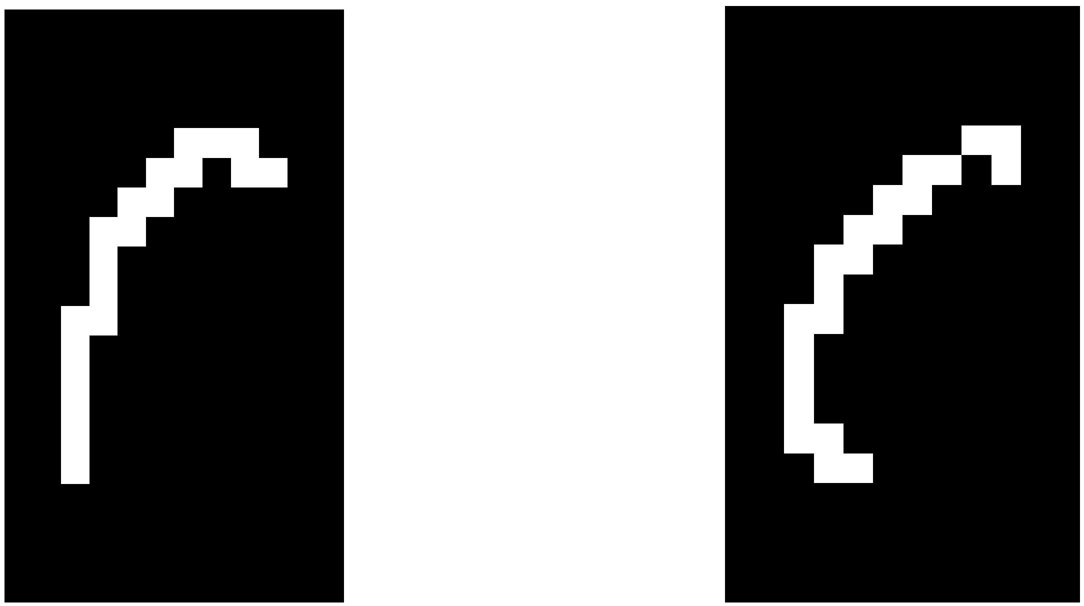
Bayesian Networks – Inference

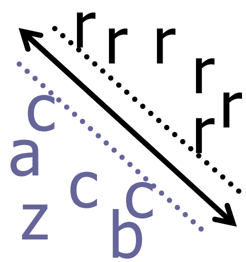
Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

