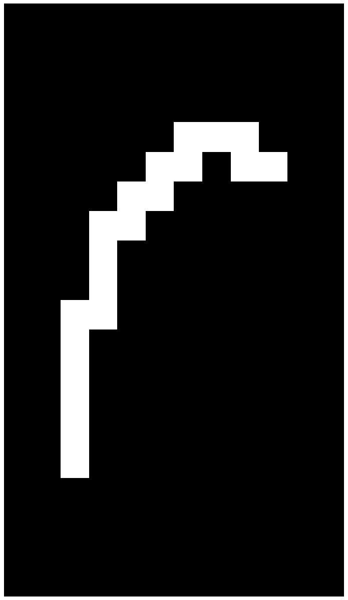
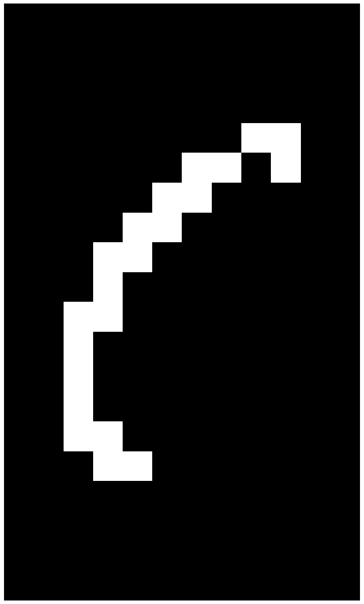
Bayesian Networks – Representation (cont.) Inference

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

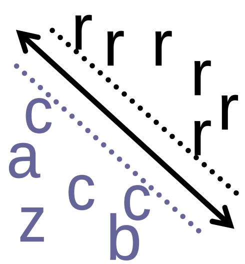
March 21st, 2007

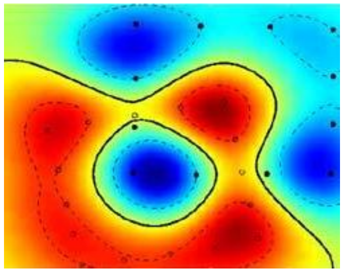
Handwriting recognition





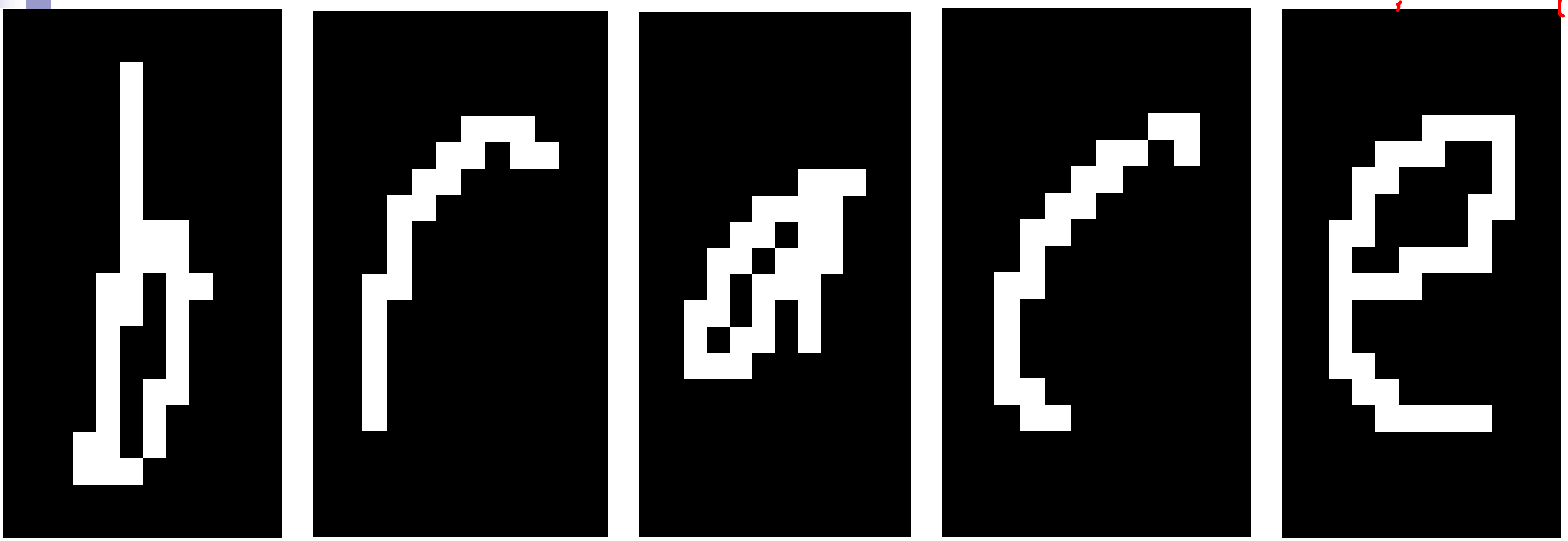
Character recognition, e.g., kernel SVMs

