, X_5 = F)$$

$$= \frac{P(X_1 = T, X_3 = F, X_5 = F)}{P(X_3 = F, X_5 = F)}$$

Marginal Distr.

- If all X_i binary: How many terms?

2^{n-1} !

Independent RVs

- What if RVs are independent?

  $P(X_1=x_1,\ldots,X_n=x_n) = P(x_1)\, P(x_2) \ldots P(x_n)$

- How many parameters are needed in this case?

  n

- How about computing $P(x_i)$?

  Indep: $P(X \mid Y, Z) = P(X)$

- Independence too strong assumption… Is there something weaker?

Key concept: Conditional independence

- Random variables X and Y cond. indep. given Z if for all x, y, z:

$$P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)\, P(Y = y \mid Z = z)$$

- If $P(Y=y \mid Z=z)>0$, that's equivalent to

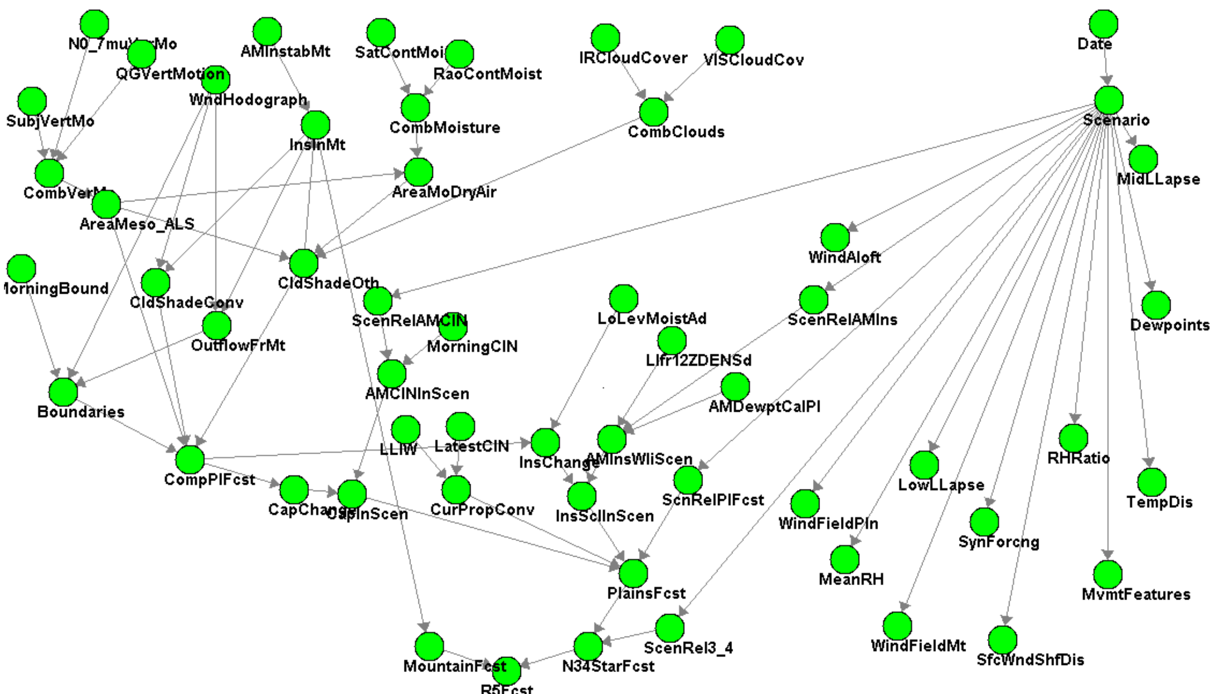
$$P(X = x \mid Z = z, Y = y) = P(X = x \mid Z = z)$$

Similarly for sets of random variables **X**, **Y**, **Z**

We write: $$P \models \mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$$

Bayesian networks

- Compact representation of distributions over large number of variables
- (Often) allows efficient exact inference (computing marginals, etc.)



JavaBayes applet

HailFinder

56 vars

~ 3 states each

➔ ~10^{26} terms

> **10.000 years**

on Top supercomputers

Bayesian networks

- A **Bayesian network structure** is a directed, acyclic graph G, where each vertex s of G is interpreted as a random variable X_s (with unspecified distribution)

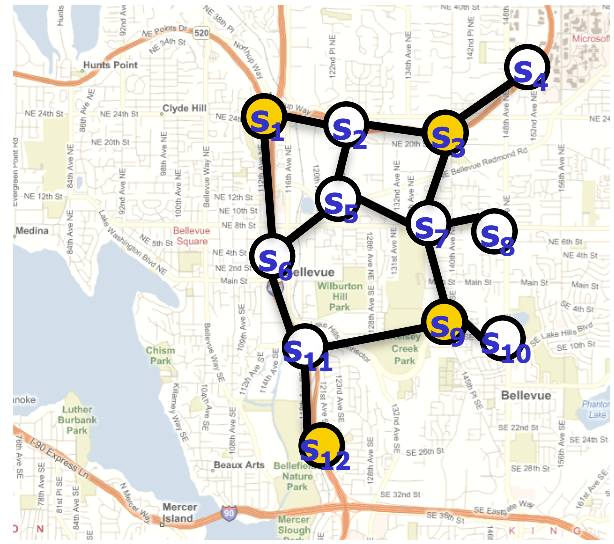
- A **Bayesian network** (G,P) consists of
  - A BN structure G and ..
  - ..a set of conditional probability distributions (CPTs) $P(X_s \mid \mathbf{Pa}_{X_s})$, where $\mathbf{Pa}_{X_s}$ are the parents of node X_s such that
  - (G,P) defines joint distribution

$$P(X_1, ..., X_n) = \prod_i P(X_i \mid \mathbf{Pa}_{X_i})$$

Representing the world using BNs

True distribution P'
with cond. ind. I(P')