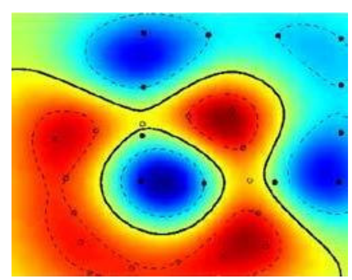
March 21st, 2007

Handwriting recognition



Character recognition, e.g., kernel SVMs





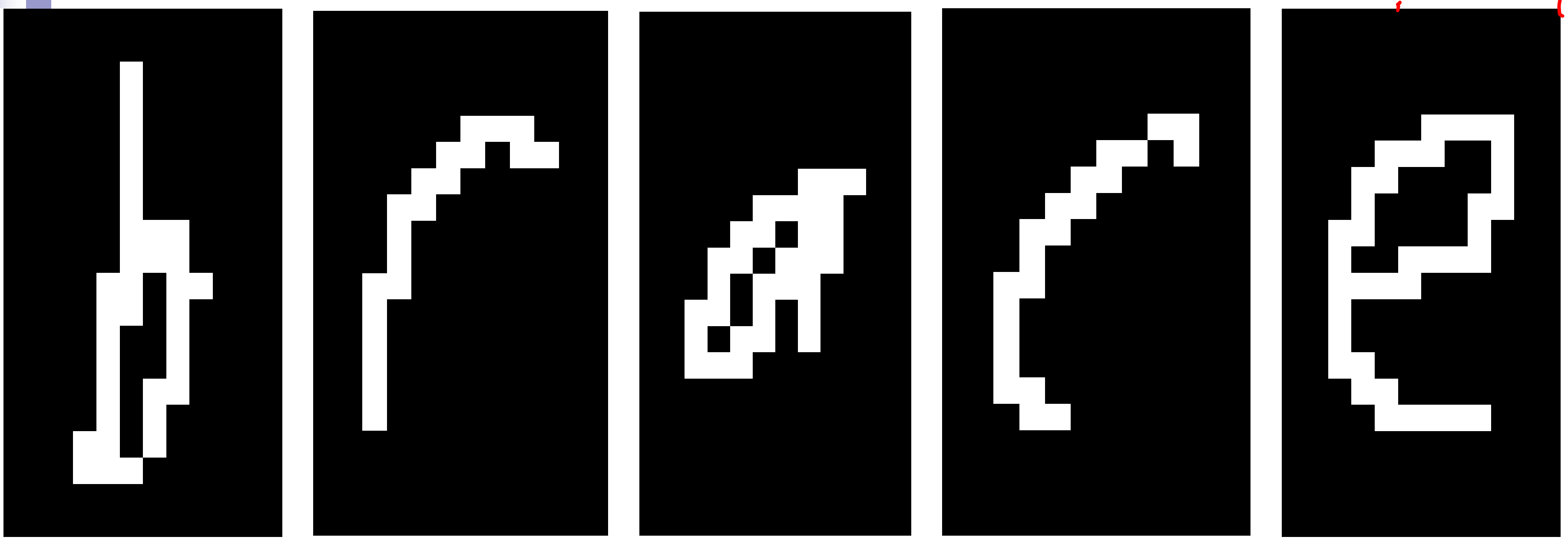
Handwriting recognition 2

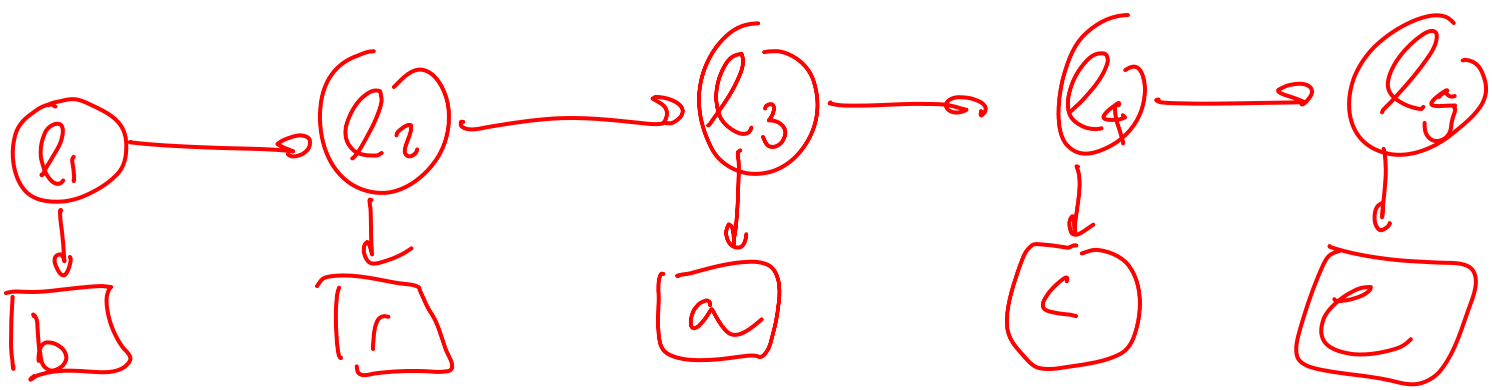
brace

~~zzzzz~~ #

brake classes

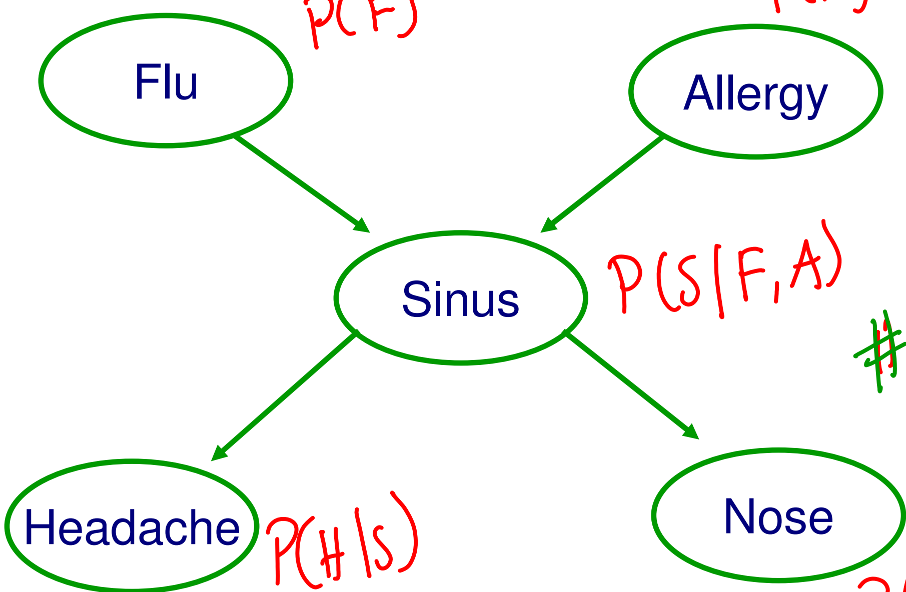
$= (26)^5$





Factored joint distribution - Preview

params? $2^5 - 1 = 31$



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

$P(A) \cdot P(F) \; P(S|FA) \; P(H|S) \cdot P(N|S)$

# params	1	1	4	2	2

$= 10$

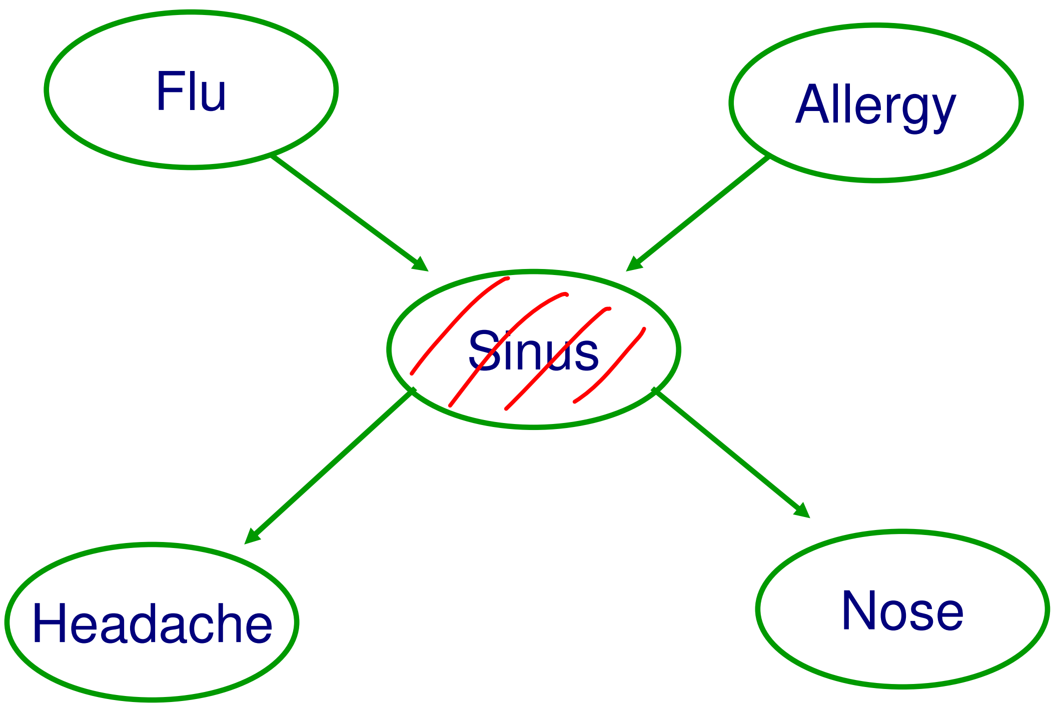
$P(S|FA) =$

A \ F	t	f
t	$P(S{=}t\|Pa)=1$, $P(S{=}f\|Pa)=0$	.5, .5
f	.7, .3	.1, .9

Key: Independence assumptions

$A \perp B \equiv A$ indep of B

not $N \perp F$



$F \perp N \mid S$

$A \perp H \mid S$

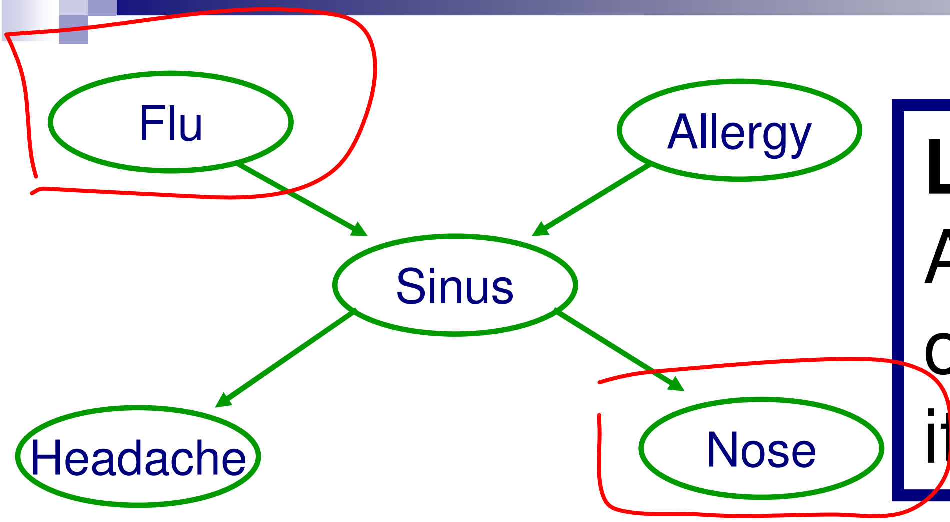
$A \perp N \mid S$

$F \perp H \mid S$

$H \perp N \mid S$

Knowing sinus separates the variables from each other

The independence assumption



Local Markov Assumption:
A variable X is independent of its non-descendants given its parents