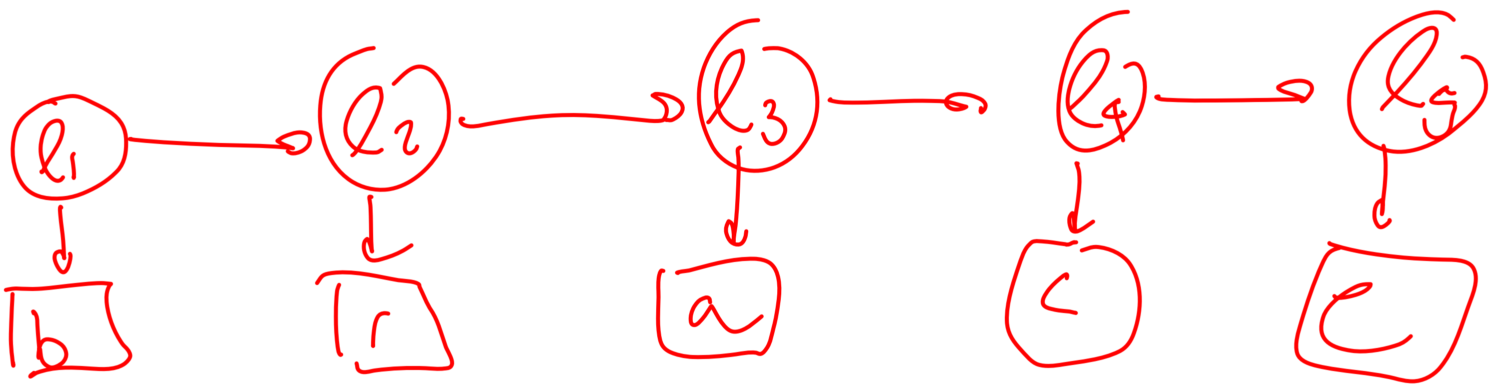




Handwriting recognition 2

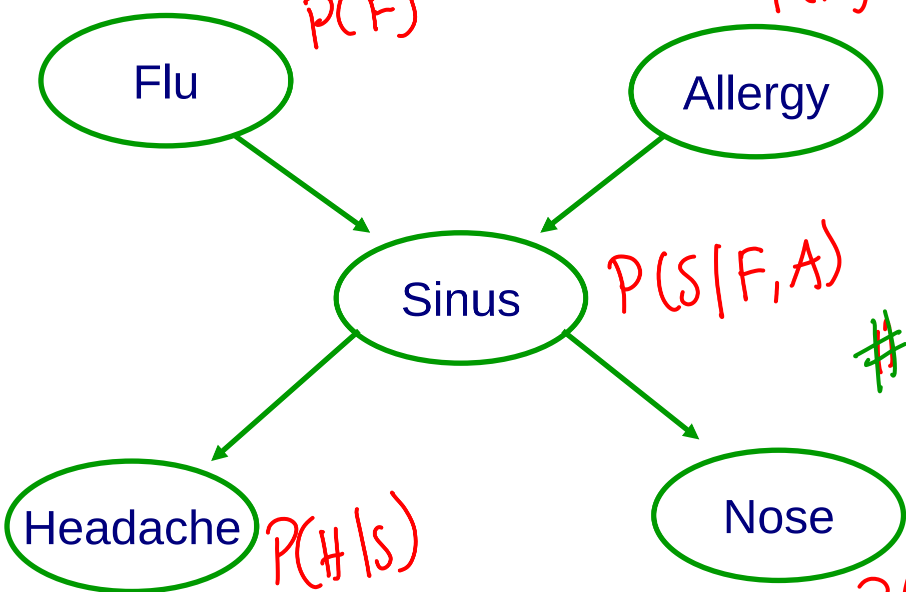
brace
~~zzzzz~~ #
brake classes
$= (26)^5$





