Bayesian Networks – Inference

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

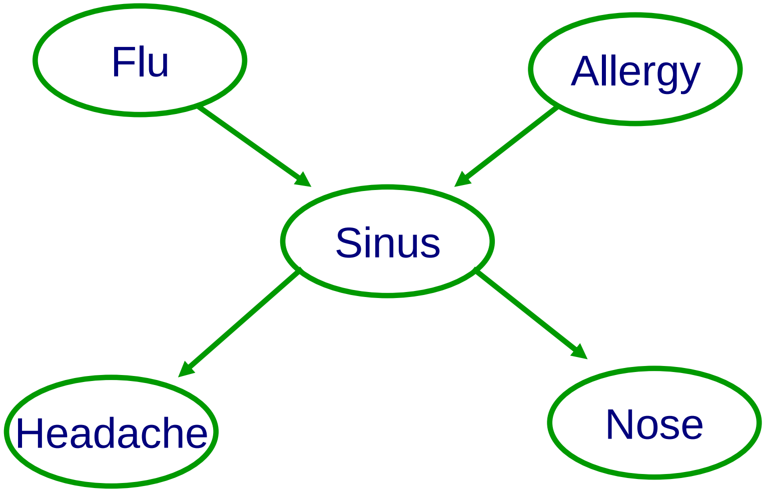
March 21st, 2005

Class project

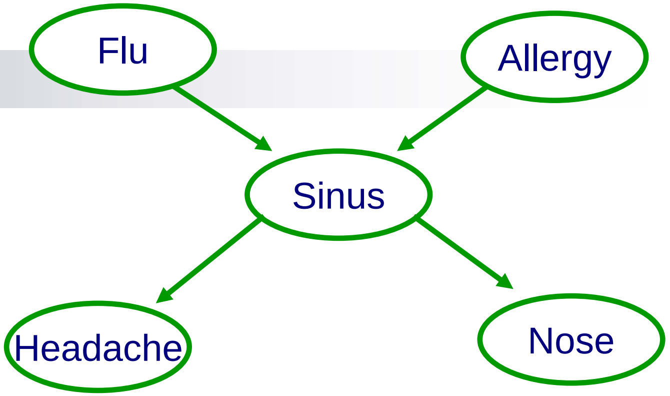
- Homework 4 out today – Due April 4th (2 weeks)
- Includes 10/100 points for your project proposal – this part is due March 28th (1 week)
- Project
  - Up 2 students per team
  - Objective: define a learning problem, experiment with real data, write a paper, and present a poster, and learn something new and have fun!
  - Ideas in class website
  - Project description **due 3/28**
  - Graded milestone **due 4/13** (20% of project grade)
  - Poster **due 4/30** (20% project grade)
  - Paper **due 5/03** (60% project grade)

Last lecture

- Bayesian Networks
  - Compact representation for probability distributions
  - Exponential reduction in number of parameters
  - Key insight: Conditional independence assumptions!
- Showed very fast inference with applet
  - Why???



General probabilistic inference



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

$P(\underbrace{F=t}_{X} \mid \underbrace{H=f, N=t}_{e})$

- Using Bayes rule:

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

$$\frac{P(F=t, H=f, N=t)}{P(H=f, N=t)}$$

constant
doesn't depend on X.

- Normalization:

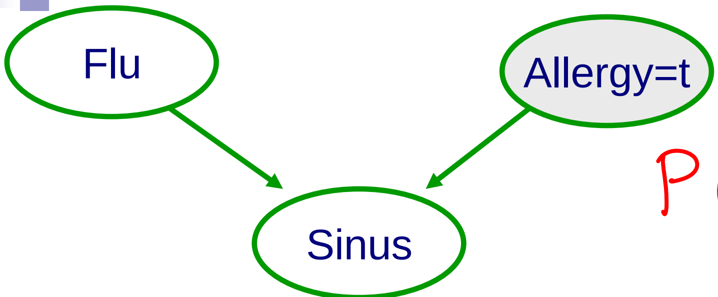
$$P(X \mid e) \propto P(X, e)$$

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

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

Normalize to get $P(X|e)$

Marginalization



$P(F \mid A=t)$ ← I want

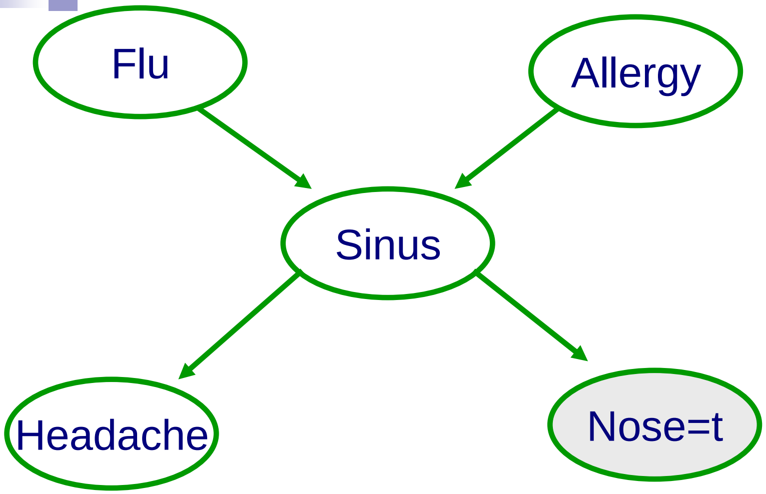
$P(F,A,S) = P(A) \cdot P(F) \cdot P(S \mid F,A)$ ← know

$$P(F=t, A=t) = \sum_{S} P(F=t, A=t, s)$$

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

Notation: $\sum_{X}$ means sum over possible assignments to X

Probabilistic inference example



I want for each assignment of F

Want: $P(F \mid N=t)$ or $P(F, N=t)$

$$P(F, N=t) = \sum_{A,S,H} P(F, A, S, H, N=t)$$

$$\left.\begin{aligned} = \; & P(F, A=t, S=t, H=t, N=t) \\ & + \\ & P(F, A=f, S=t, H=t, N=t) \\ & + \\ & \vdots \\ & + \\ & P(F, A=f, S=f, H=f, N=t) \end{aligned}\right\} 8$$

problem with n binary variables:

2^n summations

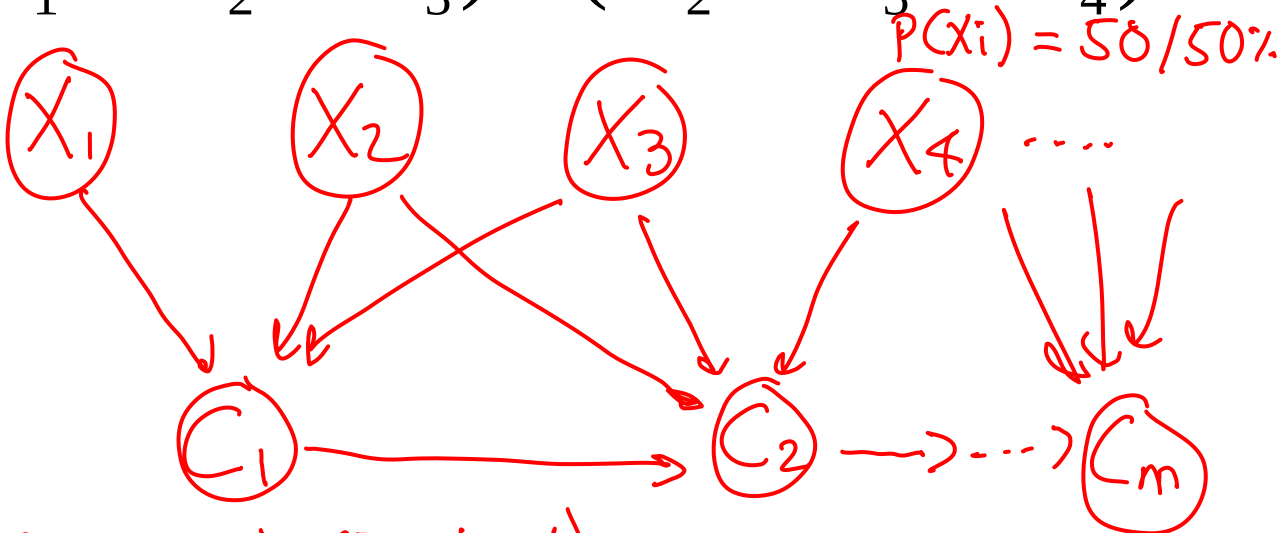
$n\,2^n$ multiplications

Inference seems exponential in number of variables!

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

Reduction – 3-SAT

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



$P(C_1 = t \mid X) = (\bar{x}_1 \vee x_2 \vee x_3)$

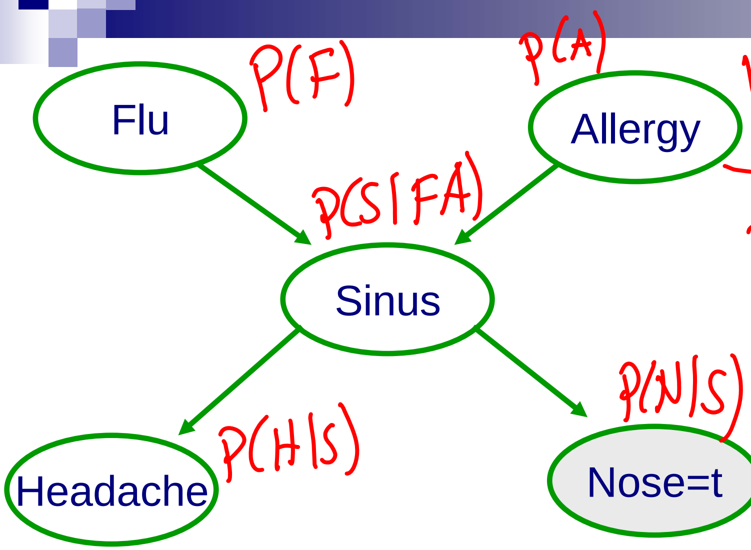
$P(C_2 = t \mid X, C_1) = (\bar{x}_2 \vee x_3 \vee x_4) \wedge C_1$

iff $P(C_m) > 0$

formula satisfied

Inference unlikely to be efficient in general, but…

Fast probabilistic inference example – Variable elimination



Want $P(F \mid N=t)$, or $P(F, N=t)$

$$P(F, N=t) = \sum_{A,S,H} P(F,A,S,H,N=t)$$

$$= \underbrace{\sum_{A,S,H}}_{\sum_A \sum_S,\ \sum_H} P(F).P(A).P(S|FA).P(H|S).P(N=t|S)$$

$$= \sum_{AS} P(F).P(A)\,P(S|FA)\,P(N=t|S) \sum_H P(H|S)$$

$$= P(F) \sum_A P(A) \sum_S P(S|FA).P(N=t|S).\underbrace{\sum_H P(H|S)}_{1}$$

(Potential for) Exponential reduction in computation!

f_0 — 0 mult., 1 sum
f_1 — 4.(1 sum + 4 mult.)
f_2 — 2 (1 sum, 2 mult.)