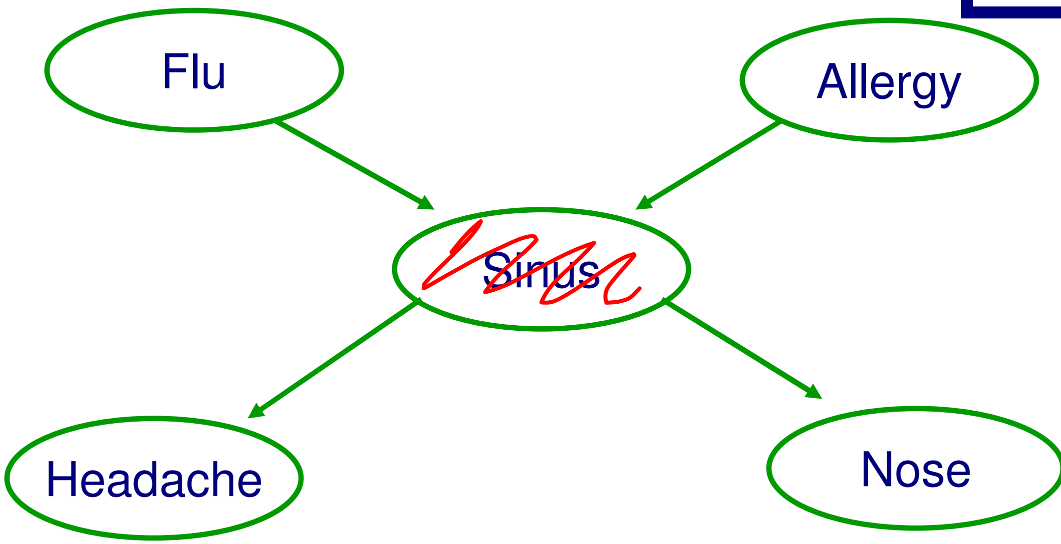
$N \perp \{F, A, H\} \mid S$

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

$F \perp A \mid \emptyset$

Explaining away

Local Markov Assumption:
A variable X is independent of its non-descendants given its parents and only its parents



$\boxed{F \perp A}$

Suppose $S = t$

$P(A=t \mid S=t) > P(A=t)$

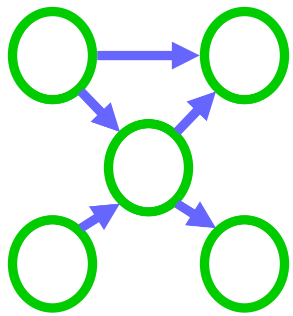
$P(A=t \mid S=t, F=t)$

$< P(A=t \mid S=t)$

not $F \perp A \mid S$

The Representation Theorem – Joint Distribution to BN

BN:



Encodes independence assumptions

If conditional independencies in BN are subset of conditional independencies in *P*

real world

Obtain

Joint probability distribution:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

can represent the real world

A general Bayes net

- Set of random variables

- Directed acyclic graph
  - Encodes independence assumptions

- CPTs

- Joint distribution:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

How many parameters in a BN?

