Required Readings from Koller & Friedman:

Representation: 2.1, 2.2

Inference: 5.1, 6.1, 6.2, 6.7.1

Optional:

2.3, 5.2, 5.3, 6.3, 6.7.2

Bayesian Networks – Representation (cont.) Inference

Machine Learning – 10701/15781

Carlos Guestrin

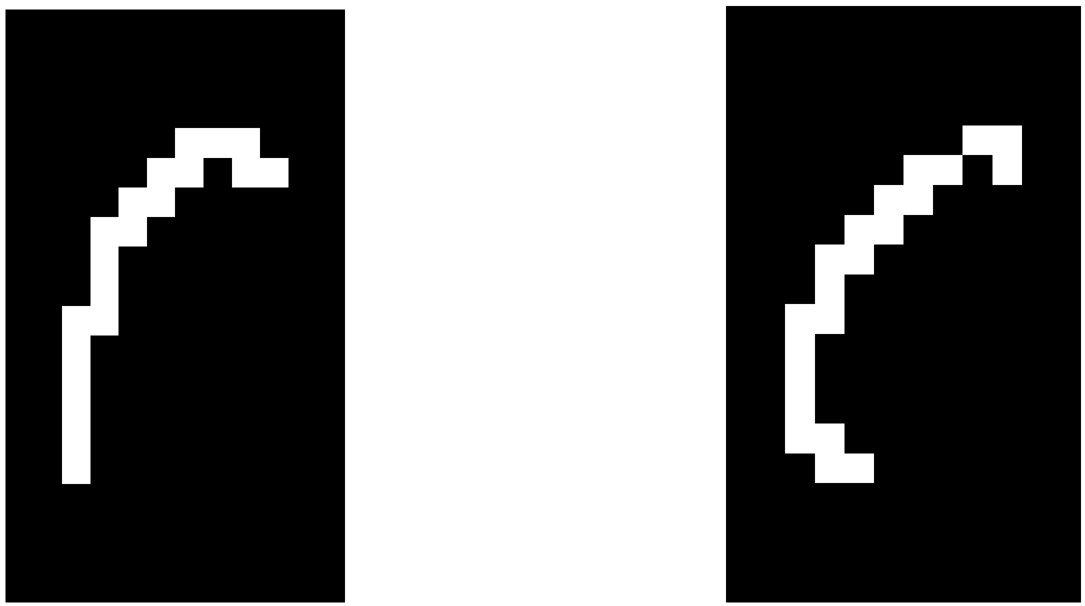
Carnegie Mellon University

March 22st, 2006

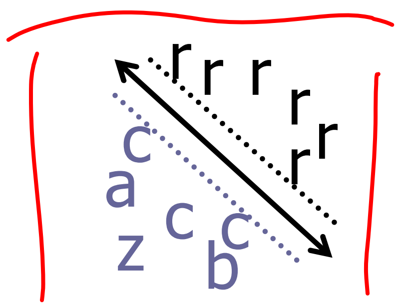
Announcements

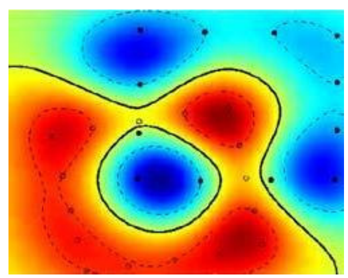
- One page project proposal due now
- We'll go over midterm in this week's recitation
- Homework 4 out later today, due April 5^{th}
  - two weeks from today