Variable Elimination

$$P(F) \sum_A P(A) \sum_S P(S|FA)\, P(N{=}t|S) \underbrace{\sum_H P(H|S)}_{1 \leftarrow f_0}$$

$$= P(F) \sum_A P(A) \underbrace{\sum_S P(S|FA)\, P(N{=}t|S)}_{f_1(F,A) \text{ — table of size } 2\times 2}$$

$$= P(F) \underbrace{\sum_A P(A)\, f_1(F,A)}_{f_2(F) \text{ — table of size } 2\times 1}$$

$$= P(F)\, f_2(F)$$

$$= P(F, N{=}t) \text{ — table of size } 2\times 1$$

Understanding variable elimination – Exploiting distributivity

Sum out S:

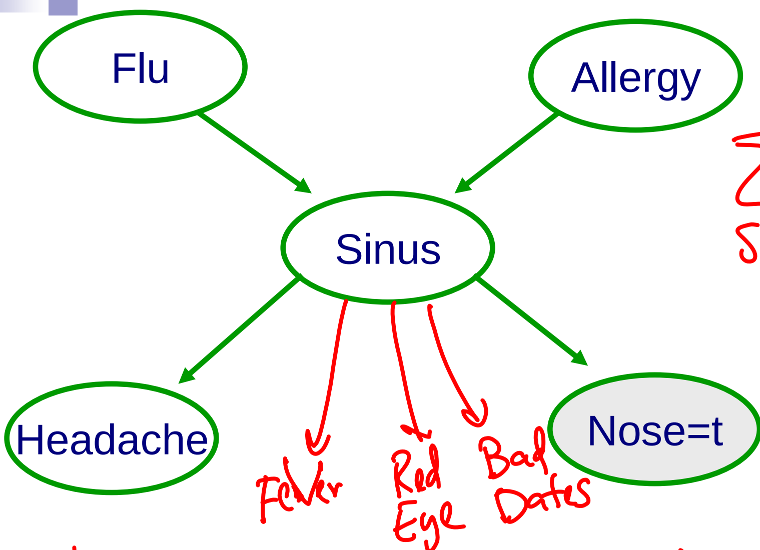
$$P(F=t) \cdot P(S=t \mid F=t) \cdot P(N=t \mid S=t)$$

$$+$$

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

$$= P(F=t) \cdot \left[P(S=t \mid F=t)\, P(N=t \mid S=t) + P(S=f \mid F=t) \cdot P(N=t \mid S=f) \right]$$

Understanding variable elimination – Order can make a HUGE difference



After sum S first

$f_1(F,A,H,V,R,B)$

table $2\times2\times2\times2\times2\times2$

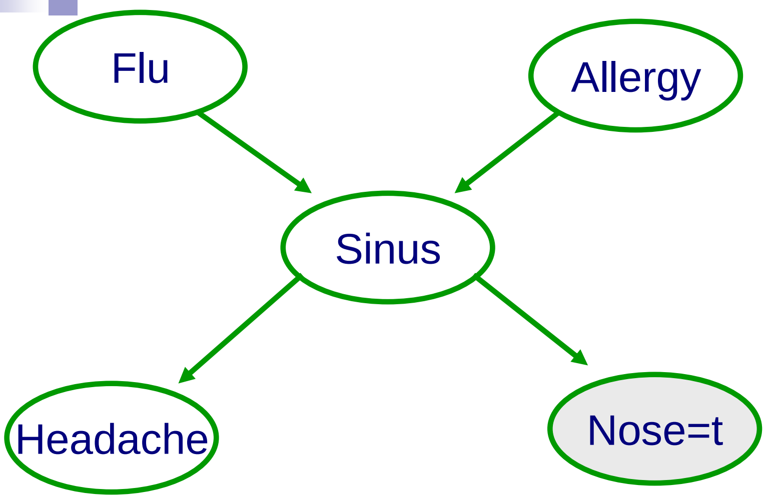
What happens if I sum S first?

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

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

$$= P(F)\,P(A)\underbrace{\sum_S P(S|FA)\,P(H|S)\,P(N=t|S)}_{f_1(F,A,H)\ -\ \text{table } 2\times2\times2}$$

