Bayesian Networks – Inference

Machine Learning – 10701/15781

Carlos Guestrin
Carnegie Mellon University

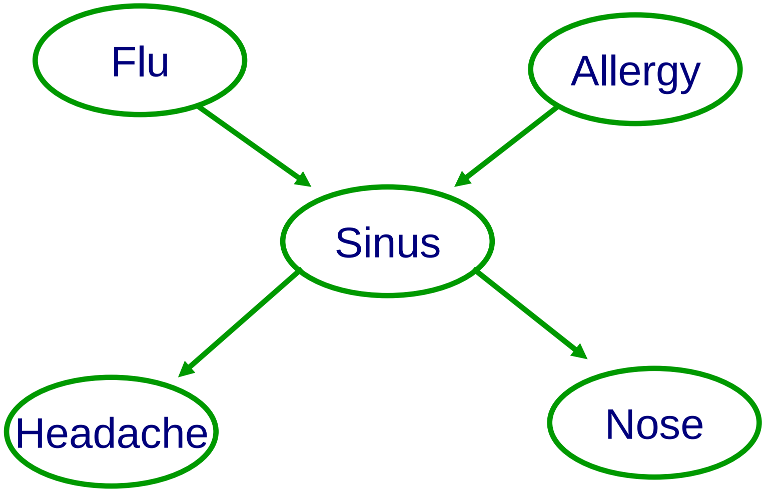
March 21st, 2005

Class project

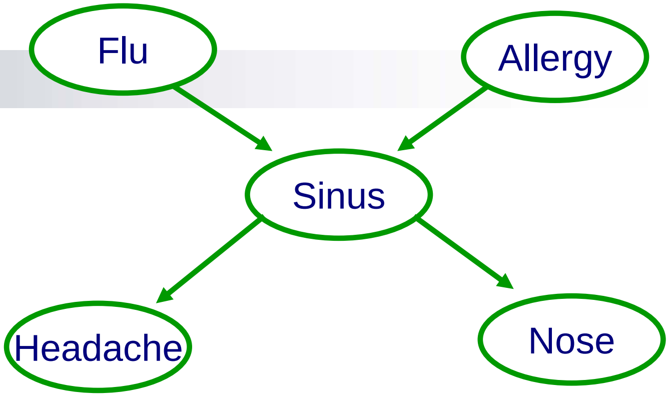
- Homework 4 out today – Due April 4th (2 weeks)
- Includes 10/100 points for your project proposal – this part is due March 28th (1 week)
- Project
  - Up 2 students per team
  - Objective: define a learning problem, experiment with real data, write a paper, and present a poster, and learn something 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: $P(X \mid e)$

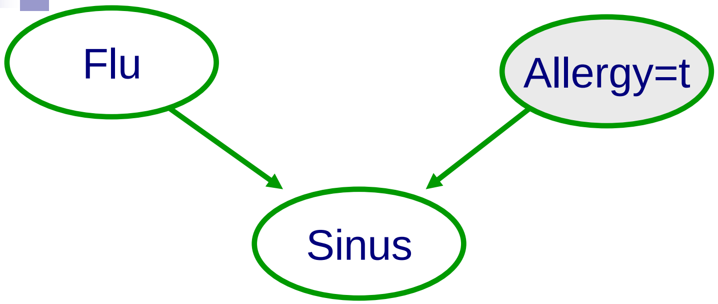
- Using Bayes rule:

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

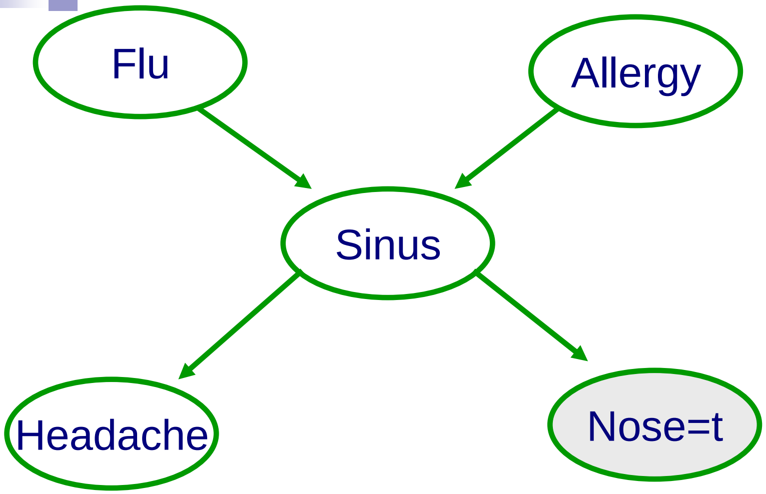
- Normalization:

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

Marginalization



Probabilistic inference example



Inference seems exponential in number of variables!