- Discrete variables $X_1, \ldots, X_n$
- Graph
  - Defines parents of X_i, $\mathbf{Pa}_{X_i}$
- CPTs – $P(X_i | \mathbf{Pa}_{X_i})$

Another example

- Variables:
  - B – Burglar
  - E – Earthquake
  - A – Burglar alarm
  - N – Neighbor calls
  - R – Radio report
- Both burglars and earthquakes can set off the alarm
- If the alarm sounds, a neighbor may call
- An earthquake may be announced on the radio

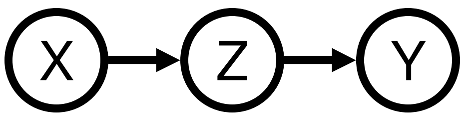
Independencies encoded in BN

- We said: All you need is the local Markov assumption
  - $(X_i \perp NonDescendants_{Xi} \mid \mathbf{Pa}_{Xi})$
- But then we talked about other (in)dependencies
  - e.g., explaining away
- What are the independencies encoded by a BN?
  - Only assumption is local Markov
  - But many others can be derived using the algebra of conditional independencies!!!

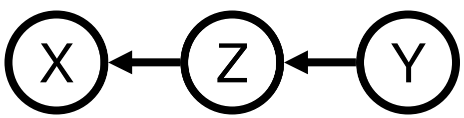
Understanding independencies in BNs – BNs with 3 nodes

Local Markov Assumption:
A variable X is independent of its non-descendants given its parents

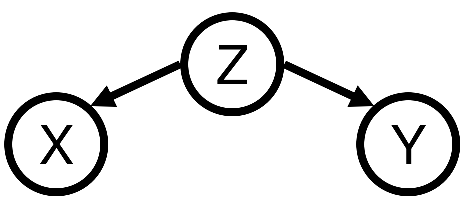
Indirect causal effect:



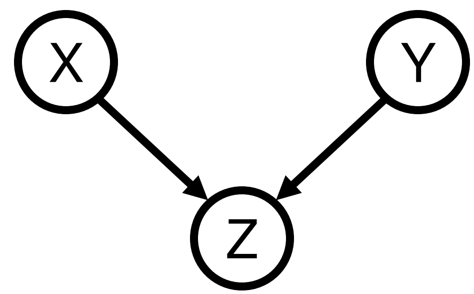
Indirect evidential effect:



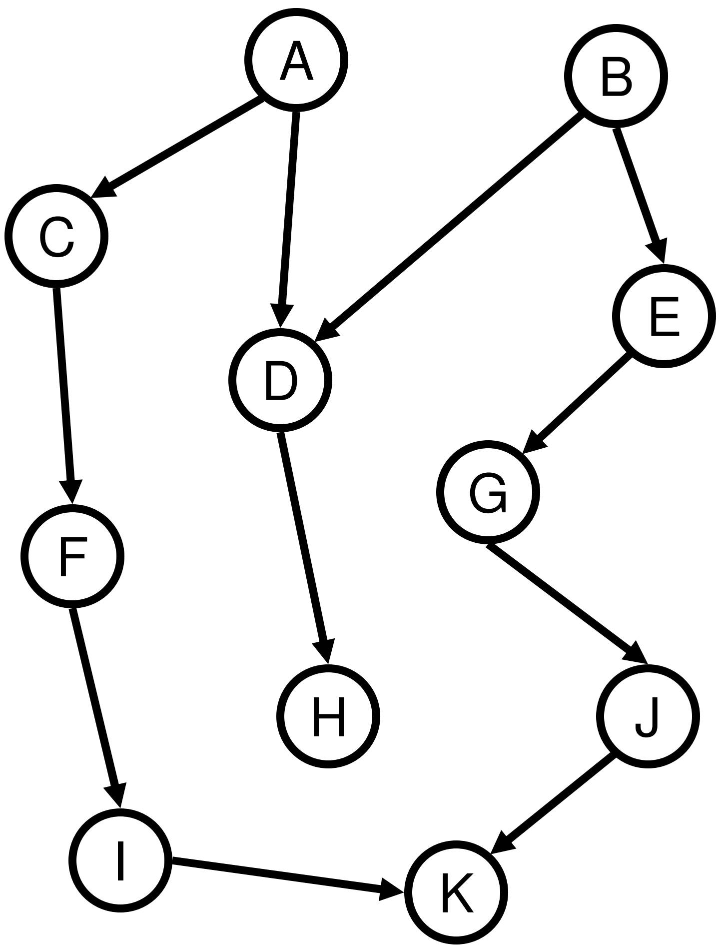
Common cause:



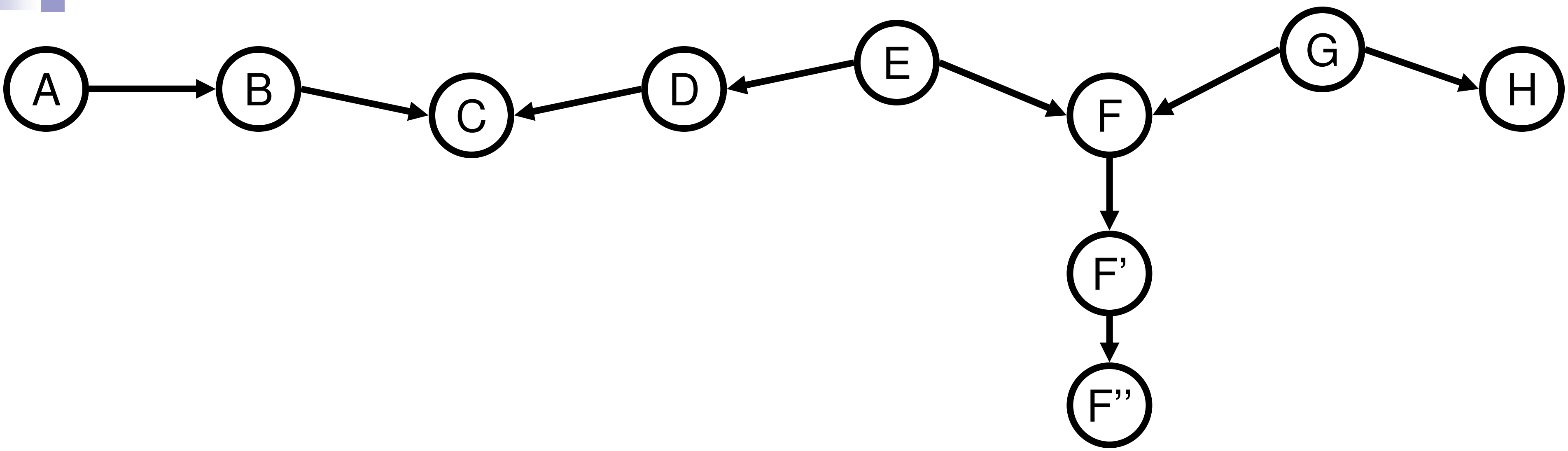
Common effect:



Understanding independencies in BNs – Some examples



An active trail – Example



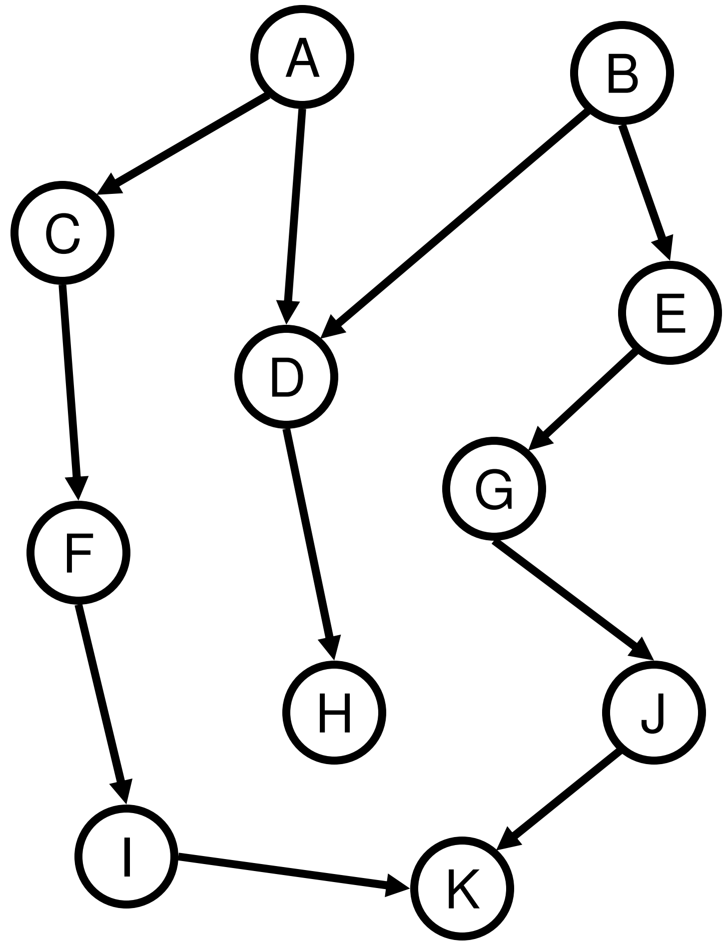
When are A and H independent?

Active trails formalized

- A path $X_1 - X_2 - \cdots - X_k$ is an **active trail** when variables $\boldsymbol{O} \subseteq \{X_1,\ldots,X_n\}$ are observed if for each consecutive triplet in the trail:
  - $X_{i-1} \rightarrow X_i \rightarrow X_{i+1}$, 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**

Active trails and independence?

- **Theorem**: Variables $\mathbf{X_i}$ **and** $\mathbf{X_j}$ **are independent given** $\boldsymbol{Z}\subseteq\{X_1,\ldots,X_n\}$ if the is **no active trail** between X_i and X_j when variables $\boldsymbol{Z}\subseteq\{X_1,\ldots,X_n\}$ are observed



The BN Representation Theorem

If conditional independencies in BN are subset of conditional independencies in *P*

Obtain

Joint probability distribution:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

Important because:
Every *P* has at least one BN structure *G*

If joint probability distribution:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

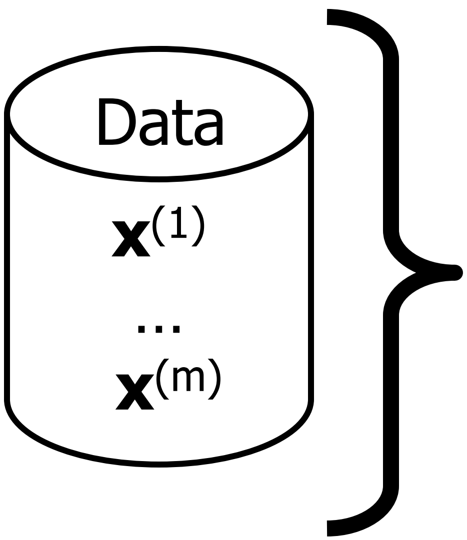
Obtain

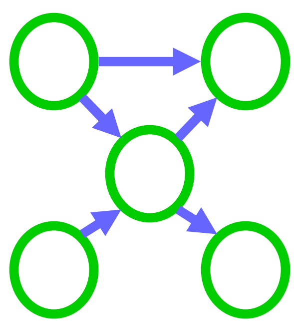
Then conditional independencies in BN are subset of conditional independencies in *P*