Understanding variable elimination – Intermediate results



I started with

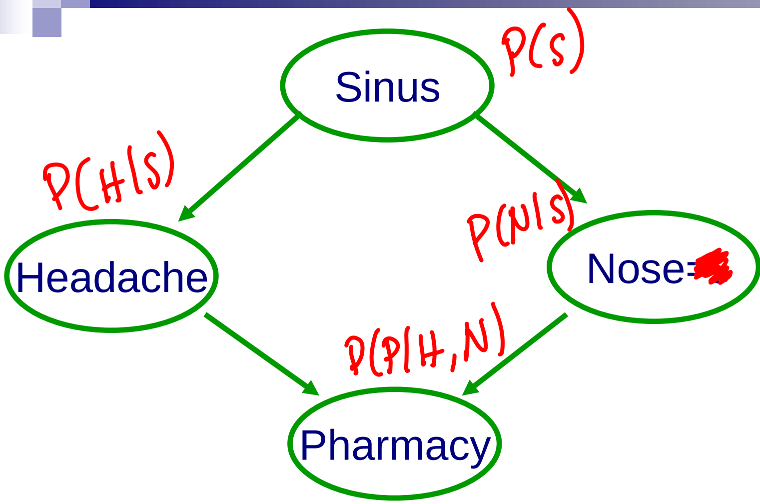
$P(F, A, S, H, N{=}t)$

Sum out H

$$P(F, A, S, N{=}t) = \sum_H P(F, A, S, H, N{=}t)$$

Intermediate results are probability distributions

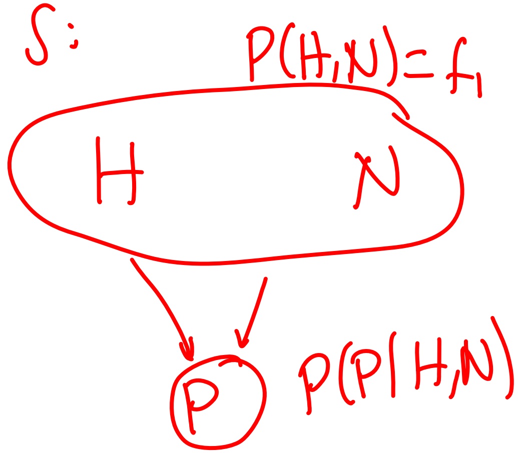
Understanding variable elimination – Another example



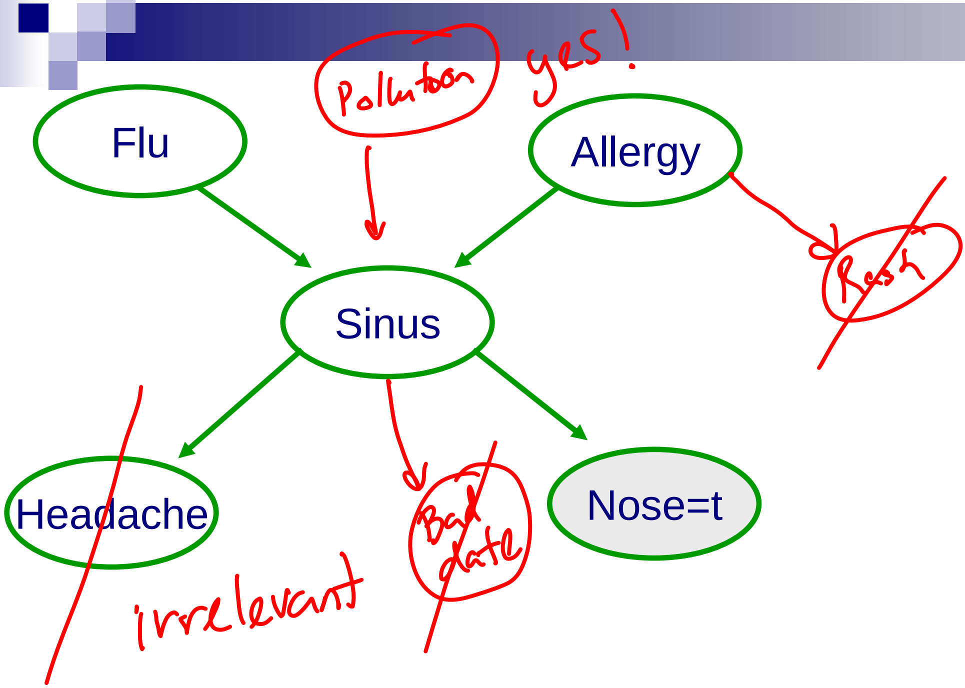
$$\sum_S P(S)\, P(H|S)\, P(N|S)\, P(P|H,N)$$

$$= P(P|H,N) \underbrace{\sum_S P(S)\, P(H|S)\, P(N|S)}_{f_1(N,H) \,=\, P(H,N)}$$

after eliminate S:



Pruning irrelevant variables



Prune all non-ancestors of query variables

Variable elimination algorithm

$\underbrace{x}$ $\underbrace{e}$

P(F | N=t)

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

IMPORTANT!!!

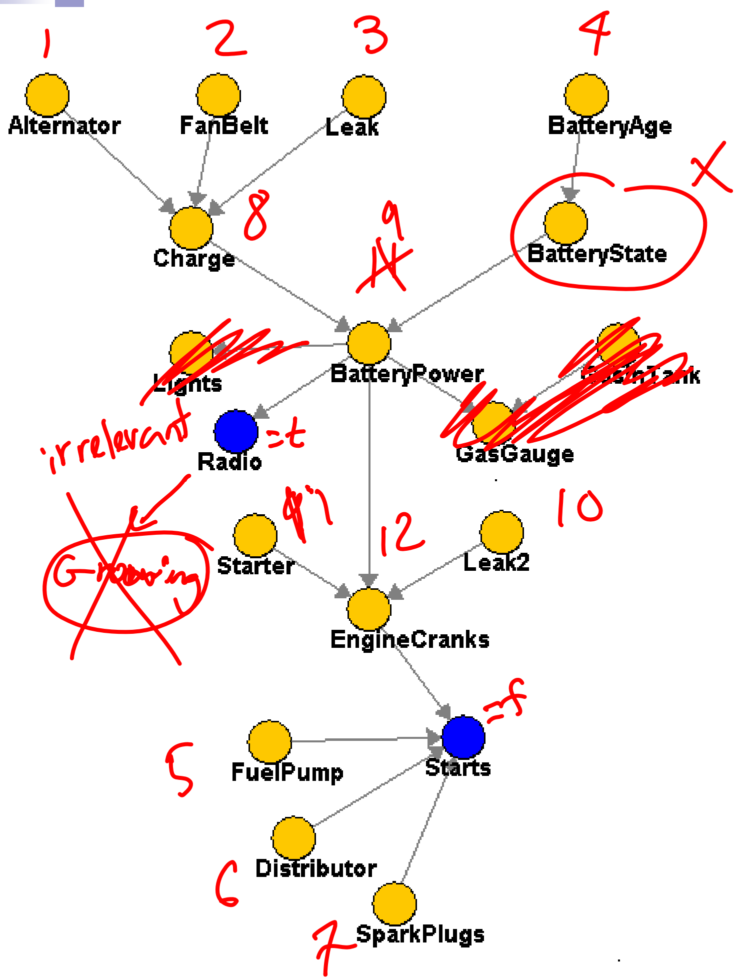
- Choose an ordering on variables, e.g., X_1, …, X_n
- For i = 1 to n, If $X_i \notin \{X,e\}$
  - Collect factors $f_1,\dots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