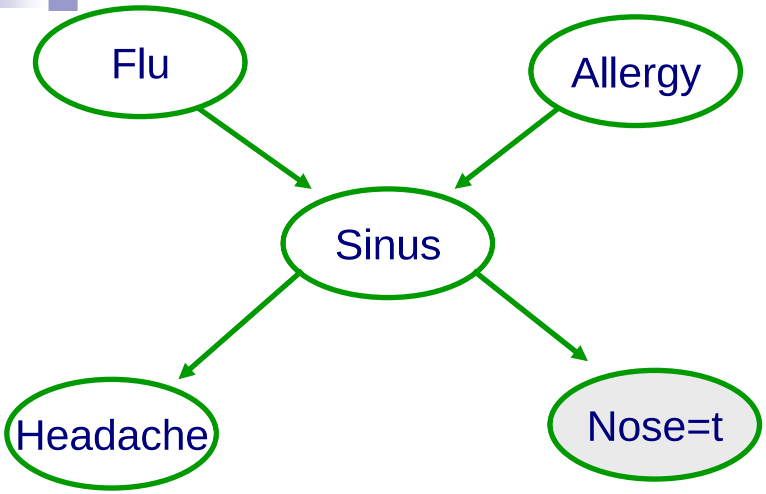
Inference is NP-hard (Actually #P-complete)

- 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…

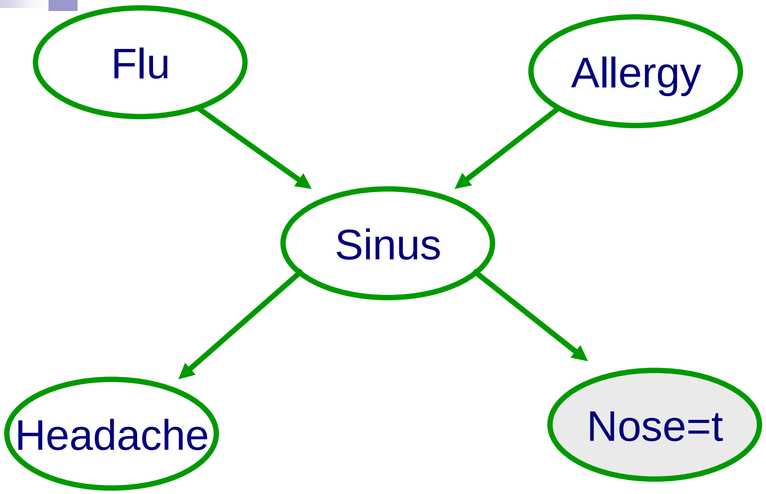
Fast probabilistic inference example – Variable elimination



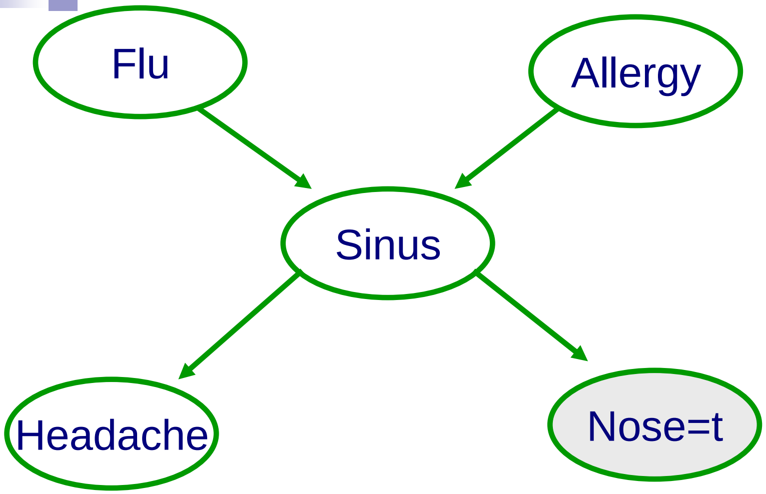
(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity

Understanding variable elimination – Order can make a HUGE difference

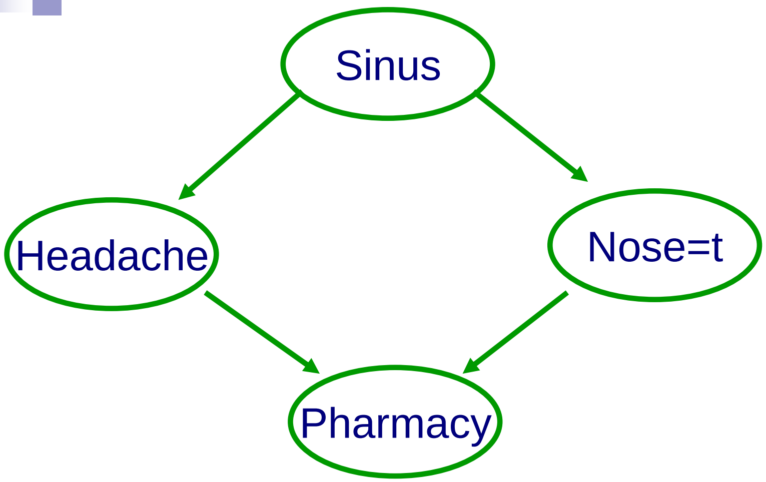


Understanding variable elimination – Intermediate results

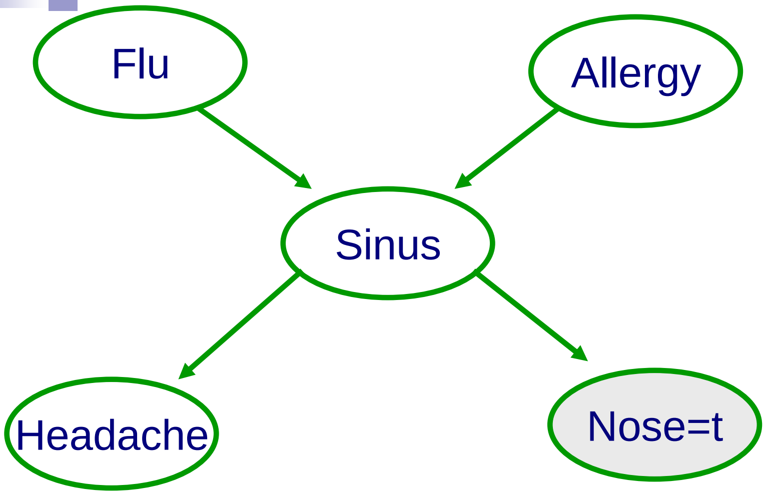


Intermediate results are probability distributions

Understanding variable elimination – Another example



Pruning irrelevant variables



Prune all non-ancestors of query variables

Variable elimination algorithm

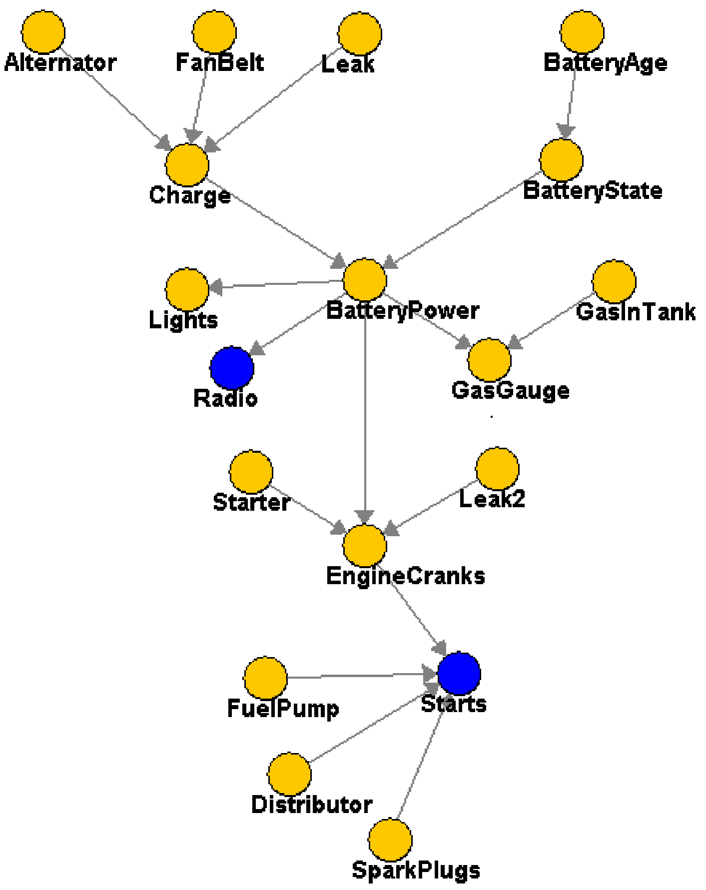
- Given a BN and a query $P(X|e) \propto P(X,e)$
- Instantiate evidence e
- Prune non-ancestors of {X,e}
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{X,e\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating 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 $P(X|e)$