Handwriting recognition

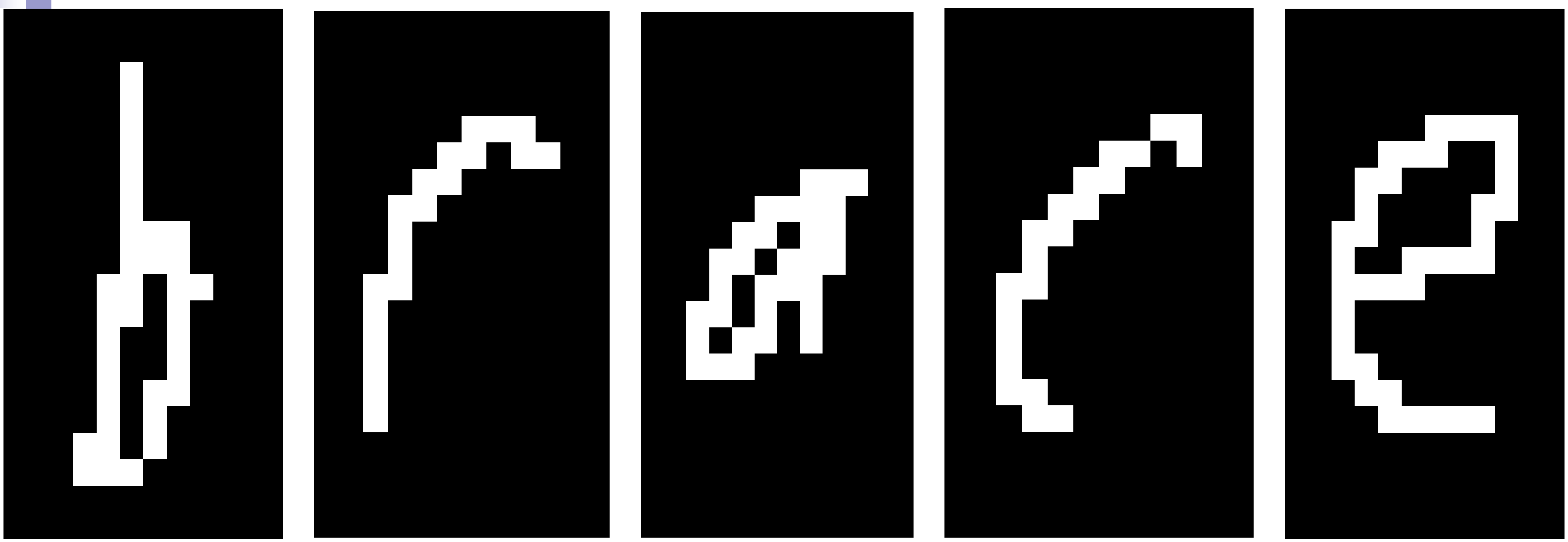


Character recognition, e.g., kernel SVMs



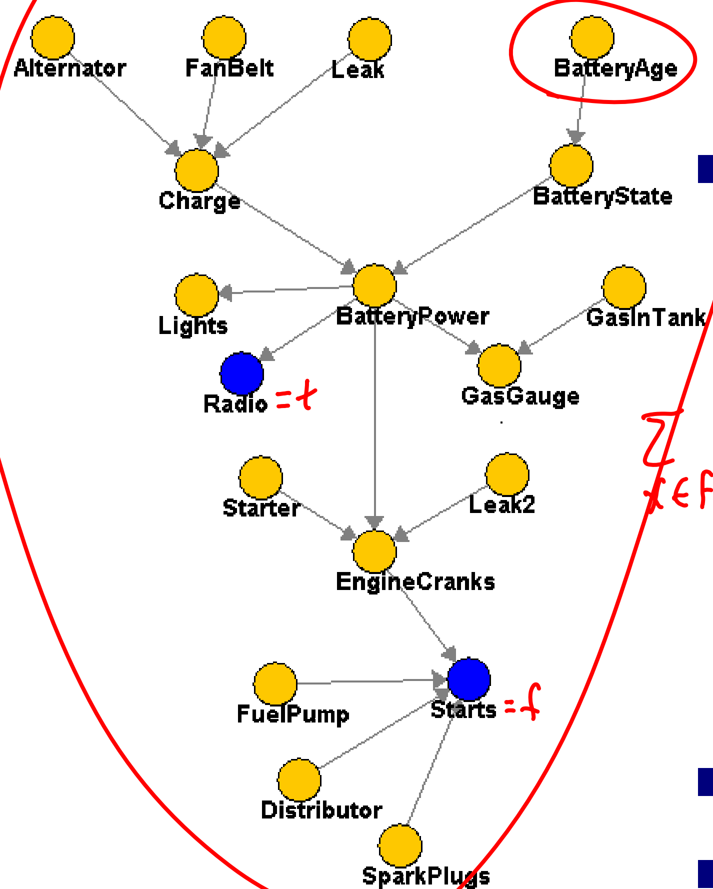


Handwriting recognition 2



- context
- examples not i.i.d.
+ correlations between labels !! ☺

Car starts BN



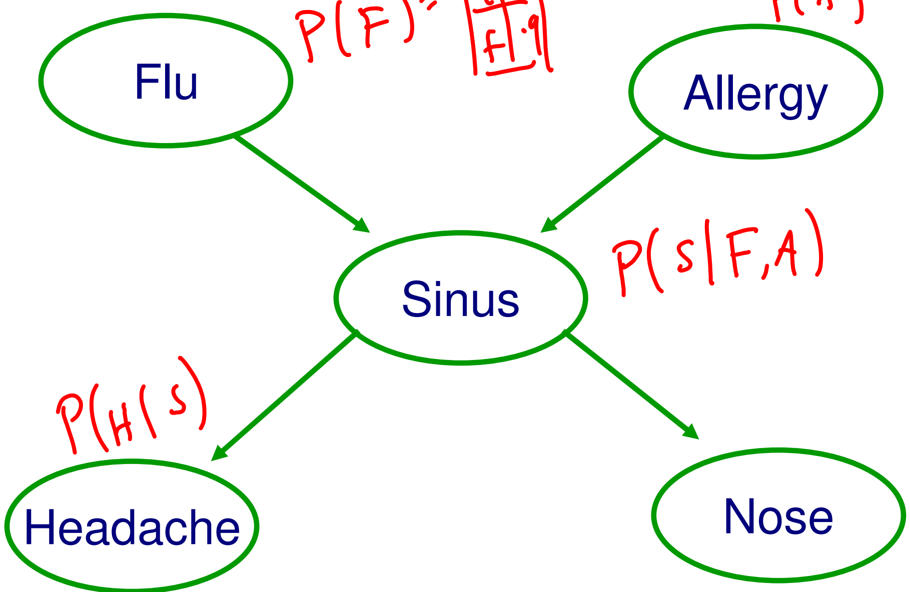
- 18 binary attributes
- Inference
  - P(BatteryAge|Starts=f) $\propto P(BA, S=f)$

Marginalization

$$\sum_{x \in f} P(BA, F=x, S=f) = P(BA, S=t)$$

- 2^{18} terms, why so fast?
- Not impressed?
  - HailFinder BN – more than 3^{54} = 58149737003040059690390169 terms

Factored joint distribution - Preview



$P(F) =$

t	.1
f	.9

$P(A)$

$P(S|F,A)$

$P(H|S)$

$P(N|S)$

Indep:

$(F \perp A)$

$(F \perp N \mid S)$

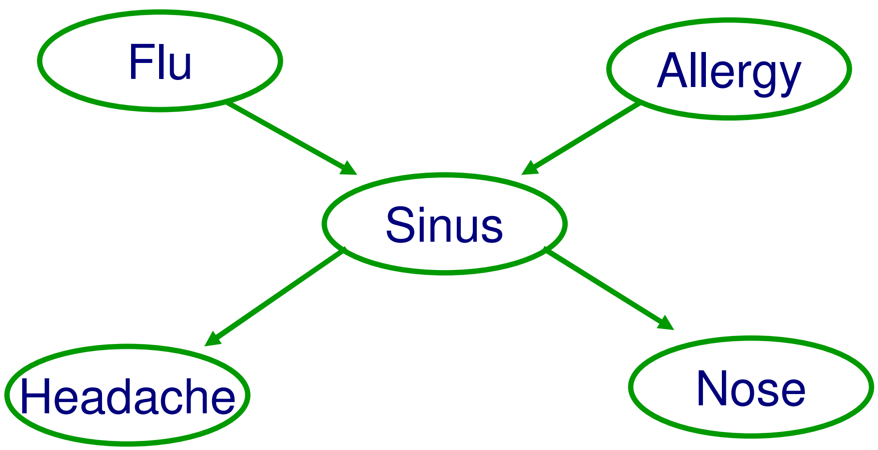
$P(F, A, S, H, N)$
$= P(F) \cdot P(A) \cdot P(S|FA) \cdot P(H|S) \cdot P(N|S)$