Important because:
Read independencies of *P* from BN structure *G*

Learning Bayes nets

	Known structure	Unknown structure
Fully observable data		
Missing data		





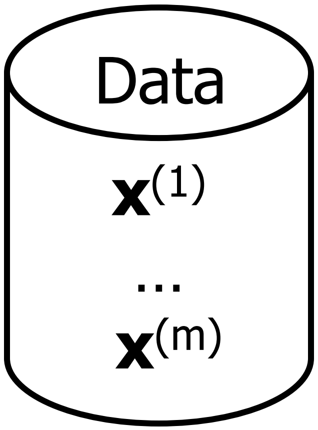
structure

+

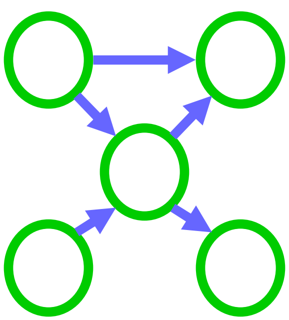
CPTs – $P(X_i | \mathbf{Pa}_{X_i})$

parameters

Learning the CPTs



For each discrete variable X_i

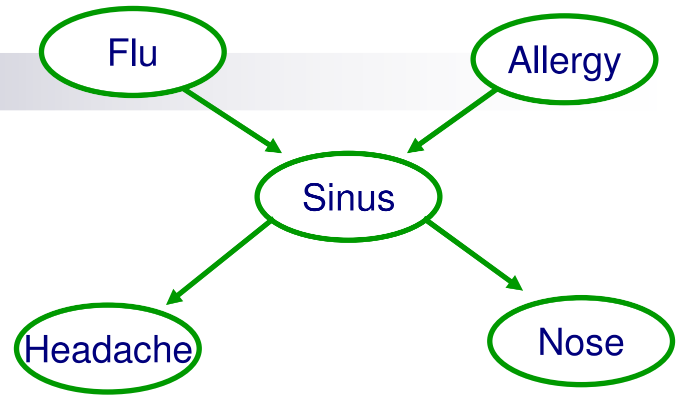


MLE: $$P(X_i = x_i \mid X_j = x_j) = \frac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$$

What you need to know

- Bayesian networks
  - A compact **representation** for large probability distributions
  - Not an algorithm
- Semantics of a BN
  - Conditional independence assumptions
- Representation
  - Variables
  - Graph
  - CPTs
- Why BNs are useful
- Learning CPTs from fully observable data
- Play with applet!!! ☺

General probabilistic inference



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

- Using Bayes rule:

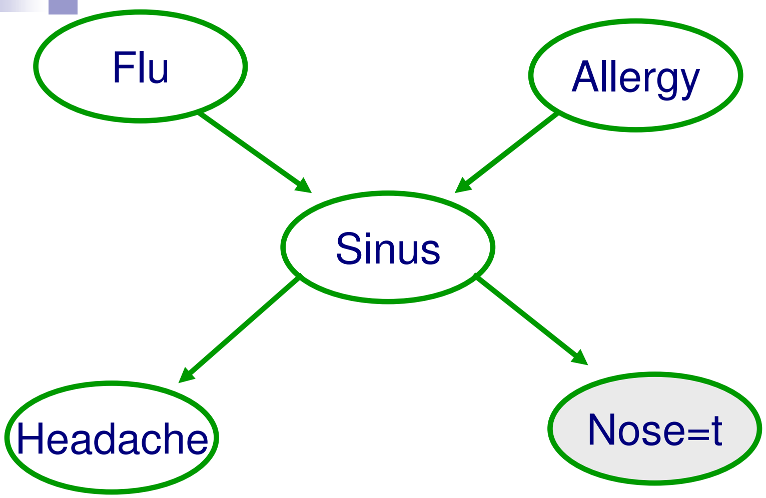
$$P(X \mid e) = \frac{P(X, e)}{P(e)}$$

- Normalization:

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

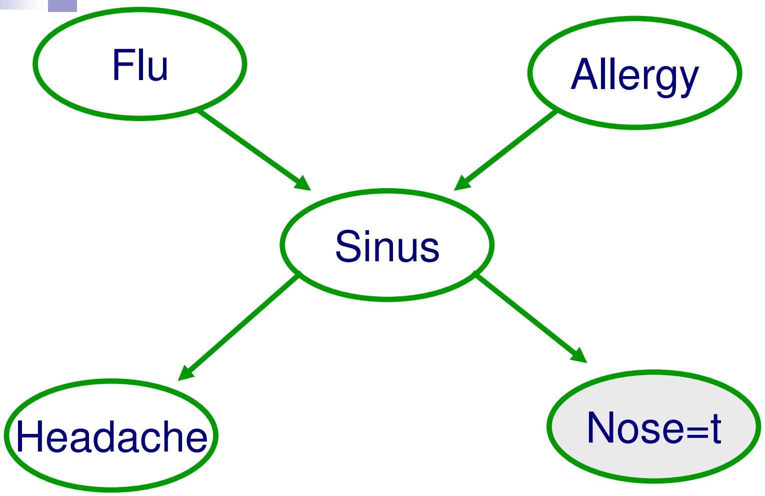
Marginalization

Probabilistic inference example



Inference seems exponential in number of variables!
Actually, inference in graphical models is NP-hard ☹

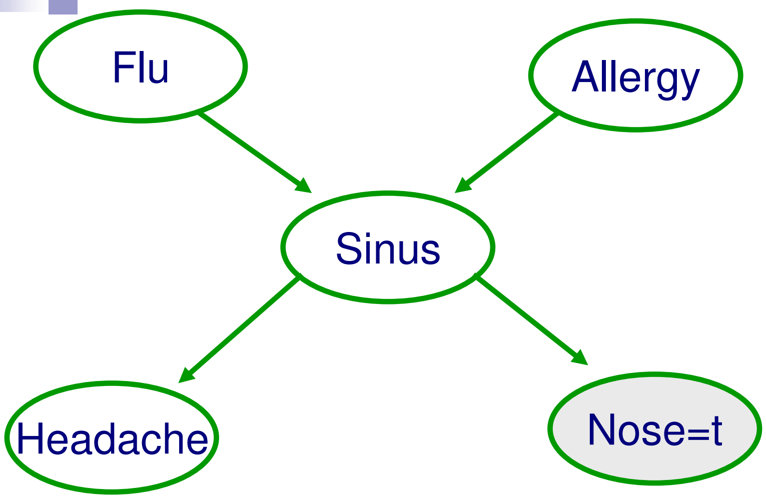
Fast probabilistic inference example – Variable elimination