Factored joint distribution - Preview

\# params? $2^5 - 1 = 31$



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

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

\# params	1	1	4	2	2

$= 10$

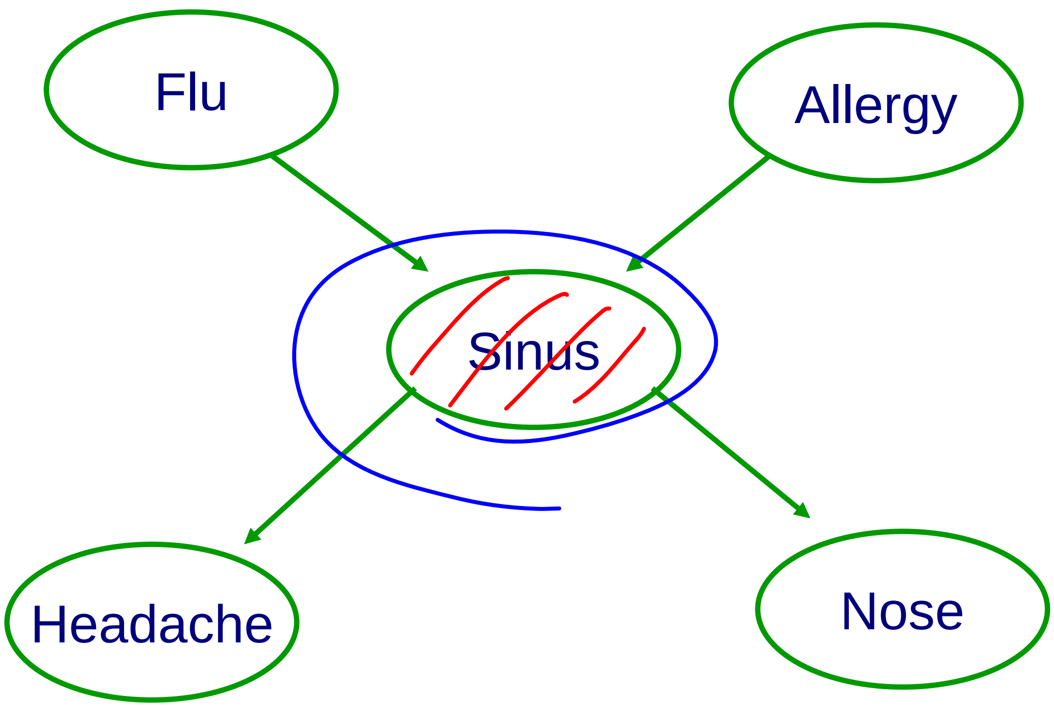
$P(S|FA) =$

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 $N \perp F$



$F \perp N \mid S$

$A \perp H \mid S$

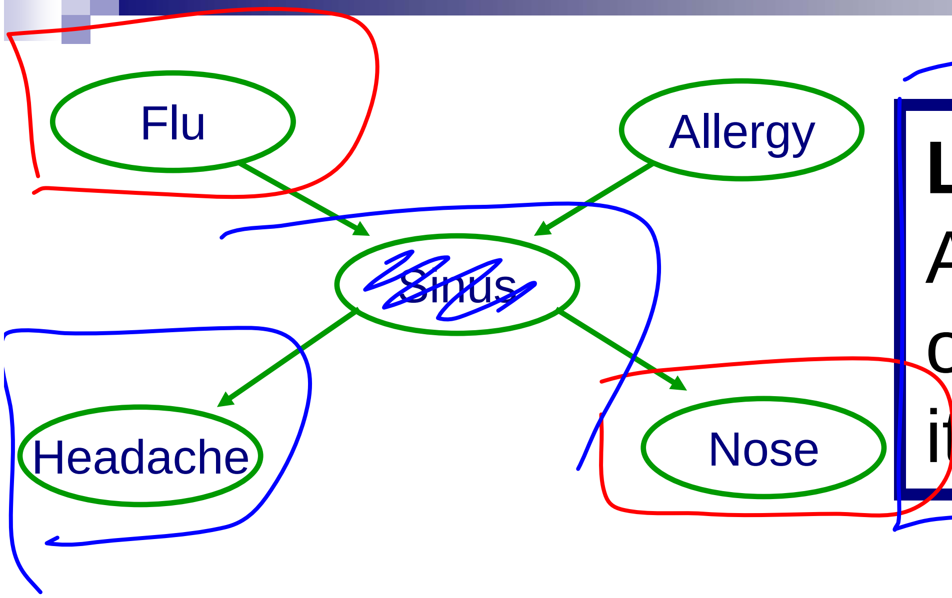
$A \perp N \mid S$

$F \perp H \mid S$

$H \perp N \mid S$

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

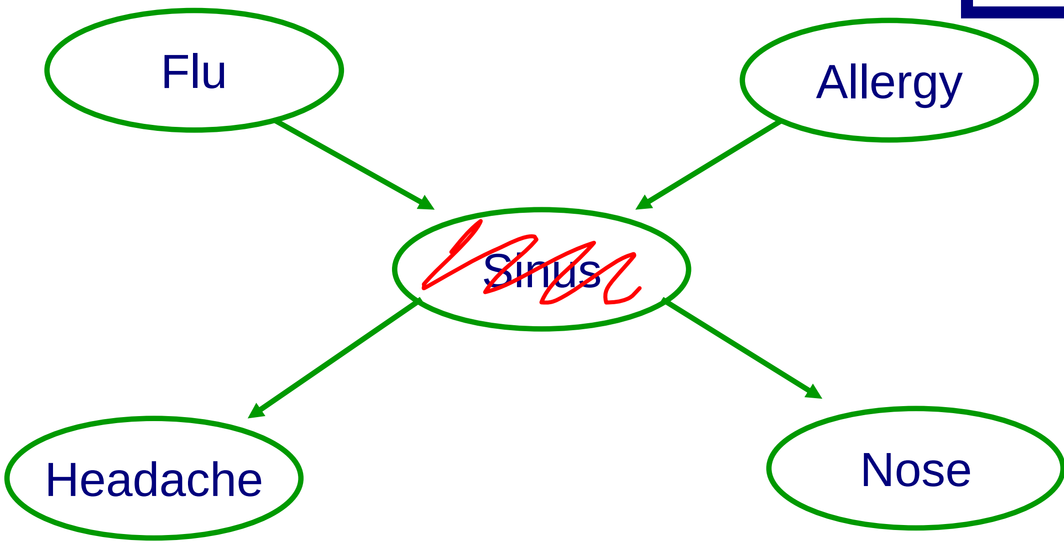
$N \perp \{F, A, H\} \mid S$

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

$F \perp A \mid \emptyset$

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 $S = t$

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

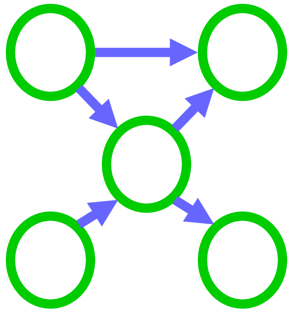
$P(A=t \mid S=t, F=t)$

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

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

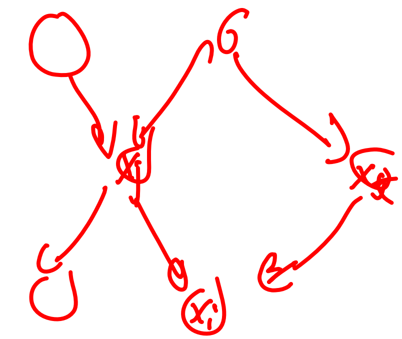
Joint probability distribution:

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

can represent the real world

A general Bayes net

- Set of random variables $\{X_1, \ldots, X_n\}$
- Directed acyclic graph
  - Encodes independence assumptions



$P(X_i \mid X_j, X_k)$

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

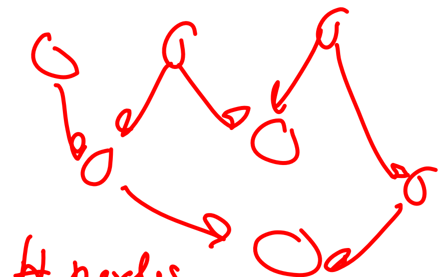
  $P(S \mid A, F)$

- 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 – $P(X_i \mid \mathbf{Pa}_{X_i})$



nodes

for each node, parents

each var can take k values

pars in $P(X_i \mid Pa_{X_i})$ → for assignment of parents, prob. dist. over X_i

parents → $K^{|Pa_{X_i}|} \cdot (K-1)$

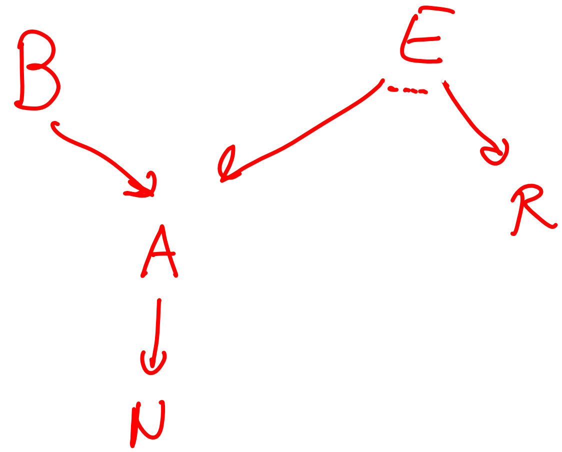
e.g., if nodes have at most d parents

total # params in BN $= O(K^d (K-1)\, n)$

if explicit joint # params $= O(K^n - 1)$

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
  - ($X_i \perp \text{NonDescendants}_{X_i} \mid \mathbf{Pa}_{X_i}$)
- But then we talked about other (in)dependencies
  - e.g., explaining away

$$\begin{tikzcd} B \arrow[dr] & & E \arrow[dl] \\ & A & \end{tikzcd}$$

$B \perp E$

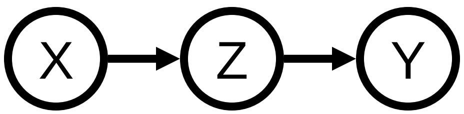
$\neg\, B \perp E \mid A$

- 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 and only its parents

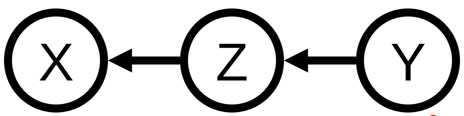
Indirect causal effect:



$\neg X \perp Y$

$X \perp Y \mid Z$

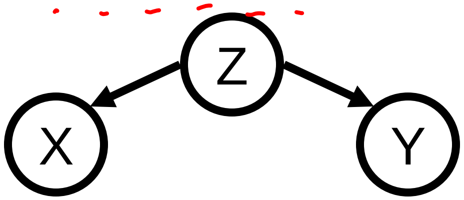
Indirect evidential effect:



$\neg X \perp Y$

$X \perp Y \mid Z$

Common cause:

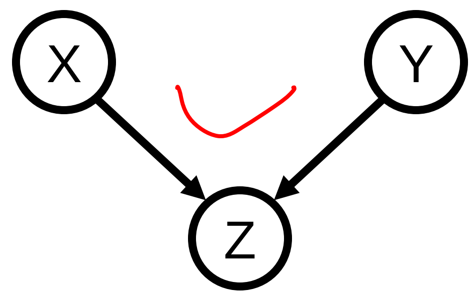


$\neg X \perp Y$

$X \perp Y \mid Z$

v-structure

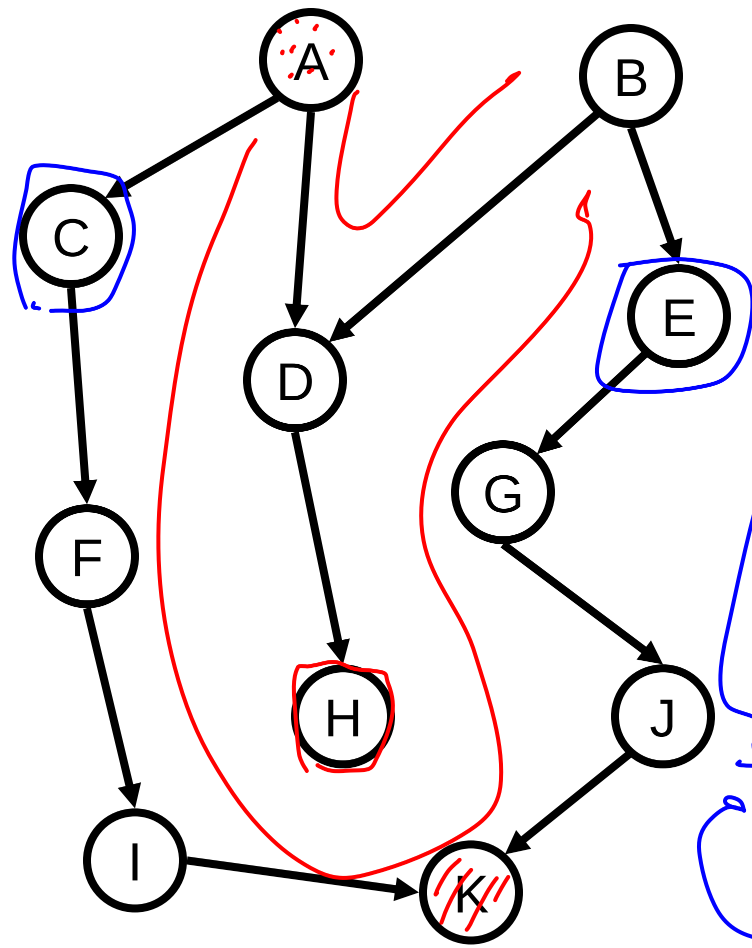
Common effect:



$X \perp Y$

$\neg X \perp Y \mid Z$

Understanding independencies in BNs – Some examples



$A \perp \{B, E, G, J\}$

$\neg A \perp B \mid D$

$\neg A \perp B \mid K$

$\neg A \perp B \mid H$

can prove $C \perp E$

local Markov not enough to prove this...

$\boxed{\text{local Markov: } C \perp E \mid A}$

Also: $C \perp E \mid D$ & $C \perp E \mid \{A, B\}$

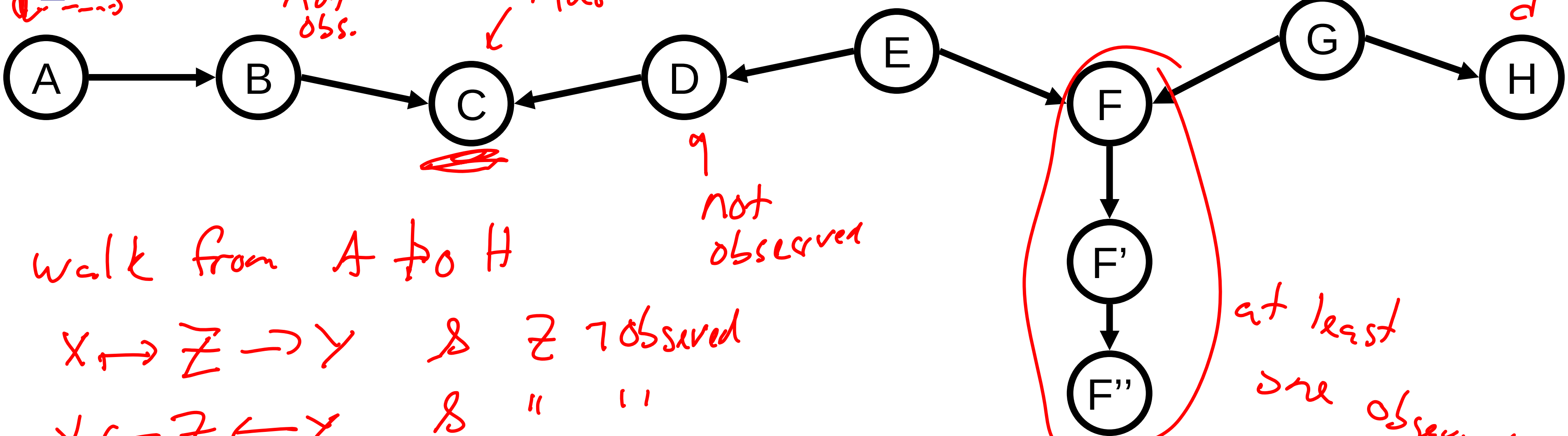
An active trail – Example

not obs.

must be observed

not observed

not observed



not observed

at least one observed

walk from A to H

$X \rightarrow Z \rightarrow Y$ & Z ¬observed

$X \leftarrow Z \leftarrow Y$ & " "

$X \leftarrow Z \rightarrow Y$ & " "

When are A and H independent?