$P(N|S) =$

N \ S	t	f
t	.8	.3
f	1 − .8 = .2	1 − .3 = .7

2 parameters

The independence assumption



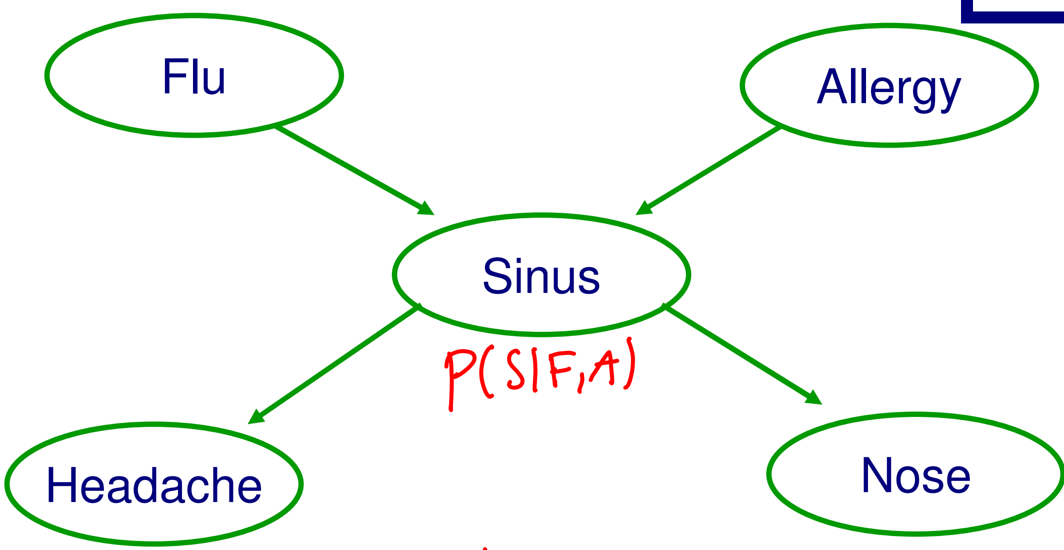
Local Markov Assumption:
A variable X is independent of its non-descendants given its parents

$(F \perp A)$

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

Explaining away

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



$(F \perp A)$ marginally

what if $S=t$

$S=t$ $\quad P(F=t \mid S=t) > P(F=t)$

but

$S=t$ & $A=t$:

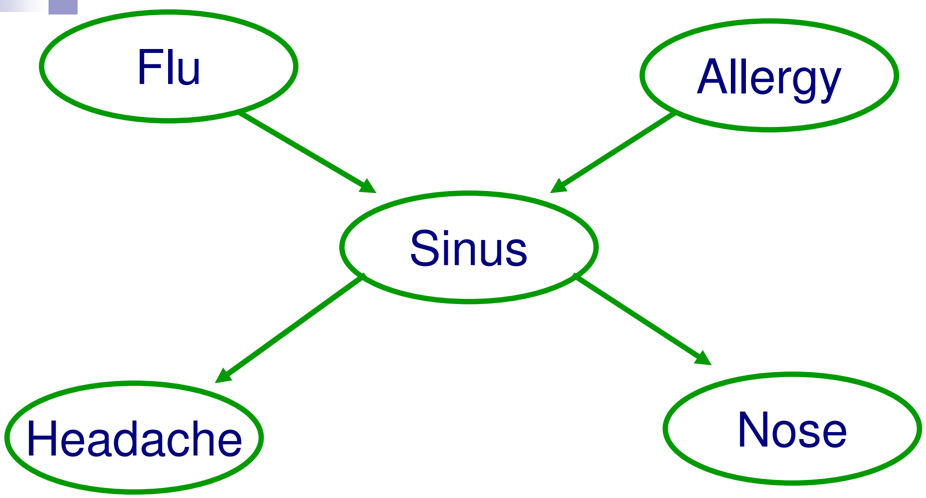
$$P(F=t \mid S=t) > P(F=t \mid S=t, A=t)$$
$$> P(F=t)$$

F not indep. A given S

what if $N=t$

Same explaining away!!

Chain rule & Joint distribution



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

$(F \perp A) \Rightarrow P(A|F) = P(A)$

$(H \perp \{F,A\} | S) \Rightarrow P(H|SFA) = P(H|S)$

$(N \perp \{H,F,A\} | S) \Rightarrow P(N|FAHS) = P(N|S)$

$P(F,A,S,H,N) =$ chain rule no assumptions

$$P(F) \cdot \underbrace{P(A|F)}_{=P(A)} \cdot P(S|FA) \underbrace{P(H|SFA)}_{=P(H|S)} \underbrace{P(N|FASH)}_{=P(N|S)}$$

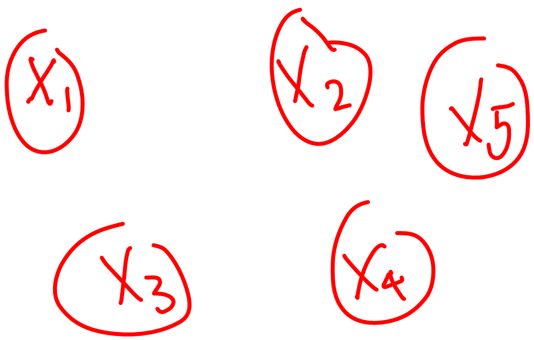
with local Markov Assumption:

$$\Rightarrow P(F)\,P(A)\,P(S|FA)\,P(H|S)\,P(N|S)$$

Two (trivial) special cases

give you some P

Edgeless graph



$(X_1 \perp X_4)$

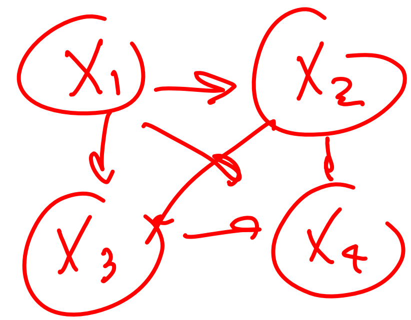
$(X_2 \perp X_3 \mid X_5)$

$\vdots$

only if all vars indep.

always!