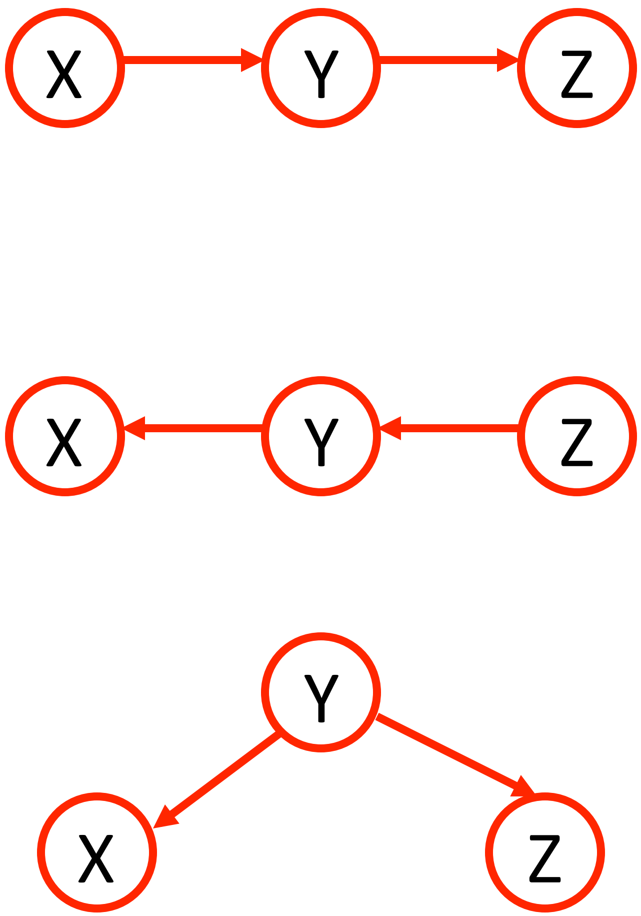
represent



Bayes net (G,P)
with I(P)

- Want to make sure that I(P) is a subset of I(P')
- Need to understand conditional independence properties of BN (G,P)

BNs with 3 nodes

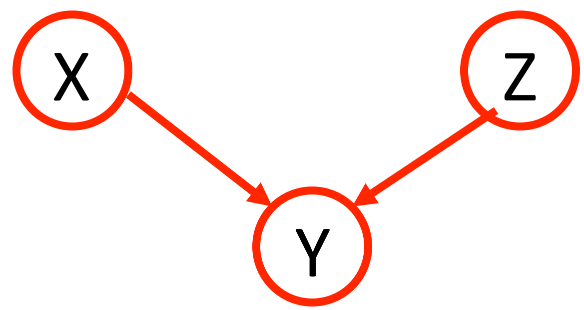


$X \perp Z \mid Y$

$\neg(X, Z)$

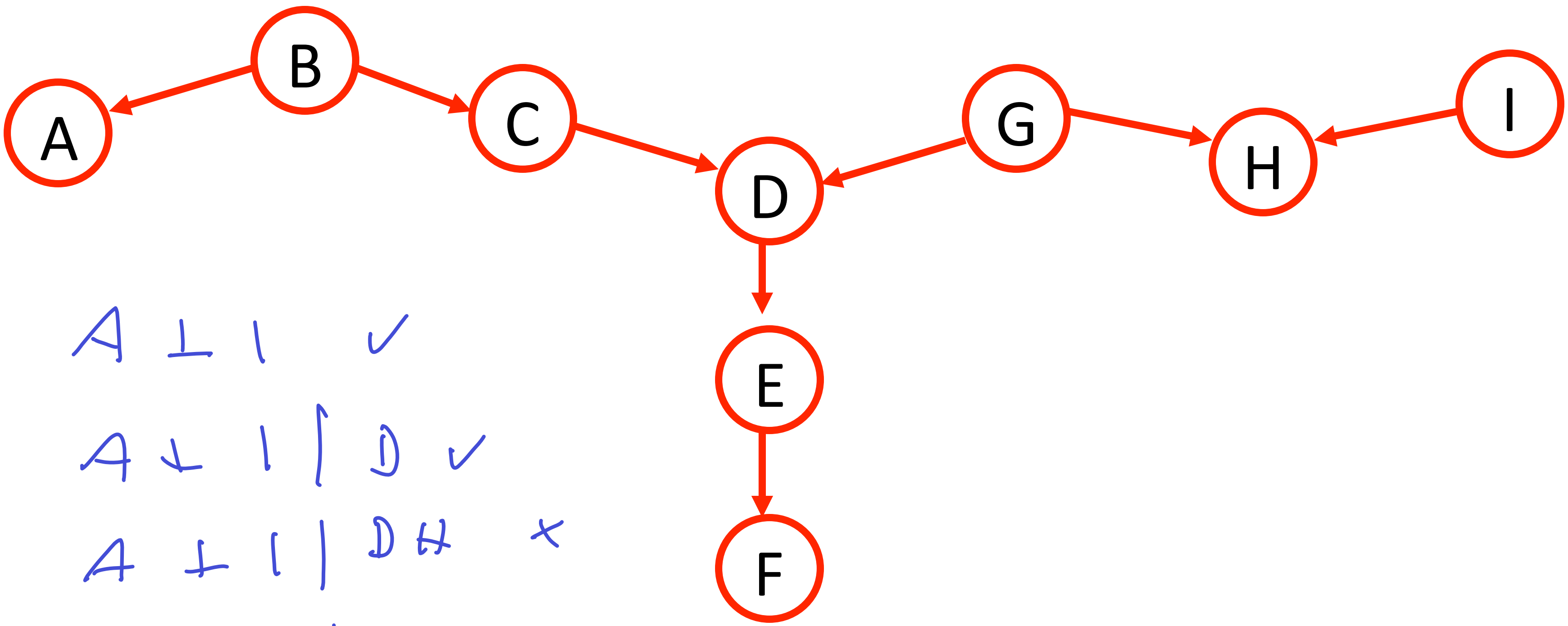
$X \perp Z$

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



Active trails

- When are A and I independent?