$X \rightarrow Z \leftarrow Y$

Z is observed or at least one descendant of Z observed

Active trails formalized

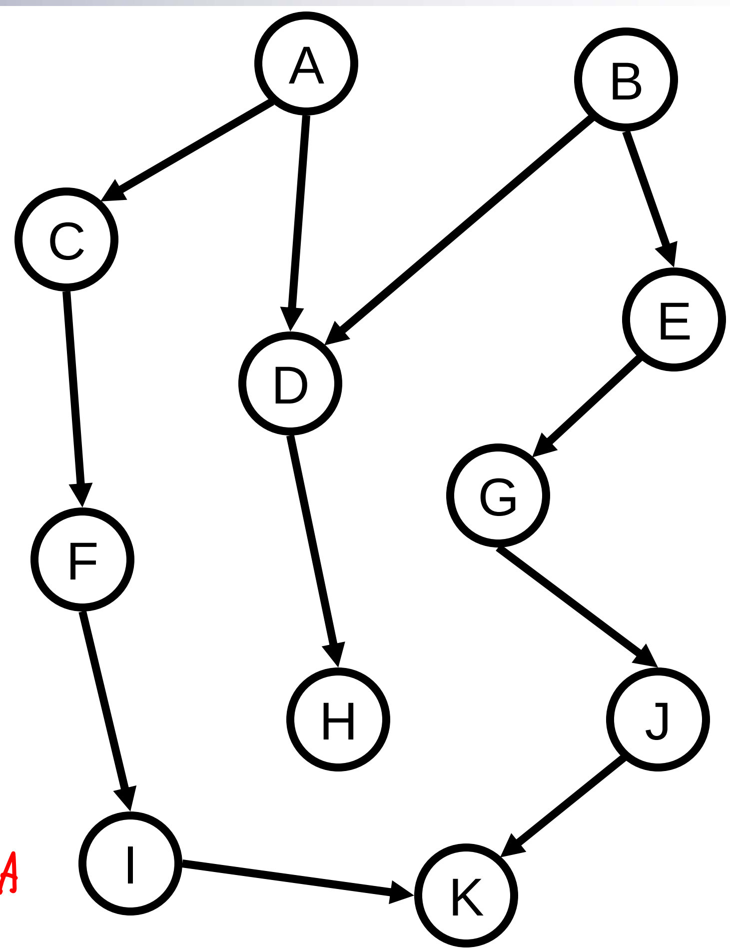
- A path $X_1 - X_2 - \cdots - X_k$ is an **active trail** when variables $\boldsymbol{O} \subseteq \{X_1,\ldots,X_n\}$ are observed if for each consecutive triplet in the trail:
  - $X_{i-1} \rightarrow X_i \rightarrow X_{i+1}$, and X_i is **not observed** ($X_i \notin \boldsymbol{O}$)
  - $X_{i-1} \leftarrow X_i \leftarrow X_{i+1}$, 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** ($X_i \notin \boldsymbol{O}$)

V-structure

  - $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, 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



$\boxed{C \perp E}$

$\neg\, C \perp E \mid H$

$\neg\, F \perp G \mid H, K$

$\neg\, F \perp G \mid H, K\, J$

$F \perp G \mid H\, K\, J\, A$

The BN Representation Theorem

represented exactly

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

indep true in world

Obtain

Joint probability distribution:

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

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

if I write P using BN

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*

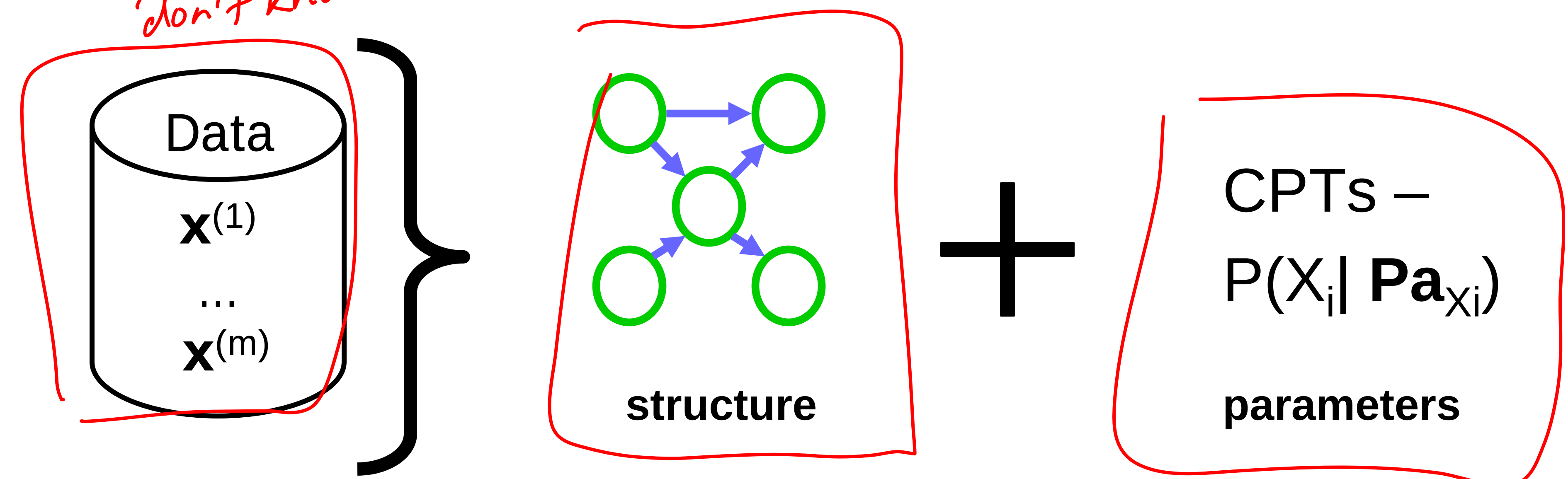
Important because:
Read independencies of *P* from BN structure *G*