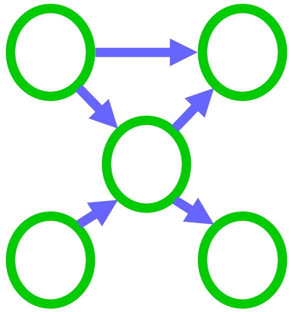
Fully-connected graph



no indep. in graph

The Representation Theorem – Joint Distribution to BN

BN:



Encodes independence assumptions

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

Obtain

Joint probability distribution:

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

P can be represented with BN

$\forall P$ exists at least one BN

Real Bayesian networks applications

it's all about exploiting indep. (problem structure)

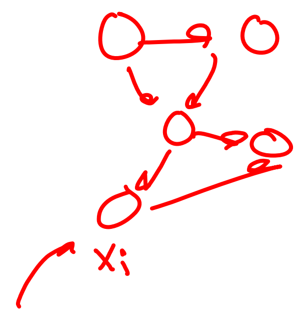
- Diagnosis of lymph node disease
- Speech recognition
- Microsoft office and Windows
  - http://www.research.microsoft.com/research/dtg/
- Study Human genome
- Robot mapping
- Robots to identify meteorites to study
- Modeling fMRI data
- Anomaly detection
- Fault dianosis
- Modeling sensor network data

A general Bayes net

- Set of random variables

  $X_1, X_2, X_3 \dots, X_n$

- Directed acyclic graph
  - Encodes independence assumptions



- CPTs $\longrightarrow$ $P(X_i \mid Pa_{X_i})$

- Joint distribution:

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

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

Another example – Building the BN

- B – Burglar
- E – Earthquake
- A – Burglar alarm
- N – Neighbor calls
- R – Radio report

Independencies encoded in BN

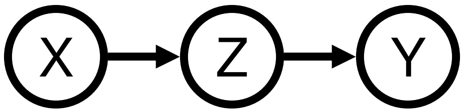
- We said: All you need is the local Markov assumption
  - $(X_i \perp NonDescendants_{Xi} \mid \mathbf{Pa}_{Xi})$
- But then we talked about other (in)dependencies
  - e.g., explaining away

- 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!!!

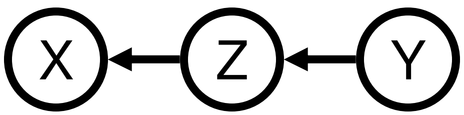
Understanding independencies in BNs – BNs with 3 nodes

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

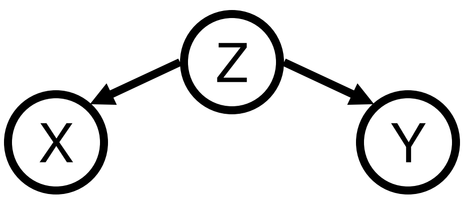
Indirect causal effect:



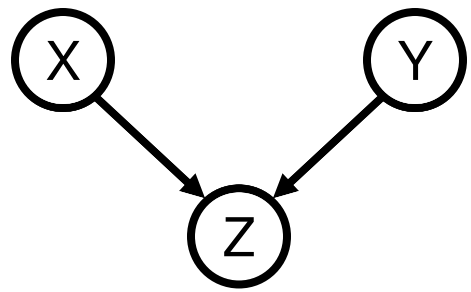
Indirect evidential effect:



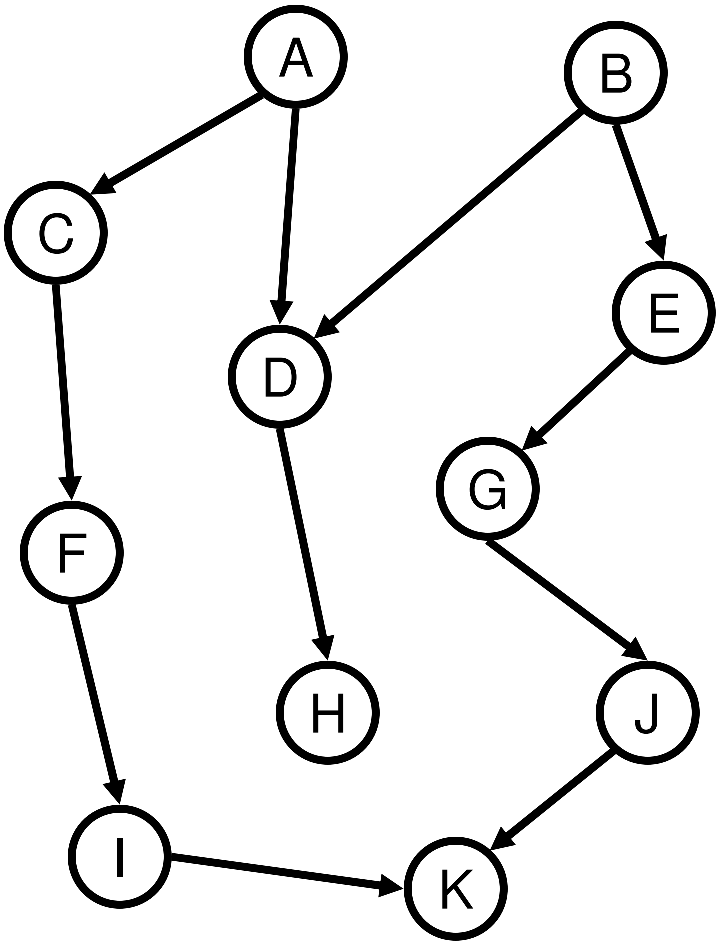
Common cause:



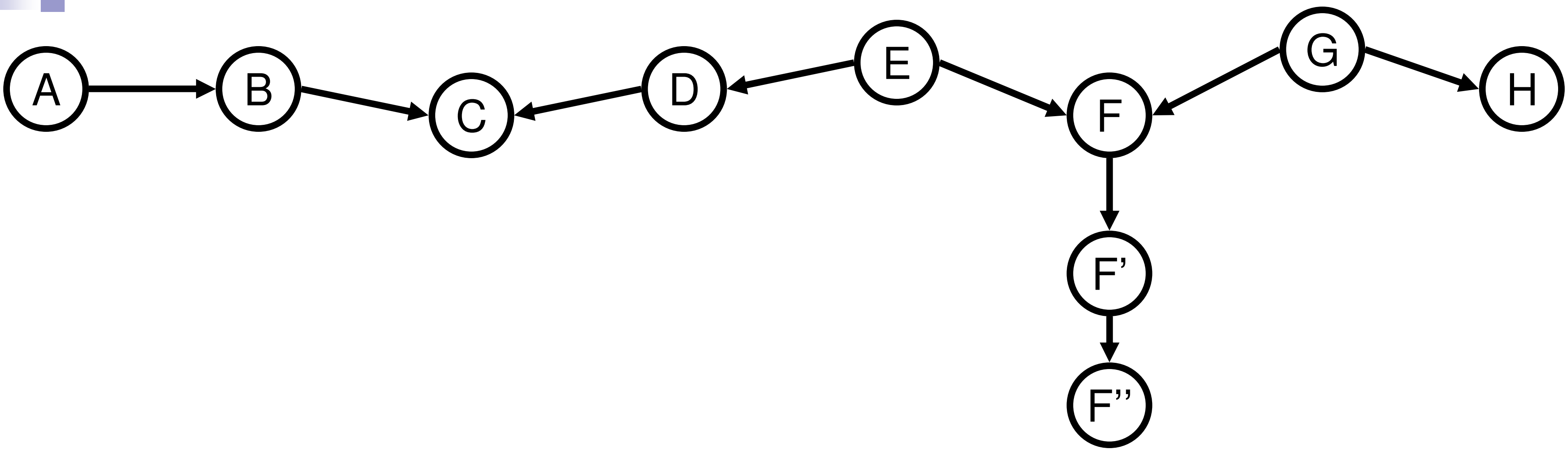
Common effect:



Understanding independencies in BNs – Some examples



An active trail – Example



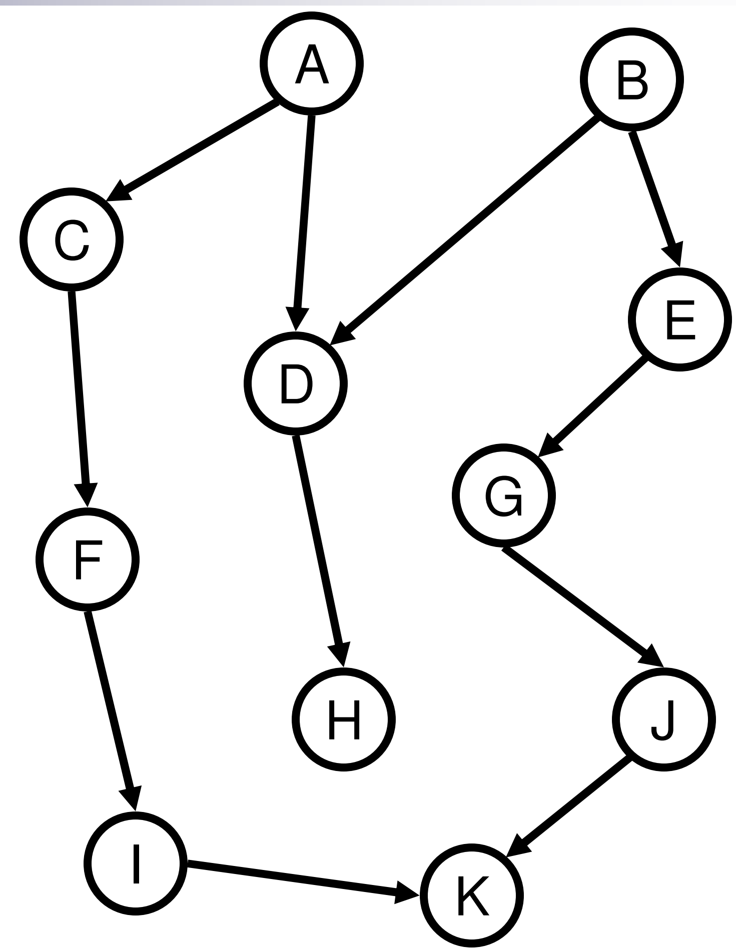
When are A and H independent?

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*

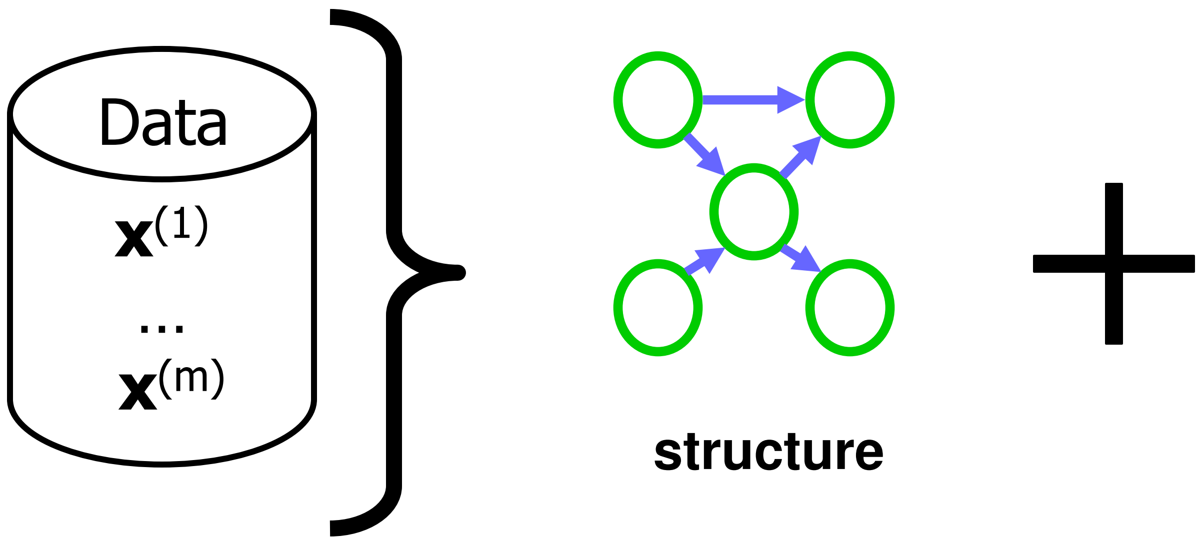
“Simpler” BNs

- A distribution can be represented by many BNs:

- Simpler BN, requires fewer parameters

Learning Bayes nets

	Known structure	Unknown structure
Fully observable data		
Missing data		

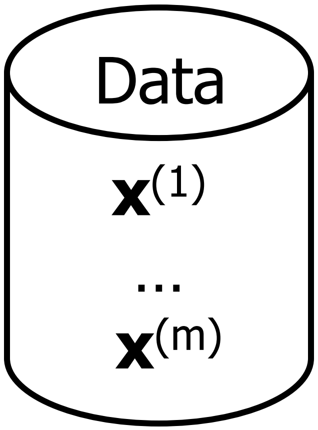


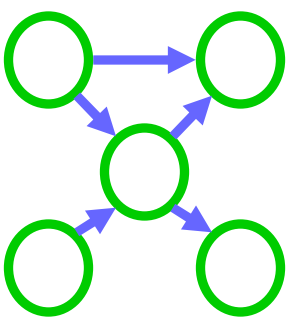
structure

CPTs –
$P(X_i | \mathbf{Pa}_{X_i})$

parameters

Learning the CPTs





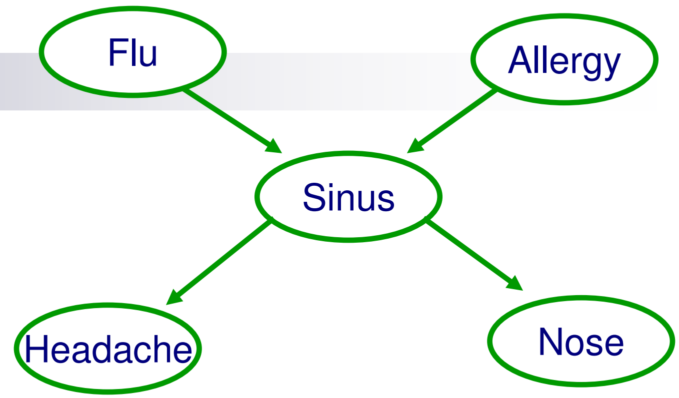
For each discrete variable X_i

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

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: $P(X \mid e)$

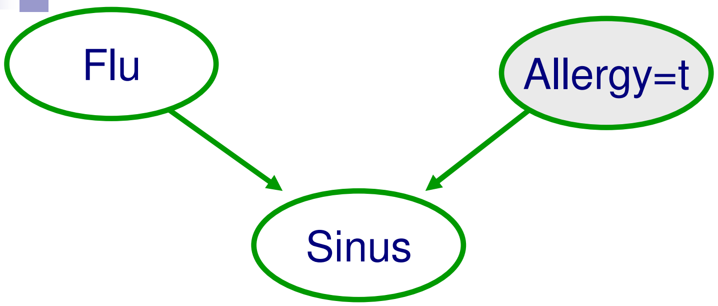
- Using Bayes rule:

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

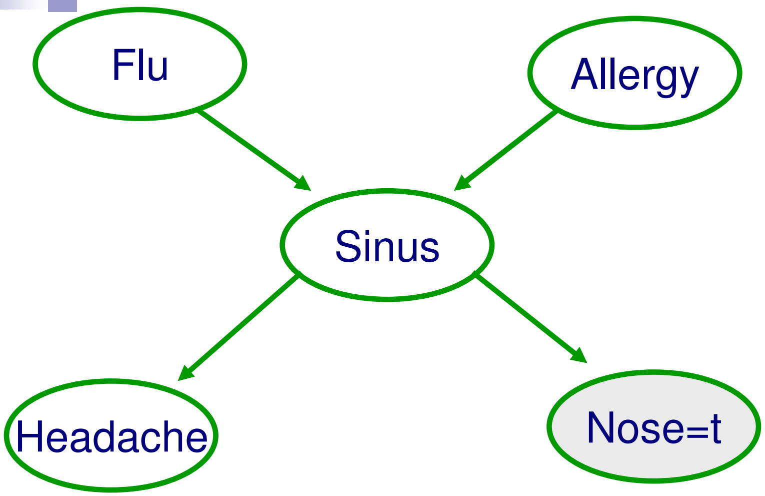
- Normalization:

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

Marginalization



Probabilistic inference example



Inference seems exponential in number of variables!

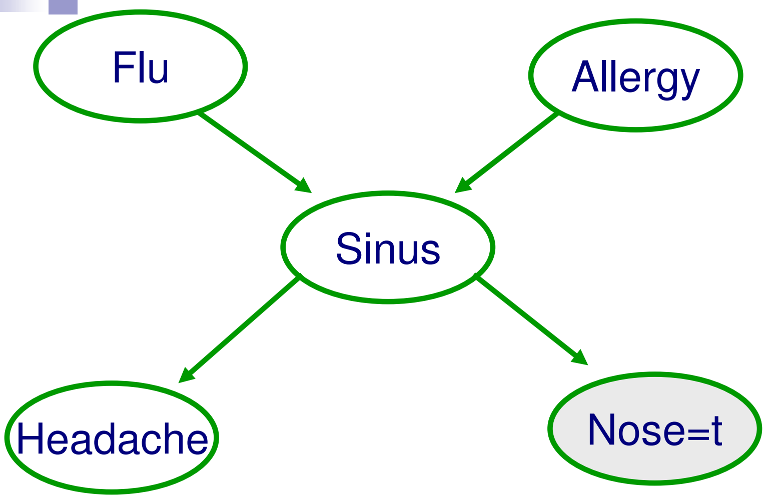
Inference is NP-hard (Actually #P-complete)

- Reduction – 3-SAT

$$(\overline{X}_1 \vee X_2 \vee X_3) \wedge (\overline{X}_2 \vee X_3 \vee X_4) \wedge \ldots$$

Inference unlikely to be efficient in general, but…

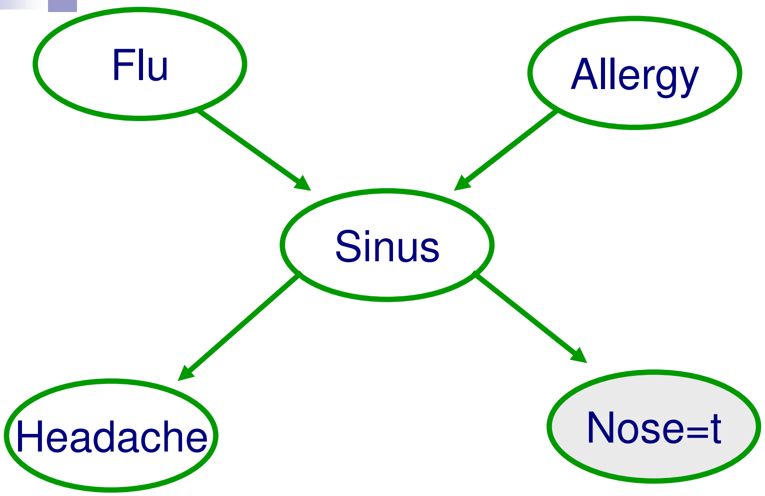
Fast probabilistic inference example – Variable elimination



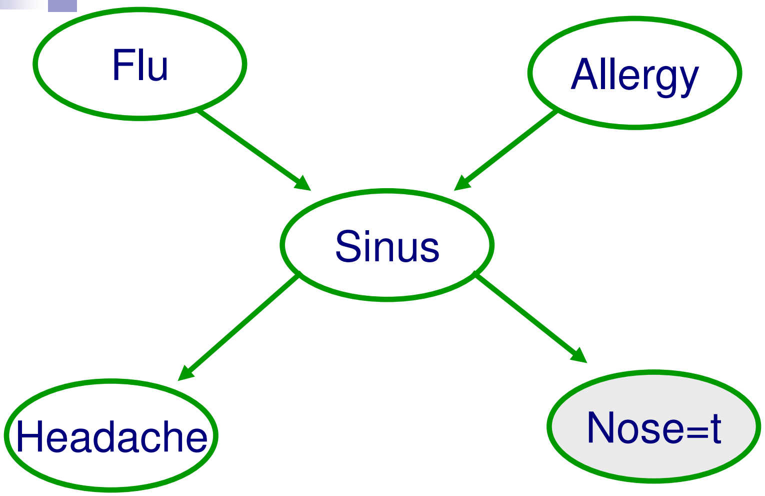
(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity

Understanding variable elimination – Order can make a HUGE difference

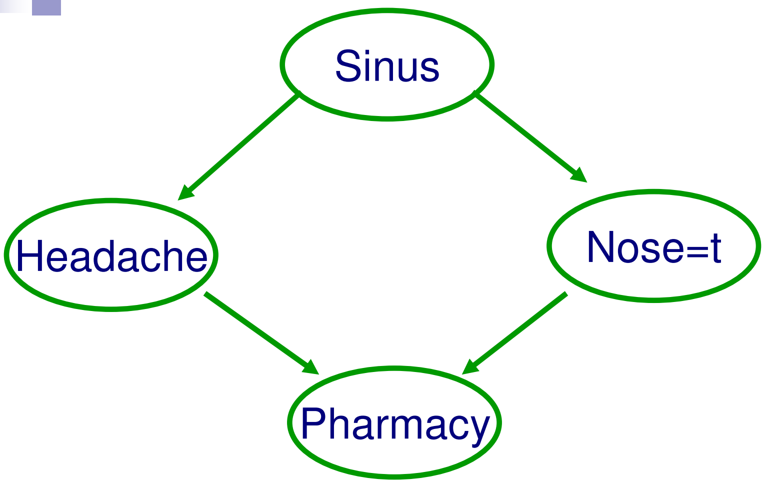


Understanding variable elimination – Intermediate results

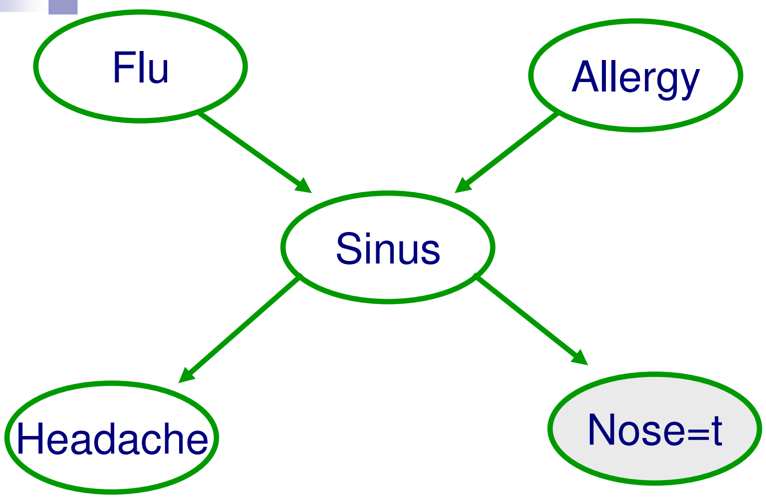


Intermediate results are probability distributions

Understanding variable elimination – Another example



Pruning irrelevant variables



Prune all non-ancestors of query variables

Variable elimination algorithm

- Given a BN and a query $P(X|e) \propto P(X,e)$
- Instantiate evidence e
- Prune non-ancestors of {X,e}

IMPORTANT!!!

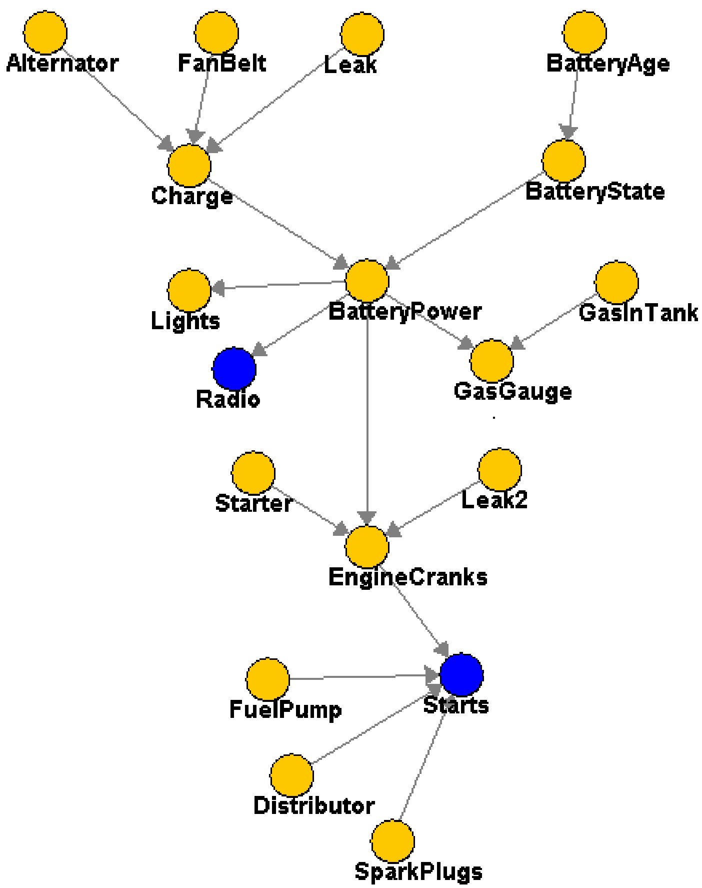
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{X,e\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

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

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

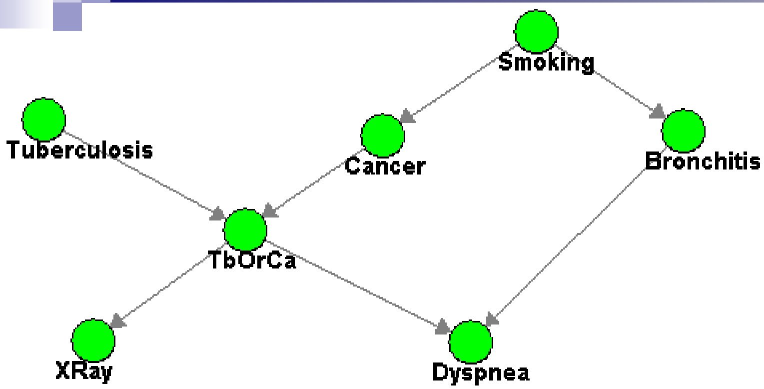
Complexity of variable elimination – (Poly)-tree graphs