$A \perp I$ ✓

$A \perp I \mid D$ ✓

$A \perp I \mid D H$ ✗

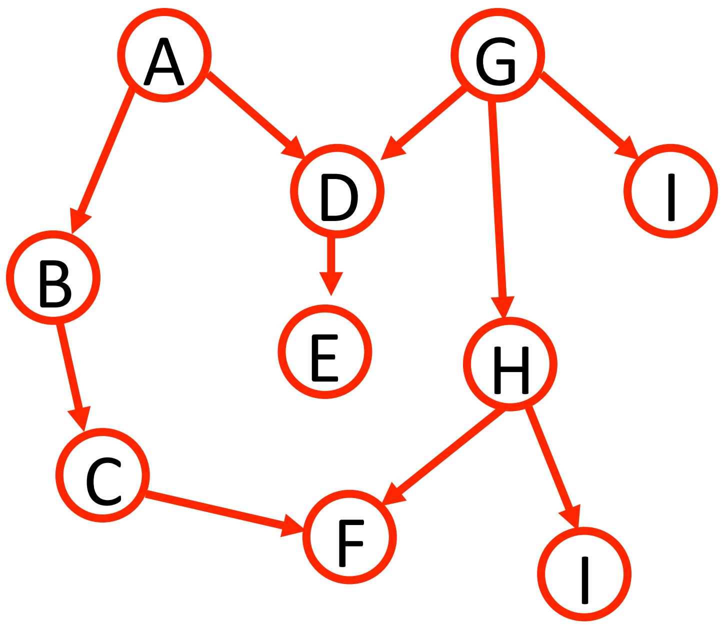
$A \perp I \mid H, F$

Active trails

- An undirected path in BN structure G is called active trail for observed variables **O** µ $\{X_1,\ldots,X_n\}$, if for every consecutive triple of vars X,Y,Z on the path
  - X → Y → Z and Y is unobserved (Y ∉ **O**)
  - X ← Y ← Z and Y is unobserved (Y ∉ **O**)
  - X ← Y → Z and Y is unobserved (Y ∉ **O**)
  - X → Y ← Z and Y *or any of Y's descendants* is observed

- Any variables X_i and X_j for which there is no active trail for observations **O** are called d-separated by **O**
  We write d-sep(X_i;X_j | **O**)
- Sets **A** and **B** are d-separated given **O** if d-sep(X,Y |**O**) for all X in **A**, Y in **B**. Write d-sep(**A; B | O**)

d-separation and independence



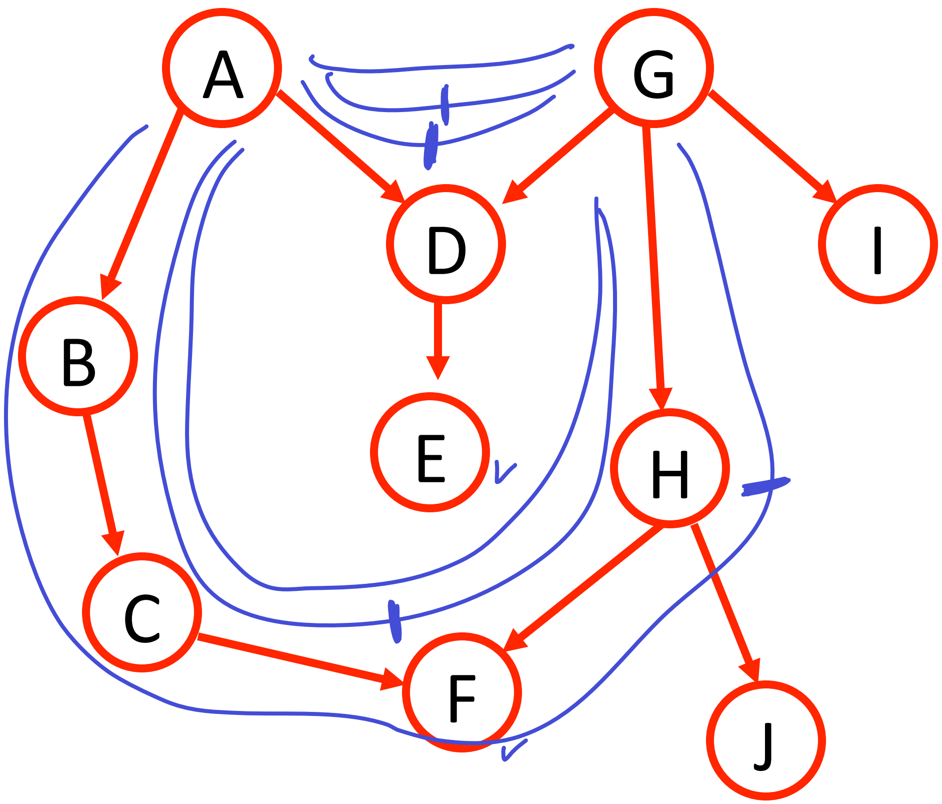
Theorem:

$$\text{d-sep}(X; Y \mid \mathbf{Z}) \Rightarrow X \perp Y \mid \mathbf{Z}$$

i.e., X cond. indep. Y given Z if there does not exist any active trail between X and Y for observations **Z**