Learning Bayes nets

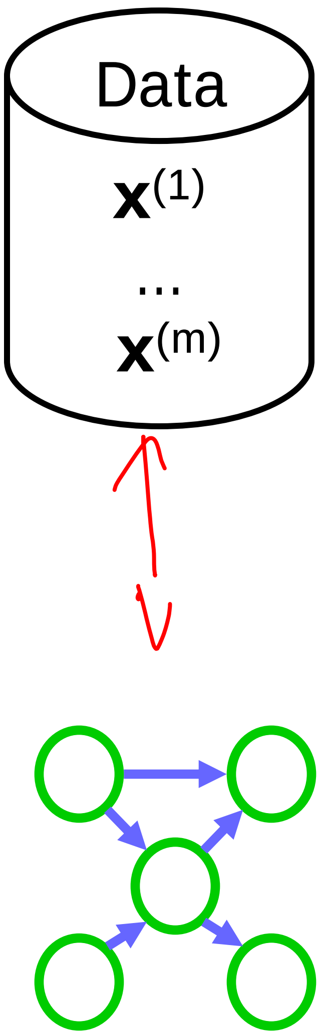
	Known structure	Unknown structure
Fully observable data ⟨A=t, H=f, S=t, F=t⟩	very easy!! :)	learning "good" structure hard ... next week
Missing data	hard... talk about in two weeks	really really hard... we'll talk about it next semester

⟨A=?, H=f, S=t, F=?, N=t⟩

? don't know



Learning the CPTs



For each discrete variable X_i

want to learn $P(X_i \mid Pa_{X_i})$

$$P(\overset{=t}{S} \mid \overset{=t}{F} \overset{=f}{A}) = \frac{\text{Count}(S=t, F=t, A=f)}{\text{Count}(F=t, A=f)}$$

Maximum likelihood estimates

$$\text{MLE:} \quad P(X_i = x_i \mid \overbrace{X_j}^{\text{set of parents}} = x_j) = \frac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$$

What you need to know

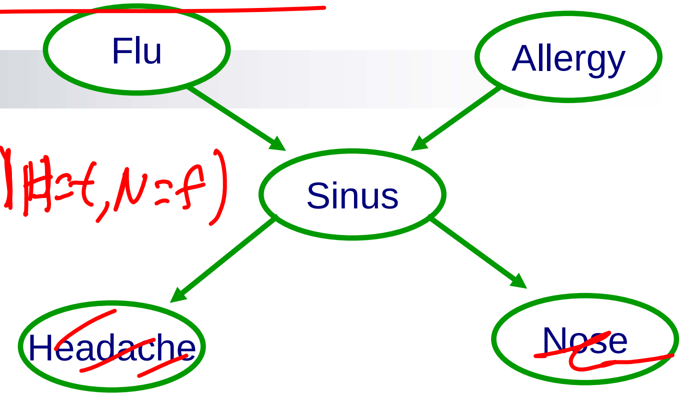
- Bayesian networks
  - A compact **representation** for large probability distributions
  - Not an algorithm
- Semantics of a BN
  - Conditional independence assumptions
- Representation
  - Variables
  - Graph
  - CPTs
- Why BNs are useful
- Learning CPTs from fully observable data & Known Structure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

Announcements

Happy Spring!!

- Tomorrow's recitation on BNs

General probabilistic inference



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

$P(F=t \mid H=t, N=f)$

Defn. cond. probs.

- Using ~~Bayes rule~~:

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

constant doesn't depend on X

$P(F, H=t, N=f)$

F	
t	.3
f	.2

not normalize

normalize

$P(F \mid H=t, N=f)$

t	.6
f	.4

- Normalization:

$$P(X \mid e) \propto \boxed{P(X, e)}$$

normalize

Compute to give answer

Marginalization

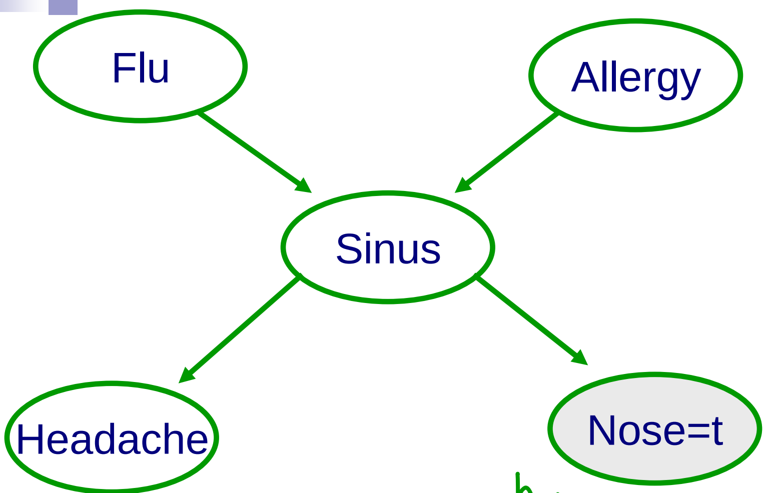
$$P(F, S, N) = P(F) \cdot P(S|F) \cdot P(N|S)$$

$$P(F=t, N=t) = P(F=t, S=t, N=t) + P(F=t, S=f, N=t)$$

$$= P(F=t) \cdot P(S=t|F=t) \cdot P(N=t|S=t) + P(F=t) \cdot P(S=f|F=t)\, P(N=t|S=f$$

marginalize out S

Probabilistic inference example



$P(F, N=t)$ ← want

know

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

per value of F
7 sums
$8 \times 4 = 32$ multiplies

grand total
14 sums
64 multiply

how many terms adding? 2^3

$$P(F, N=t) = \sum_a \sum_s \sum_h P(F, A=a, S=s, H=h, N=t)$$

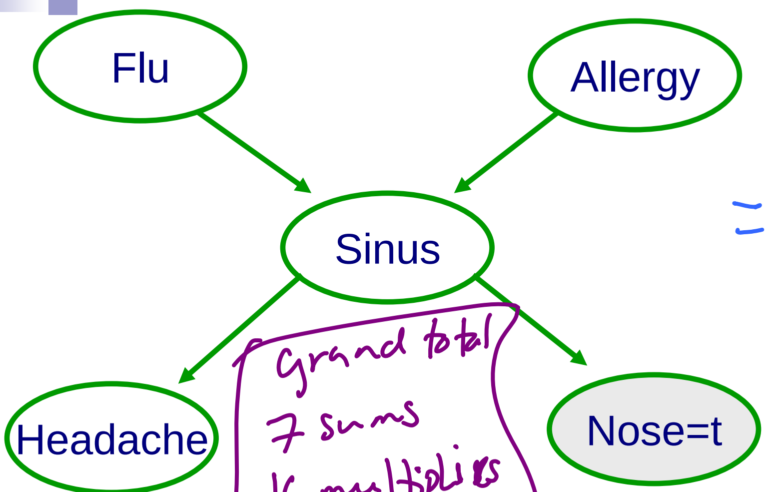
$$= \sum_a \sum_s \sum_h P(F) \cdot P(a) \cdot P(s|F,a) \cdot P(h|s) \cdot P(N=t|s)$$

Inference seems exponential in number of variables!