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



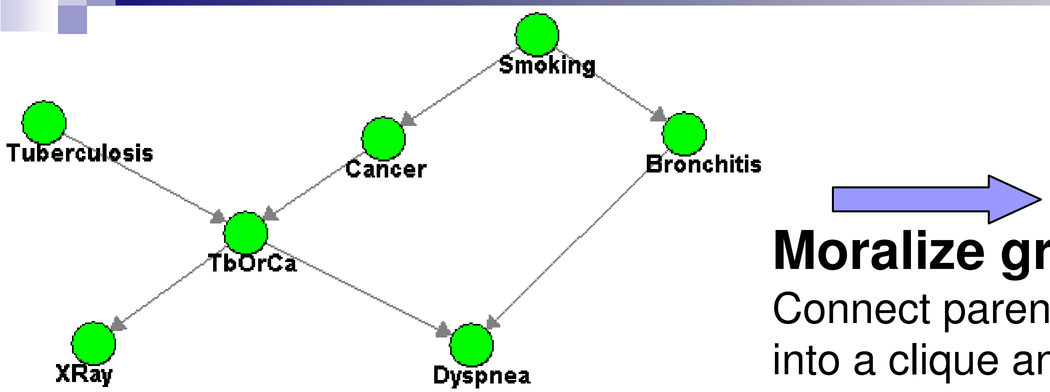
Linear in number of variables!!! (versus exponential)

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

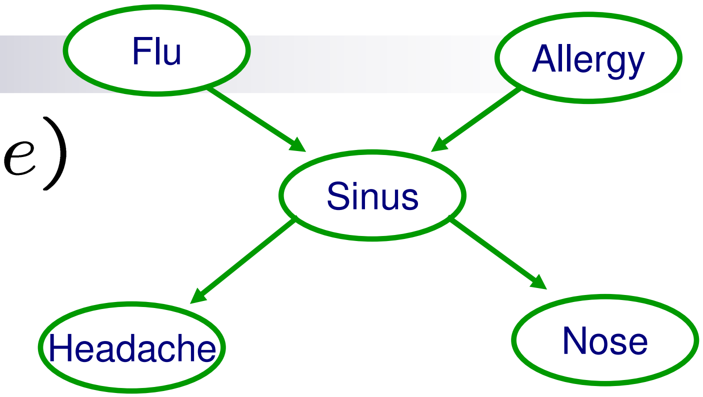
Example: Large tree-width with small number of parents

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: $\underset{x_1,\ldots,x_n}{\text{argmax}}\, P(x_1,\ldots,x_n \mid e)$

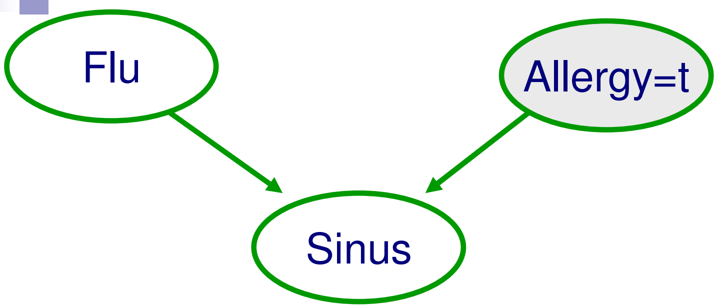
- Using Bayes rule:

$$\underset{x_1,\ldots,x_n}{\text{argmax}}\, P(x_1,\ldots,x_n \mid e) = \underset{x_1,\ldots,x_n}{\text{argmax}}\, \frac{P(x_1,\ldots,x_n,e)}{P(e)}$$

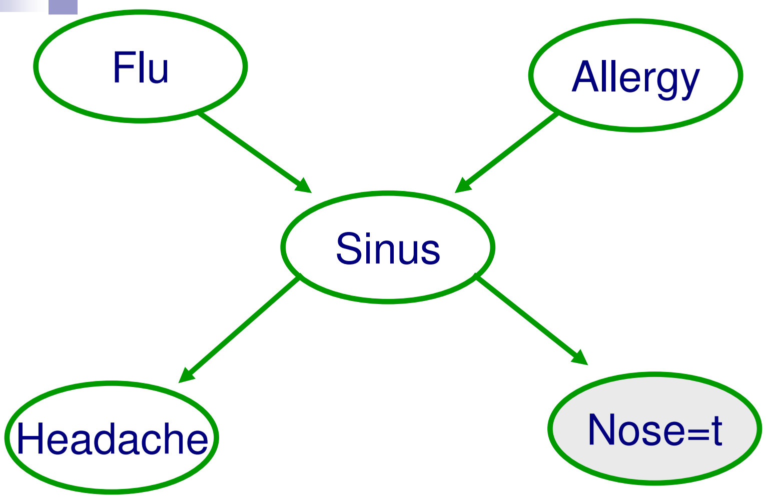
- Normalization irrelevant:

$$\underset{x_1,\ldots,x_n}{\text{argmax}}\, P(x_1,\ldots,x_n \mid e) = \underset{x_1,\ldots,x_n}{\text{argmax}}\, P(x_1,\ldots,x_n,e)$$

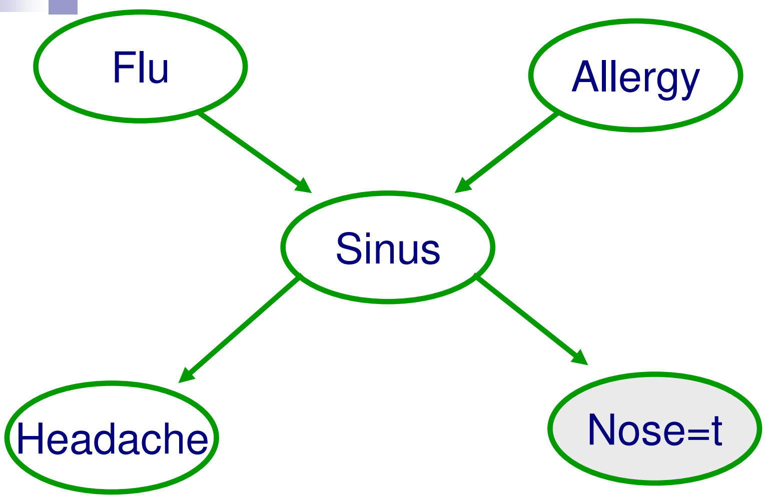
Max-marginalization



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"

Acknowledgements

- JavaBayes applet
  - http://www.pmr.poli.usp.br/ltd/Software/javabayes/Home/index.html