- Converse does not hold in general!
- But for "almost" all distributions (except set of measure 0)

Examples



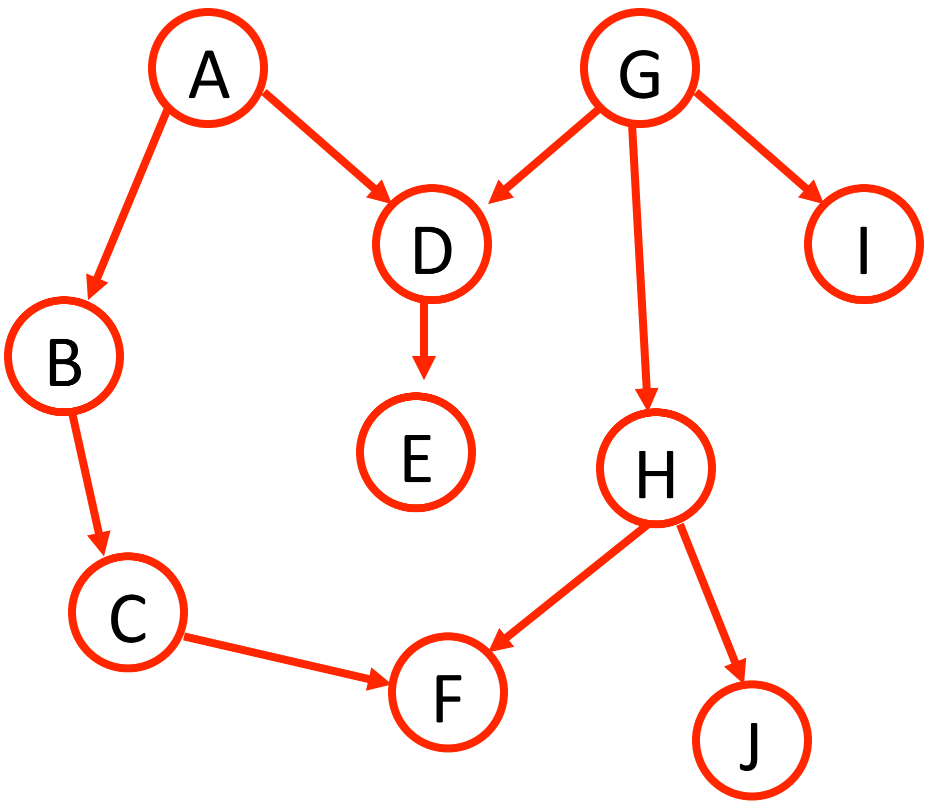
$A \perp G$ ✓

$A \perp G \mid F$ ✗

$A \perp G \mid F, H$ ✓

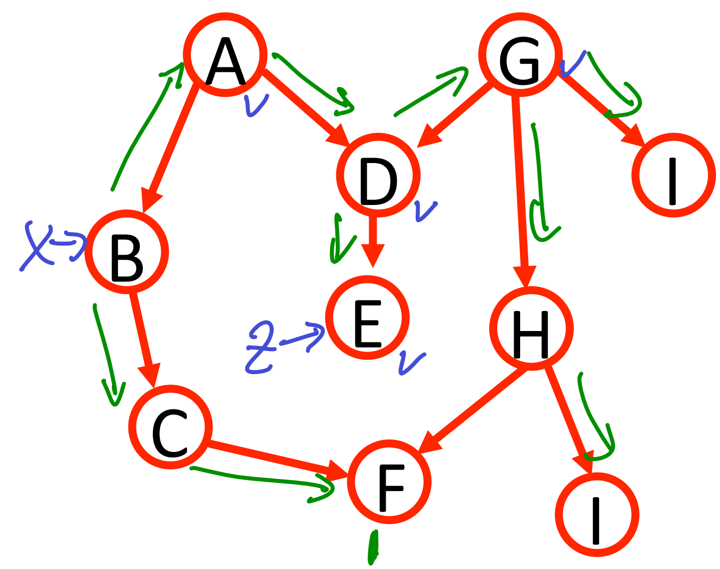
$A \perp G \mid F, H, E$ ✗

More examples

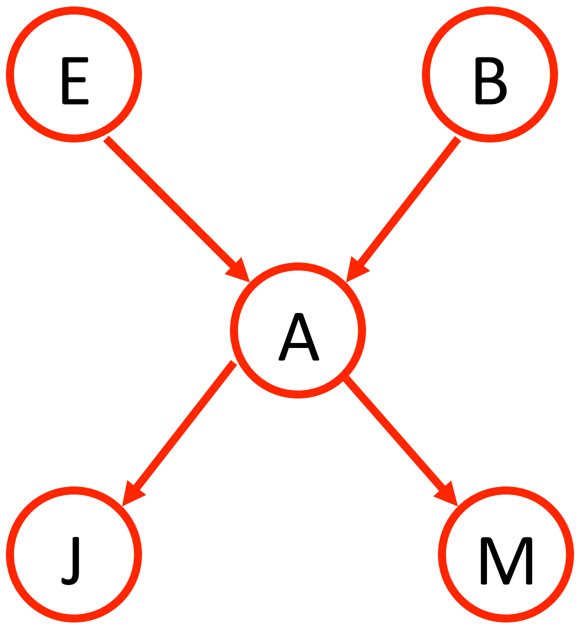


Algorithm for d-separation

- How can we check if d-sep(X; Y | **Z**)?
  - *Idea*: Check every possible path connecting X and Y and verify conditions
  - Exponentially many paths!!! ☹
- Linear time algorithm:
  Find all nodes reachable from X
  - 1. Mark **Z** and its ancestors
  - 2. Do breadth-first search starting from X; stop if path is blocked
  - Have to be careful with implementation details (see reading)