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

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

Complexity of variable elimination – (Poly)-tree graphs

P(Battery State | R=t, S=f)

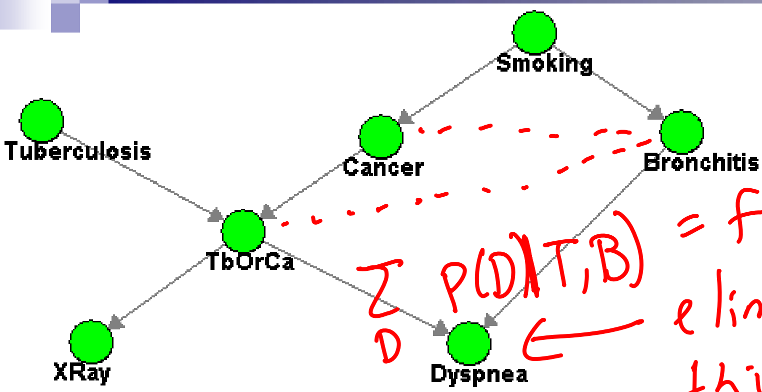


Variable elimination order:
Start from “leaves” up –
find topological order, eliminate variables in reverse order

topological order
↳ reverse it

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

Complexity of variable elimination – Graphs with loops



$\sum_D P(D|T,B) = f_1(T,B)$

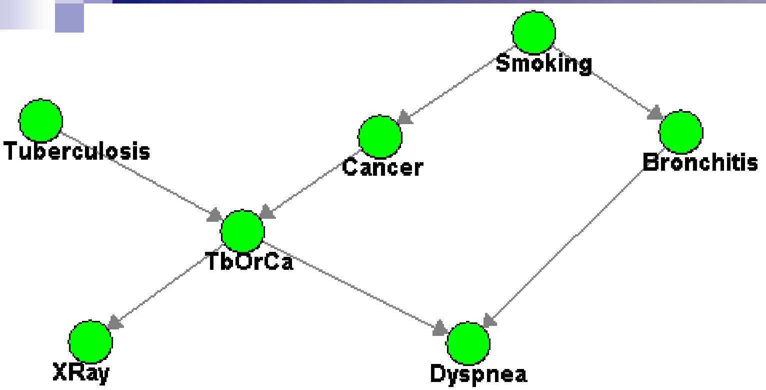
← eliminate this one

If I generate $f_i(A,B,C,D,E)$

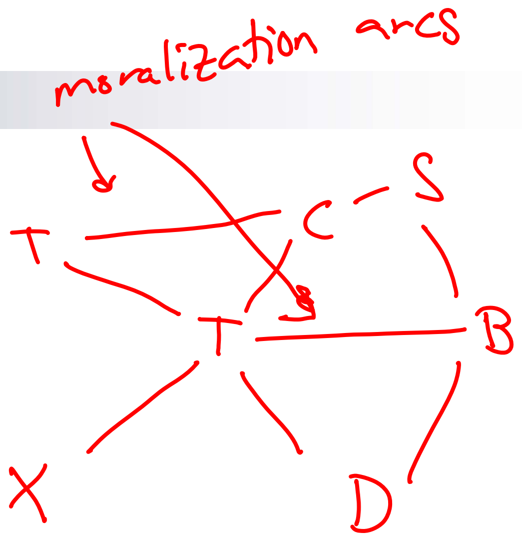
← table $2 \times 2 \times 2 \times 2 \times 2$