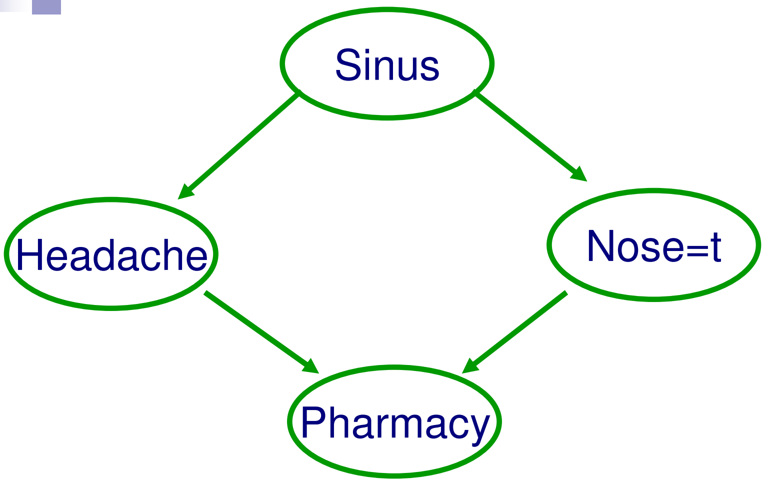
(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity

Understanding variable elimination – Order can make a HUGE difference



Understanding variable elimination – Another example



Variable elimination algorithm

- Given a BN and a query $P(X|e) \propto P(X,e)$
- Instantiate evidence e

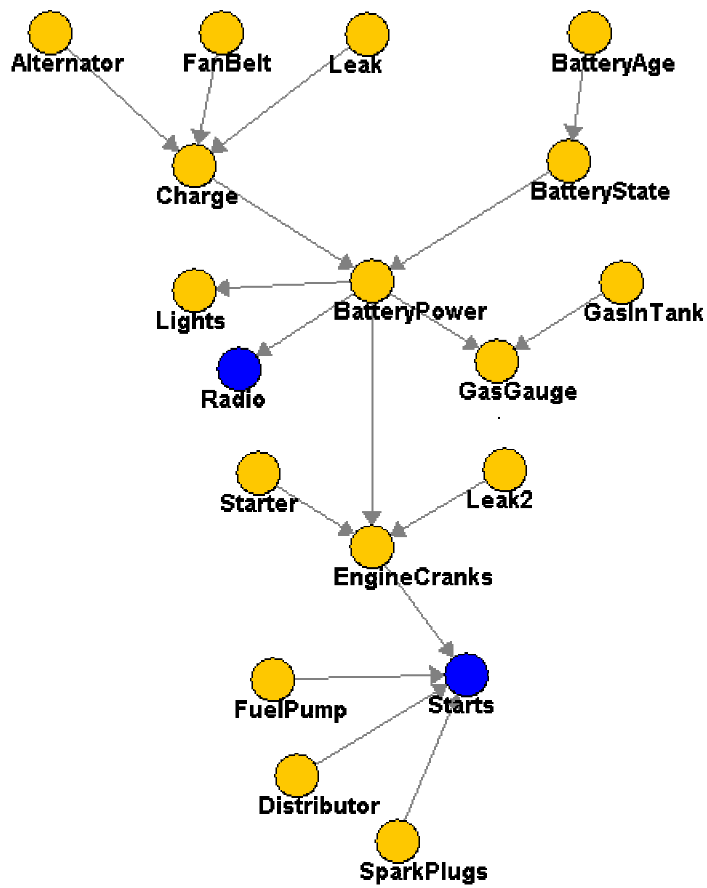
IMPORTANT!!!

- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{X,e\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

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

  - Variable X_i has been eliminated!
- Normalize $P(X,e)$ to obtain $P(X|e)$

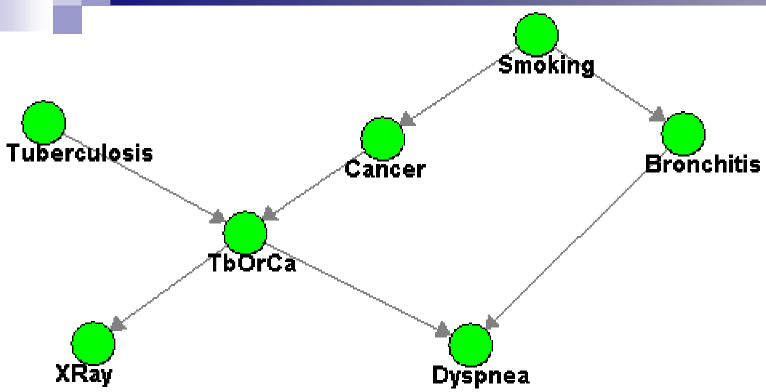
Complexity of variable elimination – (Poly)-tree graphs



Variable elimination order:
Start from “leaves” up –
find topological order, eliminate variables in reverse order

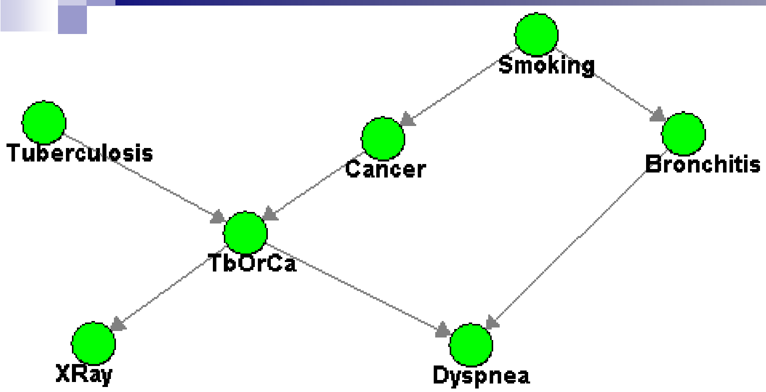
Linear in number of variables!!! (versus exponential)

Complexity of variable elimination – Graphs with loops



Exponential in number of variables in largest factor generated

Complexity of variable elimination – Tree-width



Moralize graph:
Connect parents into a clique and remove edge directions

Complexity of VE elimination:
("Only") exponential in tree-width
Tree-width is maximum node cut +1