Typical queries: Conditional distribution



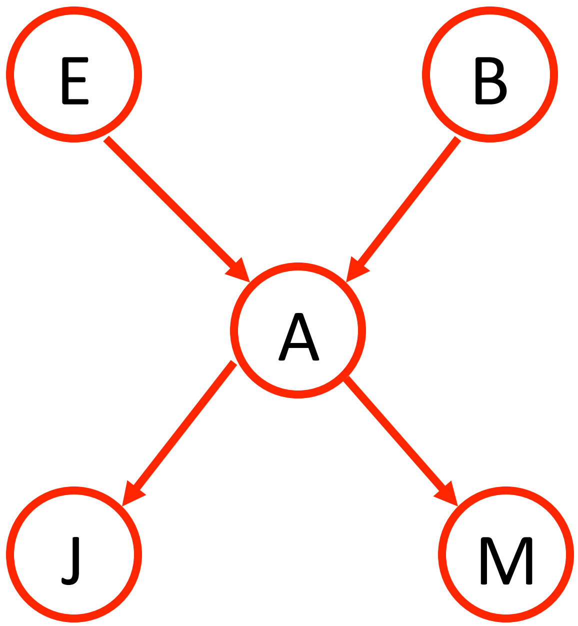
- Compute distribution of some variables given values for others

$P(E \mid J=T)$?

$P(E, B \mid J=T, M=F)$?

$= \frac{1}{Z} P(E, B, J=T, M=F)$

Typical queries: Maximization



- MPE (Most probable explanation):

  Given values for some vars, compute most likely assignment to all remaining vars

$$(a^*, e^*, b^*) = \arg\max_{e,b,a} P(E=e, B=b, A=a \mid J=T, M=F)$$

- MAP (Maximum a posteriori):

  Compute most likely assignment to some variables

More general than MPE

$$(e^*, b^*) = \arg\max_{e,b} P(e, b \mid J=T, M=F)$$

Hardness of inference for general BNs

- Computing conditional distributions:
  - Exact solution: #P-complete
  - NP-hard to obtain any nontrivial approximation

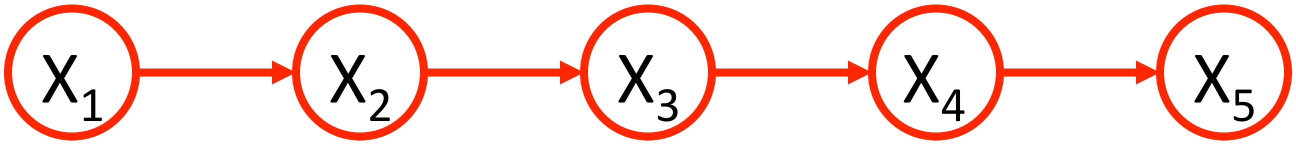
    E.g. NP-hard to obtain $\hat{P}$ s.t. $|P-\hat{P}| < \frac{1}{2}$
- Maximization:
  - MPE: NP-complete
  - MAP: NP^{PP}-complete
- Inference in general BNs is really hard ☹
- Is all hope lost?

Inference

- Can exploit structure (conditional independence) to efficiently perform **exact inference** in many practical situations

- For BNs where exact inference is not possible, can use algorithms for **approximate inference** (later)

Potential for savings: Variable elimination!



$$P(X_1, X_2, \ldots, X_5) = P(X_1)\, P(X_2 \mid X_1)\, P(X_3 \mid X_2) \cdots P(X_5 \mid X_4)$$

Sps we want $P(X_5 \mid X_1) = \frac{1}{Z} P(X_1, X_5)$

$$P(X_1, X_5) = \sum_{x_2} \sum_{x_3} \sum_{x_4} P(X_1)\, P(X_2 \mid X_1) \cdots P(X_5 \mid X_4)$$

$$= \sum_{x_2} \sum_{x_3} P(X_1)\, P(X_2 \mid X_1)\, P(X_3 \mid X_2) \underbrace{\sum_{x_4} P(X_4 \mid X_3)\, P(X_5 \mid X_4)}_{g_4(X_3, X_5)}$$

For n vars:

Naive: 2^{n-2}

VE: $2 \cdot (n-2)$

$$= \sum_{x_2} P(X_1)\, P(X_2 \mid X_1) \underbrace{\sum_{x_3} P(X_3 \mid X_2)\, g_4(X_3, X_5)}_{g_3(X_2, X_5)}$$

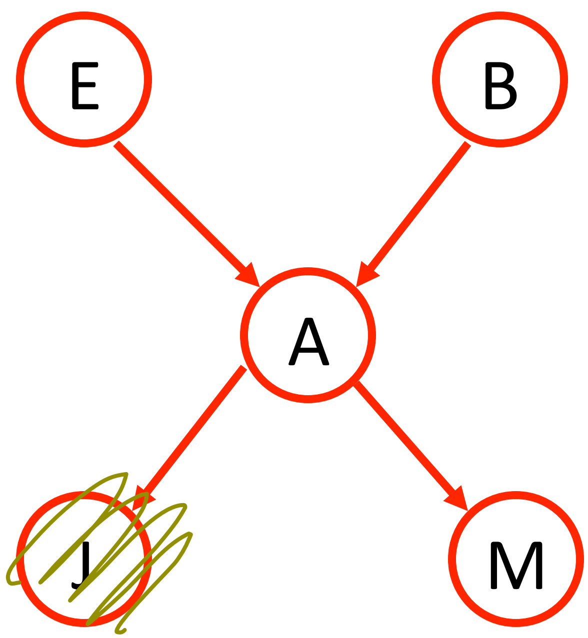
$$= P(X_1) \underbrace{\sum_{x_2} P(X_2 \mid X_1)\, g_3(X_2, X_5)}_{g_2(X_1, X_5)}$$

$f(X_1, X_2, X_5) = P(X_1) P(X_2 \mid X_1)\, g_3(X_2, X_5)$
$= P(X_1, X_2, X_5)$

Intermediate solutions are distributions on fewer variables!