Exponential in number of variables in largest factor generated

Complexity of variable elimination – Tree-width

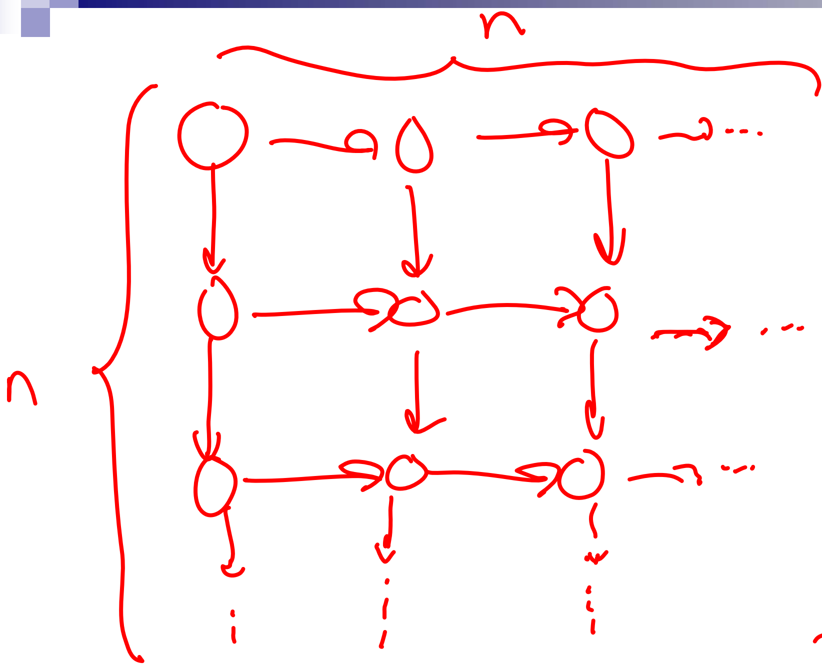


Moralize graph:
Connect parents
into a clique and
remove edge directions



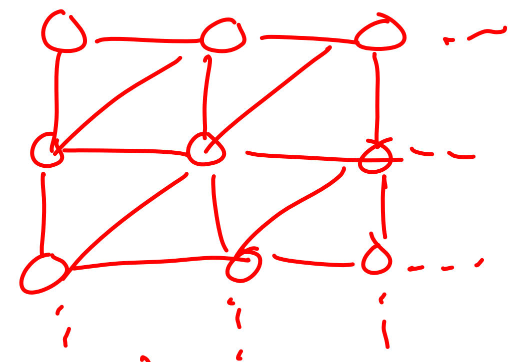
Complexity of VE elimination:
("Only") exponential in tree-width
Tree-width is maximum node cut +1

Example: Large tree-width with small number of parents



How many parameters?
if binary 2^2 per node

moralize
$\Rightarrow$



tree-width = $n+1$ $(\sqrt{2})$

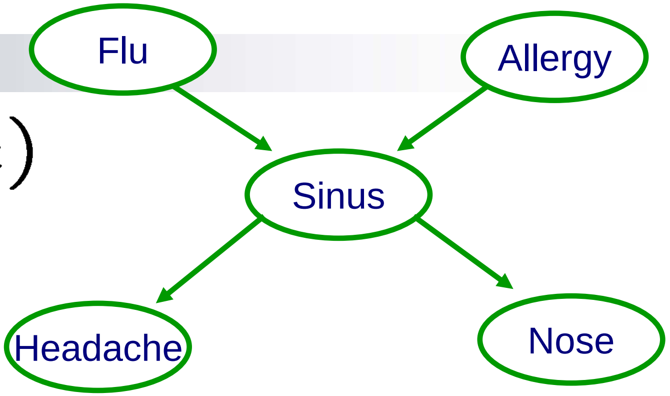
usually

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

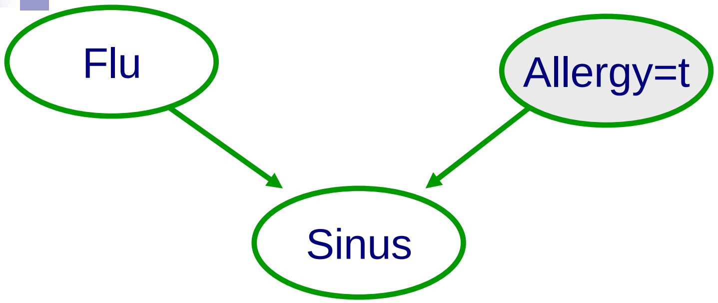
- Using Bayes rule:

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

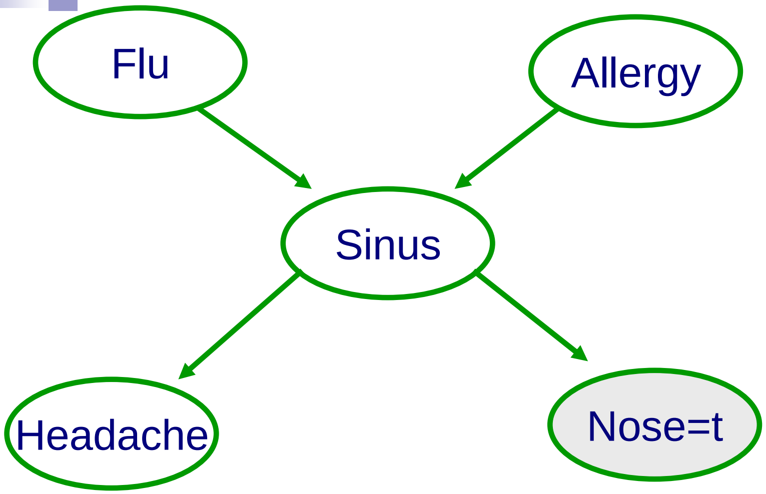
- Normalization irrelevant:

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

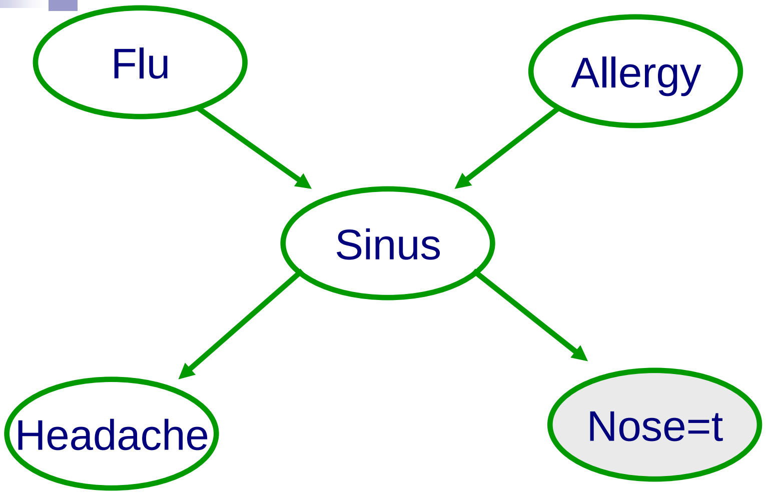
Max-marginalization



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

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

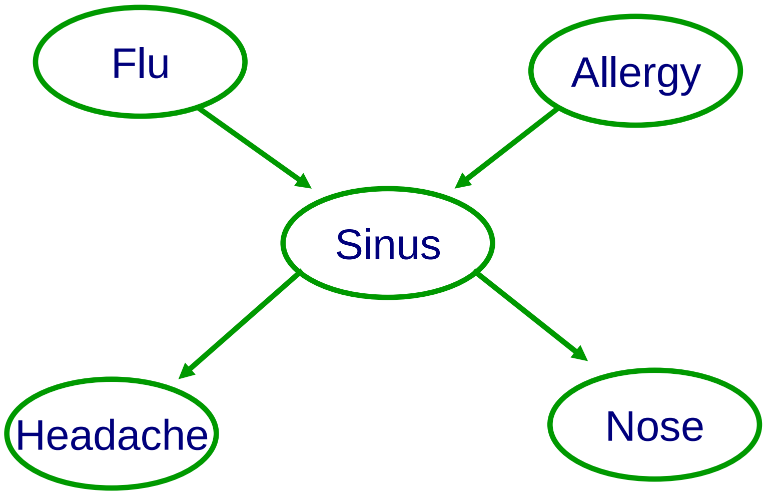
  - Variable X_i has been eliminated!

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:

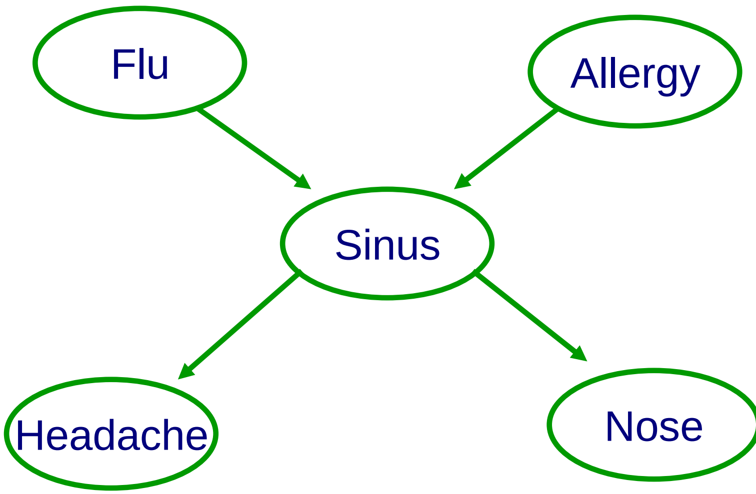
$$x_i^* \in \underset{x_i}{\operatorname{argmax}} \prod_{j=1}^{k} f_j$$

Stochastic simulation – Obtaining a sample from the joint distribution

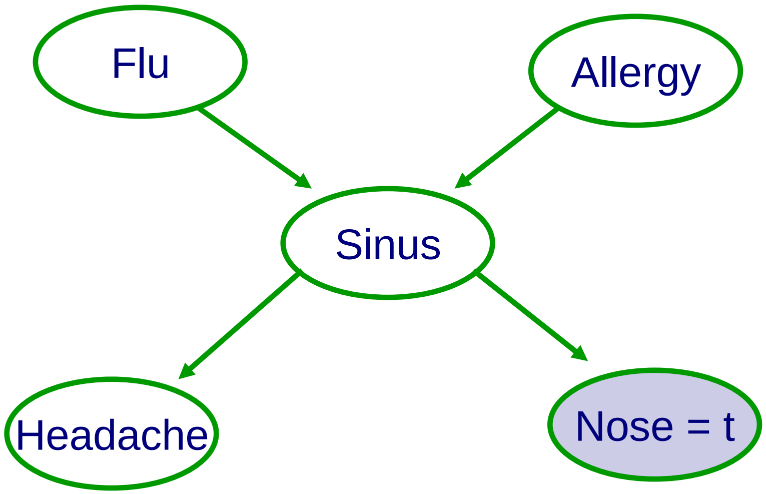


Using stochastic simulation (sampling) to compute P(X)

- Given a BN, a query P(X), and number of samples *m*
- Choose a **topological** ordering on variables, e.g., $X_1, \ldots, X_n$
- For j = 1 to m
  - $\{x_1^j, \ldots, x_n^j\}$ will be j^{th} sample
  - For i = 1 to n
    - Sample x_i^j from the distribution $P(X_i|\mathbf{Pa}_{X_i})$, where parents are instantiated to $\{x_1^j, \ldots, x_{i-1}^j\}$
  - Add $\{x_1^j, \ldots, x_n^j\}$ to "dataset"
- Use counts to compute P(X)