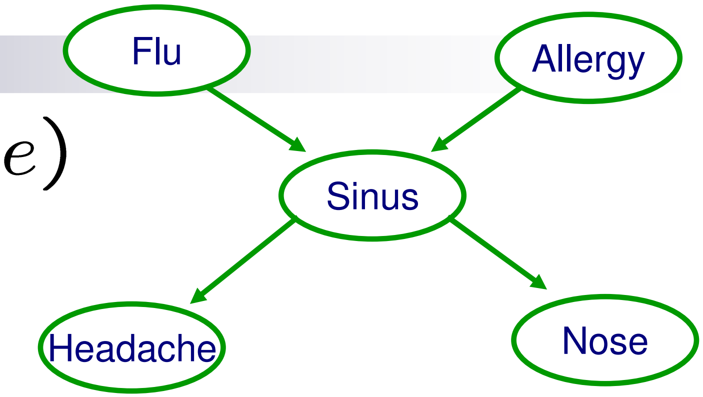
Example: Large tree-width with small number of parents

Compact representation $\not\Rightarrow$ Easy inference ☹

Choosing an elimination order

- Choosing best order is NP-complete
  - Reduction from MAX-Clique
- Many good heuristics (some with guarantees)
- Ultimately, can't beat NP-hardness of inference
  - Even optimal order can lead to exponential variable elimination computation
- In practice
  - Variable elimination often very effective
  - Many (many many) approximate inference approaches available when variable elimination too expensive

Most likely explanation (MLE)



- Query: $\operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n \mid e)$

- Using Bayes rule:

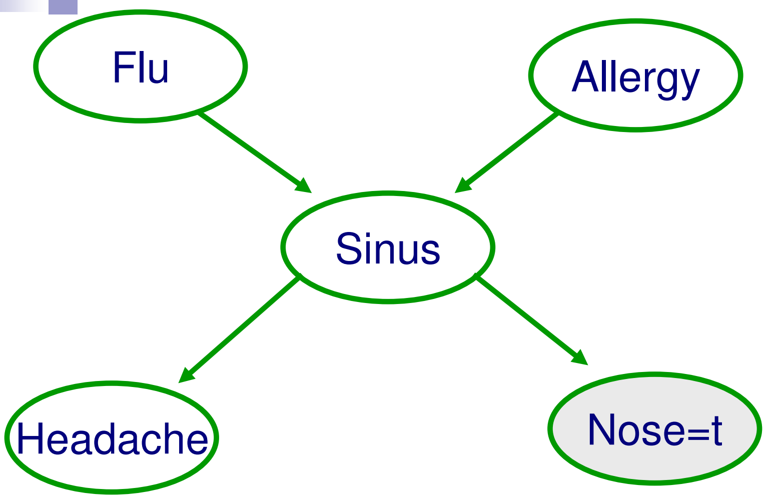
$$\operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n \mid e) = \operatorname*{argmax}_{x_1,\ldots,x_n} \frac{P(x_1,\ldots,x_n,e)}{P(e)}$$

- Normalization irrelevant:

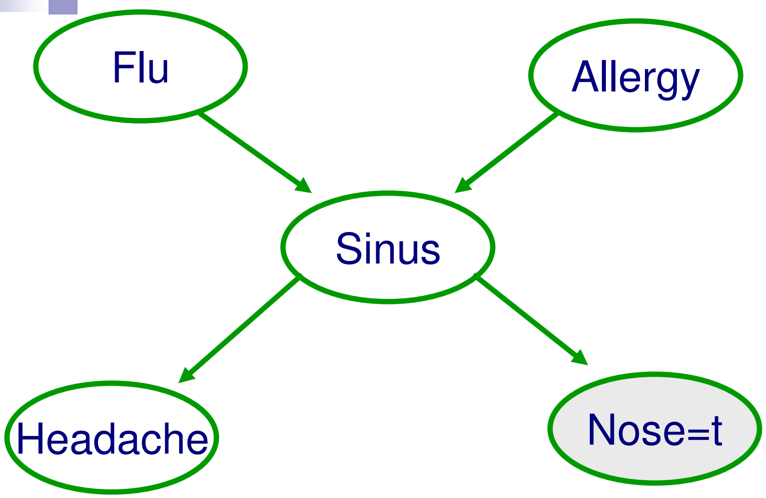
$$\operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n \mid e) = \operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,e)$$

Max-marginalization

Example of variable elimination for MLE – Forward pass



Example of variable elimination for MLE – Backward pass



MLE Variable elimination algorithm – Forward pass

- Given a BN and a MLE query $\max_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,e)$
- Instantiate evidence e
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{e\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \max_{x_i} \prod_{j=1}^{k} f_j$$

  - Variable X_i has been eliminated!

MLE Variable elimination algorithm – Backward pass

- $\{x_1^*, \ldots, x_n^*\}$ will store maximizing assignment
- For i = n to 1, If $X_i \notin \{e\}$
  - Take factors $f_1, \ldots, f_k$ used when X_i was eliminated
  - Instantiate $f_1, \ldots, f_k$, with $\{x_{i+1}^*, \ldots, x_n^*\}$
    - Now each f_j depends only on X_i
  - Generate maximizing assignment for X_i:

$$x_i^* \in \operatorname*{argmax}_{x_i} \prod_{j=1}^{k} f_j$$

What you need to know

- Bayesian networks
  - A useful compact **representation** for large probability distributions
- Inference to compute
  - Probability of X given evidence e
  - Most likely explanation (MLE) given evidence e
  - Inference is NP-hard
- Variable elimination algorithm
  - Efficient algorithm ("only" exponential in tree-width, not number of variables)
  - Elimination order is important!
  - Approximate inference necessary when tree-width to large
    - not covered this semester
  - Only difference between probabilistic inference and MLE is "sum" versus "max"