Variable elimination in general graphs

- Push sums through product as far as possible
- Create new factor by summing out variables



$$P(E \mid m) = \frac{1}{Z} P(E, m)$$

$$\sum_{b,a,j} P(E, m, b, a, j)$$

$$= \sum_{b,a,j} P(E) P(b) P(a \mid E, b) P(j \mid a) P(m \mid a)$$

$$= \sum_{b,a} P(E) P(b) P(a \mid E, b) P(m \mid a) \underbrace{\sum_j P(j \mid a)}_{1}$$

$$= \sum_{a} P(E) P(m \mid a) \underbrace{\sum_b P(b) P(a \mid E, b)}_{g_b(a, E)}$$

$$= P(E) \sum_a P(m \mid a)\, g_b(a, E)$$

Variable elimination algorithm

- Given BN and Query P(X | **E**=**e**)
- Choose an ordering of $X_1,\ldots,X_n$
- Set up initial factors: $f_i = P(X_i \mid \mathbf{Pa_i})$
- For i =1:n, $X_i \notin \{X,\mathbf{E}\}$
  - Collect all factors f that include X_i
  - Generate new factor by marginalizing out X_i

$$g = \sum_{x_i} \prod_j f_j$$

  - Add g to set of factors
- Renormalize P(x,**e**) to get P(x | **e**)

Multiplying factors

$$g = \sum_{x_i} \prod_{j} f_j$$

A	B	f_1(A,B)
0	0	.1
0	1	.3
1	0	.7
1	1	.01

B	C	f_2(B,C)
0	0	.4
0	1	.2
1	0	.5
1	1	0

$f' = f_1 \cdot f_2$

A	B	C	f'(A,B,C)
0	0	0	$.1 \cdot .4$
0	0	1	$.1 \cdot .2$
0	1	0	⋮
⋮	⋮	⋮	

Marginalizing factors

$$g = \sum_{x_i} \prod_j f_j$$

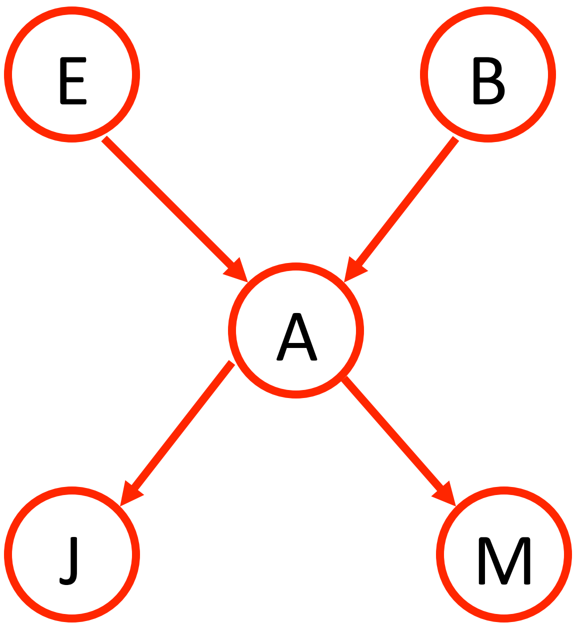
A	B	f'(A,B)
0	0	.1
0	1	.3
1	0	.7
1	1	.01

$$g = \sum_a f'$$

B	g(B)
0	.1+.7
1	.3+.01

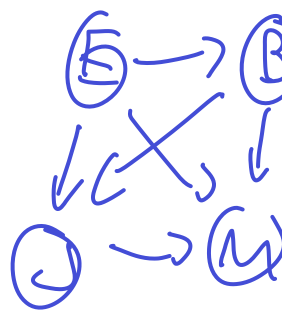
The order matters!

- $P(A,B,E,J,M) = P(E)\,P(B)\,P(A|E,B)\,P(J|A)\,P(M|A)$
- What if we eliminate A first?