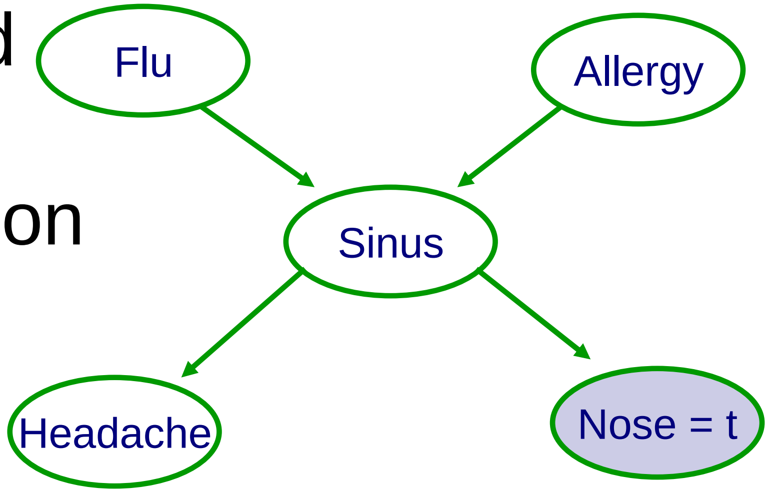


Example of using rejection sampling to compute P(X|e)

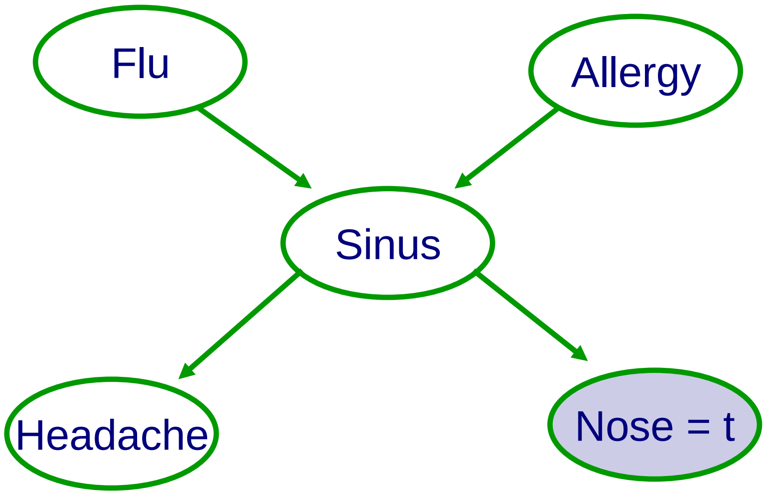


Using rejection sampling to compute P(X|e)

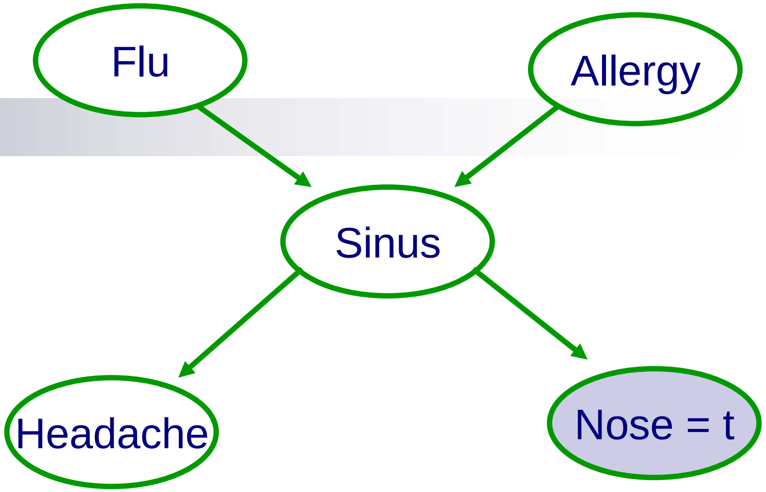
- Given a BN, a query P(X|e), and number of samples *m*
- Choose a **topological** ordering on variables, e.g., $X_1, \ldots, X_n$
- j = 0
- While j < m
  - $\{x_1^j, \ldots, x_n^j\}$ will be j^{th} sample
  - For i = 1 to n
    - Sample x_i^j from the distribution $P(X_i|\mathbf{Pa}_{X_i})$, where parents are instantiated to $\{x_1^j, \ldots, x_{i-1}^j\}$
  - If $\{x_1^j, \ldots, x_n^j\}$ consistent with evidence, add it to "dataset" and j = j +1
- Use counts to compute P(X|e)



Example of using importance sampling to compute P(X|e)



Using importance sampling to compute P(X|e)



- For j = 1 to m
  - $\{x_1^j, \ldots, x_n^j\}$ will be j^{th} sample
  - Initialize weight of sample $w^j = 1$
  - For i = 1 to n
    - If $X_i \notin \{e\}$
      - Sample x_i^j from the distribution $P(X_i|\mathbf{Pa}_{X_i})$, where parents are instantiated to $\{x_1^j, \ldots, x_{i-1}^j\}$
    - else
      - Set x_i^j to assignment in evidence e
      - Multiply weight w^j by $P(x_i^j|\mathbf{Pa}_{X_i})$, where parents are instantiated to $\{x_1^j, \ldots, x_{i-1}^j\}$
  - Add $\{x_1^j, \ldots, x_n^j\}$ to "dataset" with weight w^j
- Use weighted counts to compute P(X|e)

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
  - Only difference between probabilistic inference and MLE is "sum" versus "max"
- Sampling – Example of approximate inference
  - Simulate from model
  - Likelihood weighting for inference
  - Can be very slow

Acknowledgements

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