Actually, inference in graphical models is in general **NP-hard** ☹

Fast probabilistic inference example – Variable elimination

eliminate (marginalize) vars one at a time



grand total
7 sums
16 multiplies

$$P(F, N=t) = \sum_a \sum_s \sum_h P(F) . P(a) . P(s|F,a) . P(h|s) . P(N=t|s)$$

$$= \sum_a \sum_s P(F) . P(a) . P(s|F,a) . P(N=t|s) . \sum_h P(h|s)$$

$\sum_h P(h|s) = 1$ special case

1 sum
0 multiplies

$$= \sum_a \sum_s P(F) . P(a)\, P(s|F,a)\, P(N=t|s) . 1$$

$$= \sum_a P(F)\, P(a) \underbrace{\sum_s P(s|F,a) . P(N=t|s)}_{g_1(F,a)}$$

For each assignment of F & A: 1 sum, 2 multiplies; total 4 sums, 8 multiplies

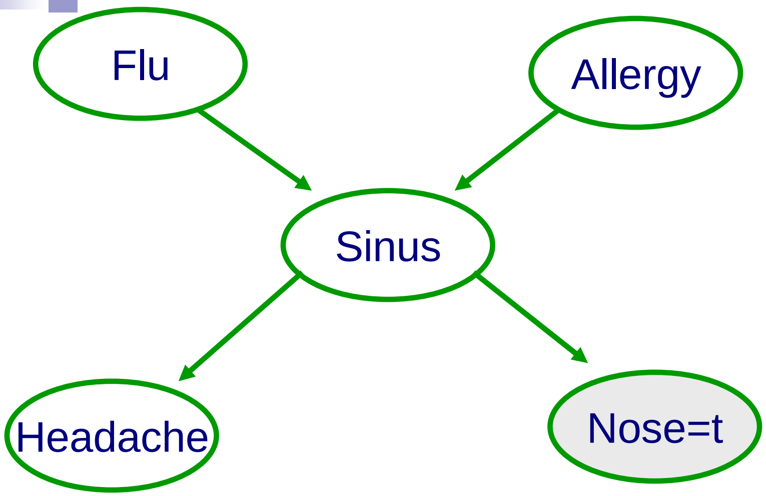
$$= \underbrace{\sum_a P(F) . P(a) . g_1(F,a)} = P(F, N=t)$$

for each assignment of F: 1 sum, 4 multiplies; total 2 sums, 8 multiplies

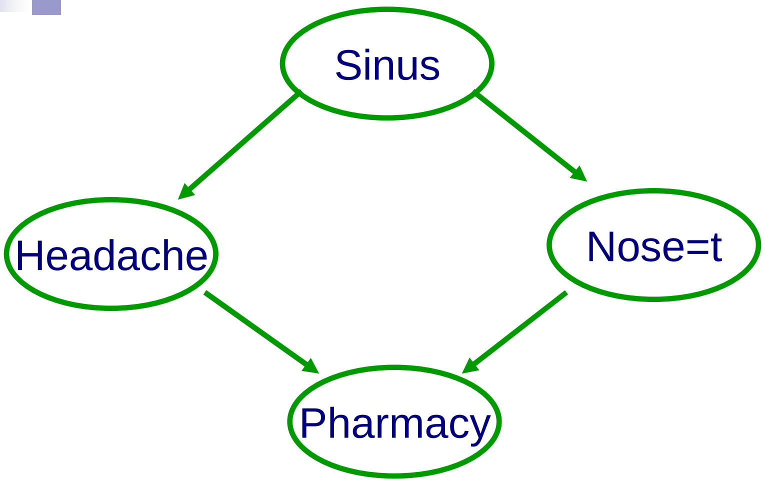
(Potential for) Exponential reduction in computation!

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

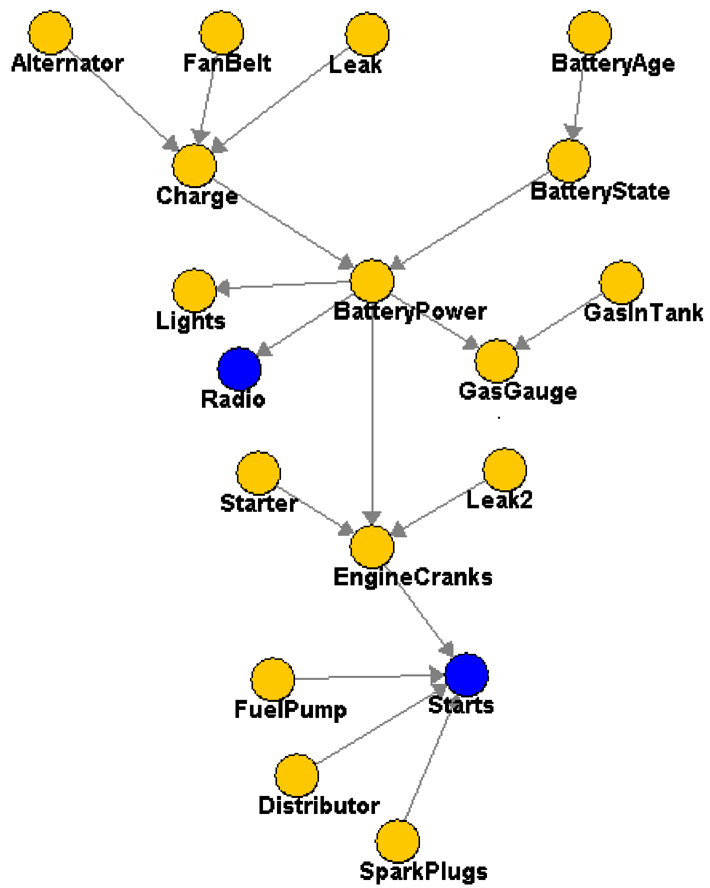
IMPORTANT!!!