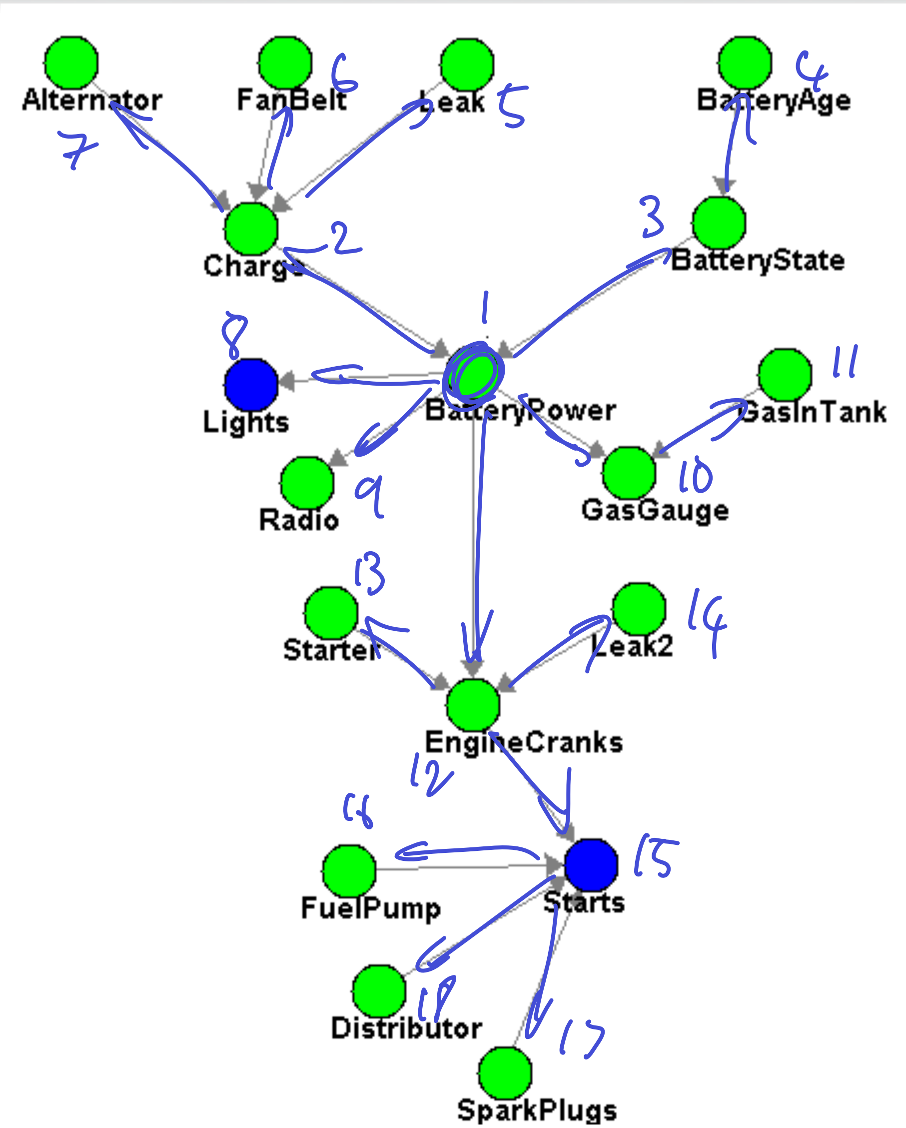


$$\underbrace{\sum_a P(a|e,b)\,P(j|a)\,P(m|a)}_{g_a(E,B,J,M)}$$

$\Rightarrow$



Variable elimination for polytrees



A DAG is a polytree iff dropping edge directions results in a tree

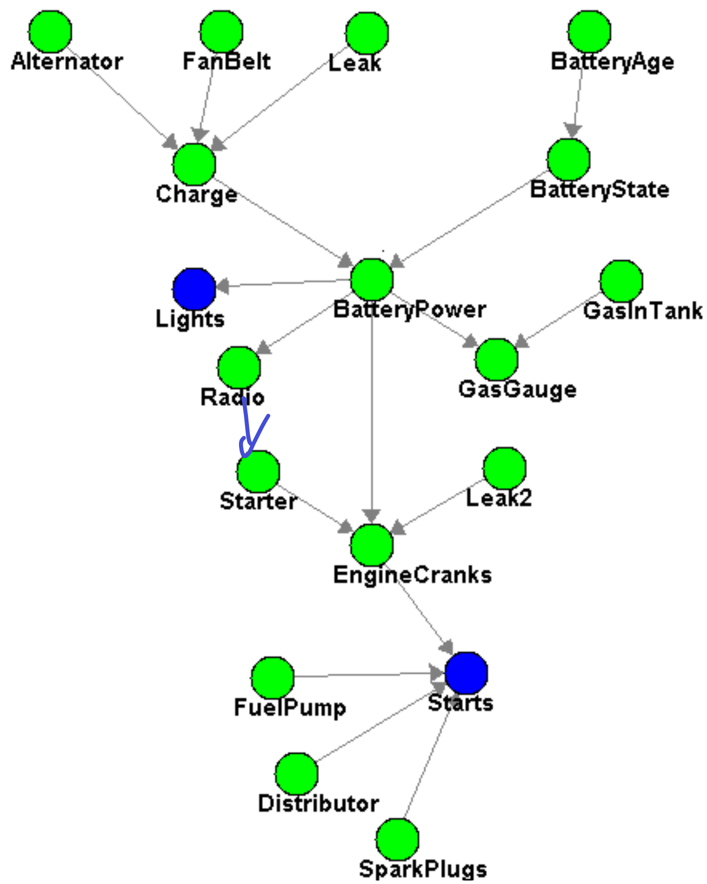
Eliminate in increasing order of degree in undirected graph

⟺

Pick root, orient edges away from root
use inverse topol. ordering

What about loops?

- Can do efficient inference on trees.
- What if the graph has loops?

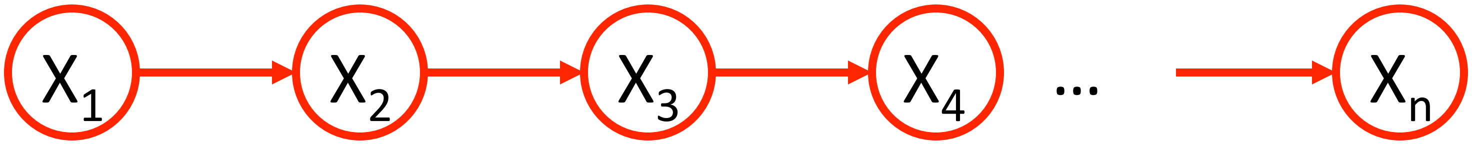


Cutset-Conditioning

- Suppose we would like to compute $P(X_i \mid \mathbf{E=e})$
- Pick subset of variables **A** (called "cutset") such that remaining variables form a tree
- Calculate $P(X_i, \mathbf{A=a} \mid \mathbf{E=e})$ for each assignment **A=a**
- Then $P(X_i \mid \mathbf{E=e}) = \Sigma_a P(X_i, \mathbf{A=a} \mid \mathbf{E=e})$

- *Analog to Constraint Satisfaction Problems*

Answering multiple queries



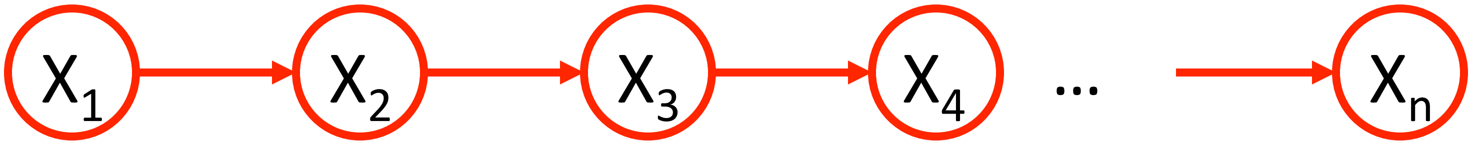
- Suppose, I would like $P(X_i \mid X_n = T)$ for all I
- Naïve approach?

Re run VE for each i

Cost $\Theta(n)$ for each i

$\Rightarrow \Theta(n^2)$

Reusing computation



$$P(x_1, x_n) = \sum_{x_2 \ldots x_{n-2}} P(x_1)\, P(x_2|x_1) \cdots P(x_{n-2}|x_{n-3}) \underbrace{\sum_{x_{n-1}} P(x_{n-1}|x_{n-2})\, P(x_n|x_{n-1})}_{g_{n-1}}$$

$$P(x_2, x_n) = \sum_{x_1, x_3, \ldots, x_{n-2}} P(x_1) \cdots \underbrace{\sum_{x_{n-1}} P(x_{n-1}|x_{n-2})\, P(x_n|x_{n-1})}_{g_{n-1}}$$

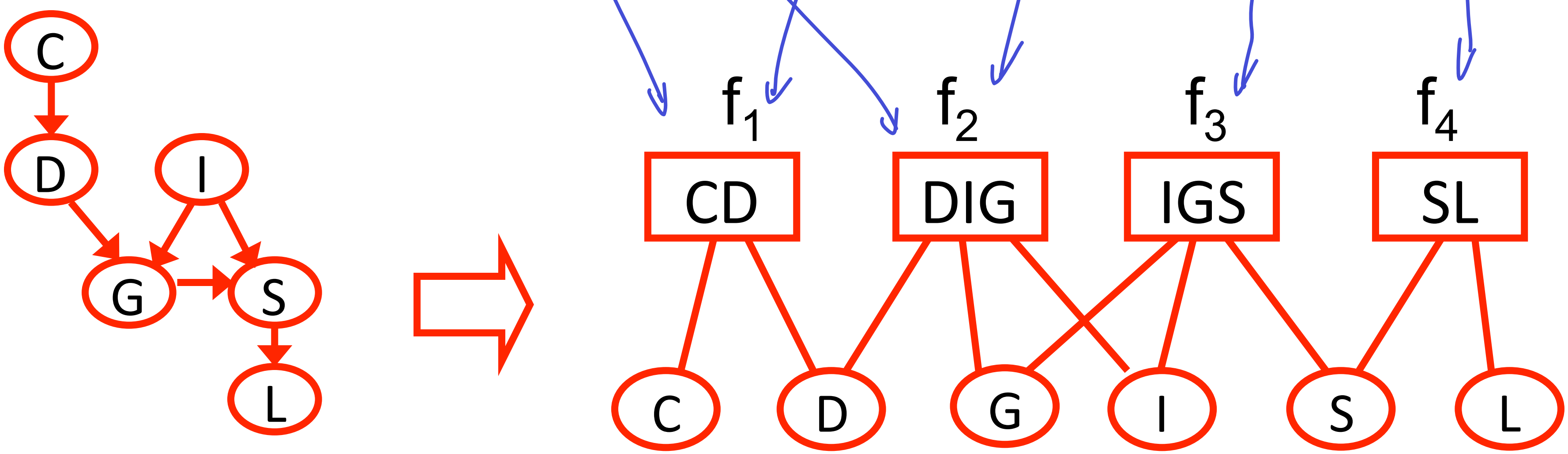
Reusing computation

- Often, want to compute conditional distributions of many variables, for fixed observations
- E.g., probability of *Pits* at different locations given observed *Breezes*
- Repeatedly performing variable elimination is wasteful (many factors are recomputed)
- Need right data-structure to avoid recomputation

➔ Message passing on factor graphs

Factor graphs

- P(C,D,G,I,S,L) = P(C) P(I) P(D|C) P(G|D,I) P(S|I,G) P(L|S)



$f_1(c,d) = P(c)\,P(d|c)$

$f_2(d,i,g) = P(i)\,P(g|d,i)$

$f_3(i,g,s) = P(s|i,g)$

Factor graph

- A factor graph for a Bayesian network is a bipartite graph consisting of
  - Variables and
  - Factors
- Each factor is associated with a subset of variables, and all CPDs of the Bayesian network have to be assigned to one of the factor nodes