- 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 $P(X|e)$

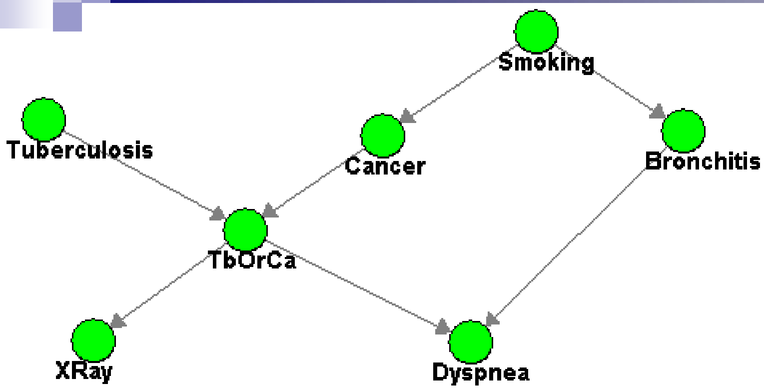
Complexity of variable elimination – (Poly)-tree graphs



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

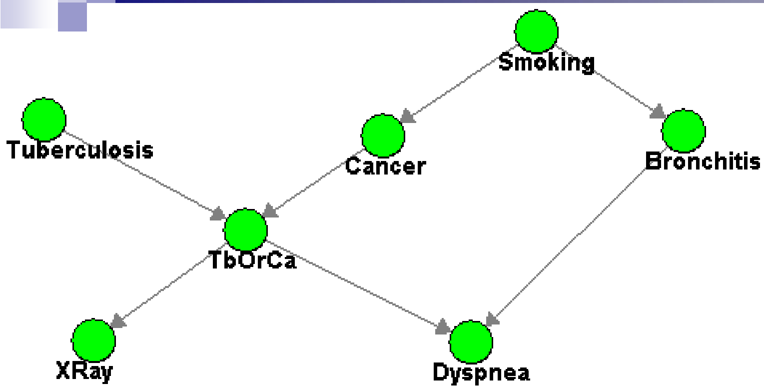
Linear in number of variables!!! (versus exponential)

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

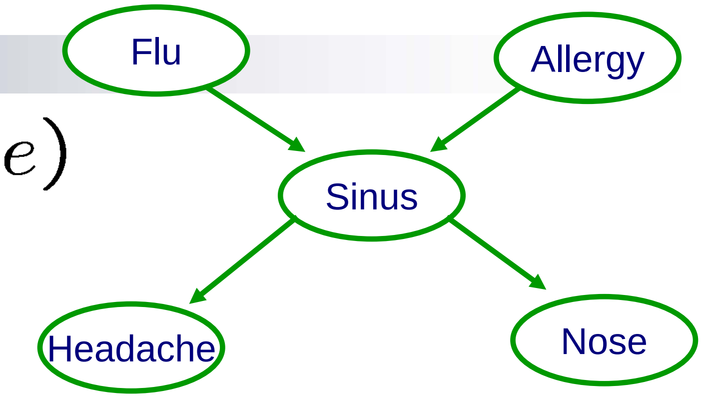
Example: Large tree-width with small number of parents

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

- Using Bayes rule:

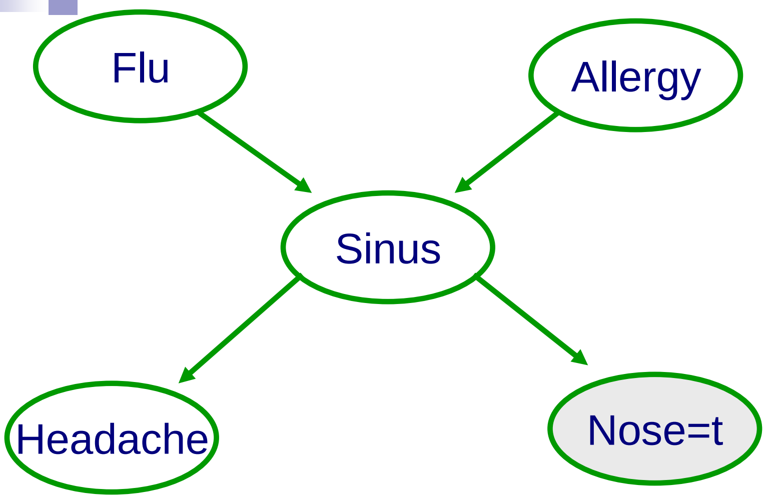
$$\underset{x_1,\ldots,x_n}{\text{argmax}} P(x_1, \ldots, x_n \mid e) = \underset{x_1,\ldots,x_n}{\text{argmax}} \frac{P(x_1, \ldots, x_n, e)}{P(e)}$$

- Normalization irrelevant:

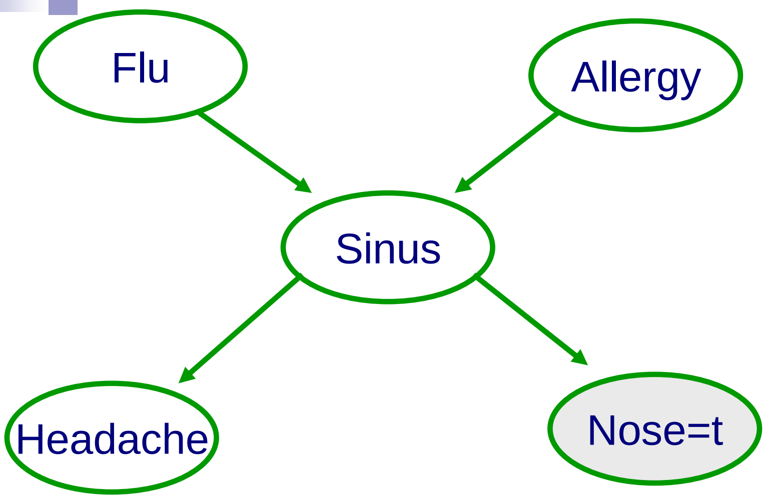
$$\underset{x_1,\ldots,x_n}{\text{argmax}} P(x_1, \ldots, x_n \mid e) = \underset{x_1,\ldots,x_n}{\text{argmax}} P(x_1, \ldots, x_n, e)$$

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"