IMPORTANT!!!

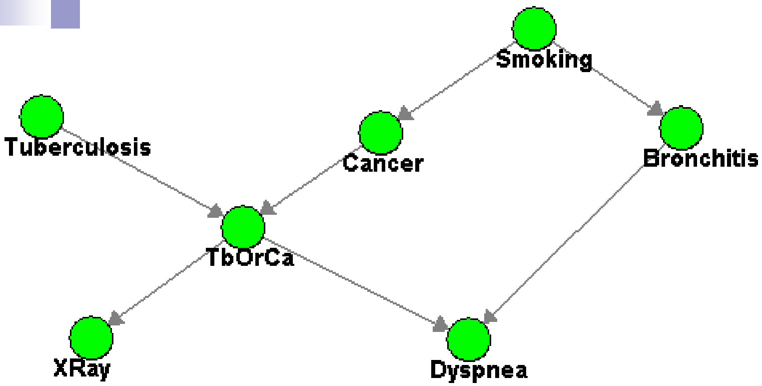
Complexity of variable elimination – (Poly)-tree graphs



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

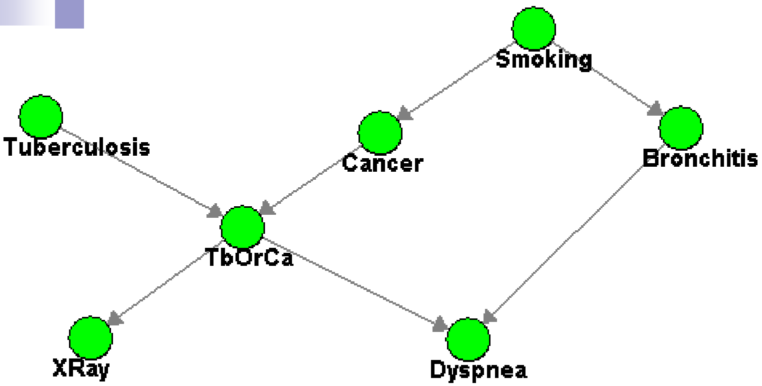
Linear in number of variables!!! (versus exponential)

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

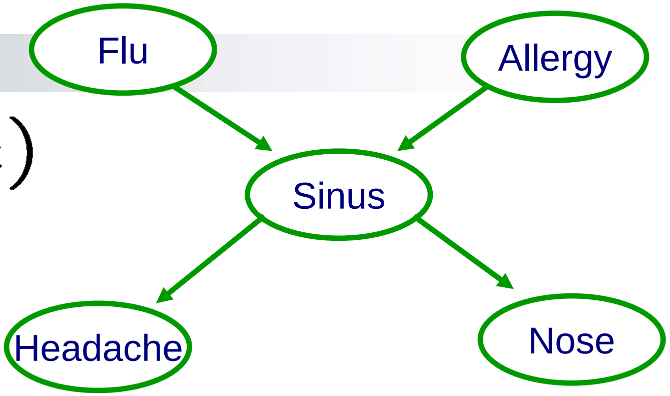
Example: Large tree-width with small number of parents

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

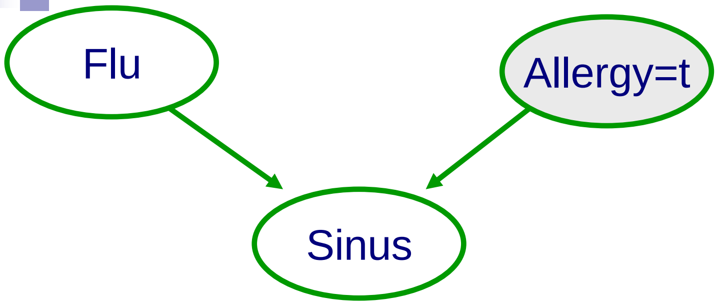
- Using Bayes rule:

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

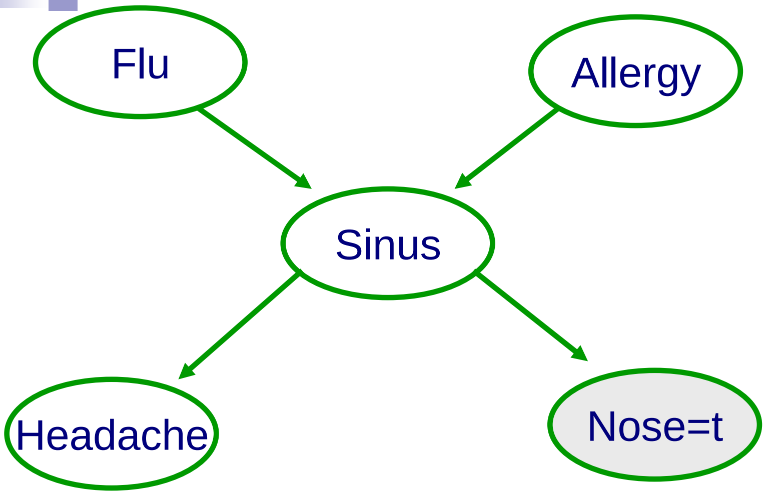
- Normalization irrelevant:

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

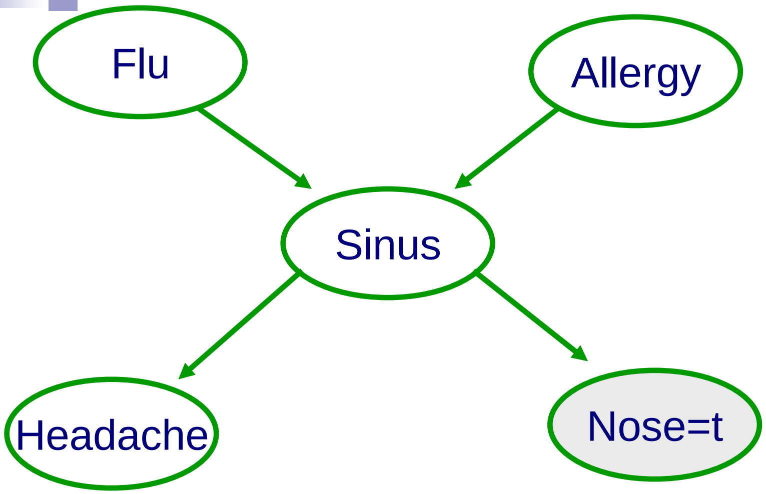
Max-marginalization



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

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

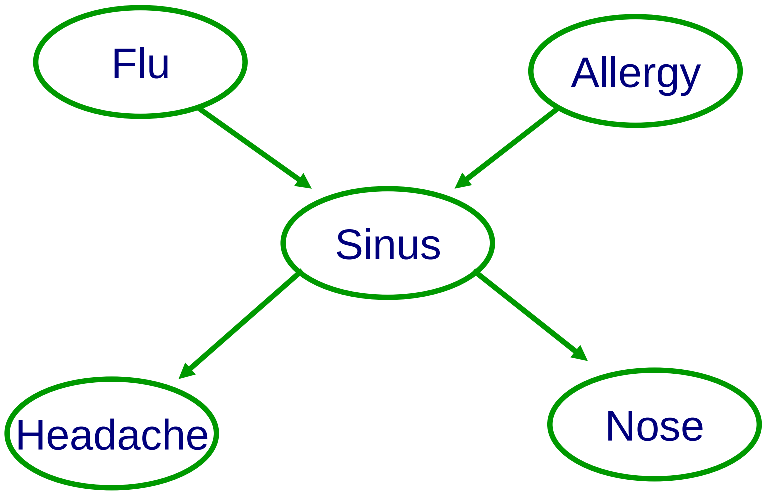
  - Variable X_i has been eliminated!

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:

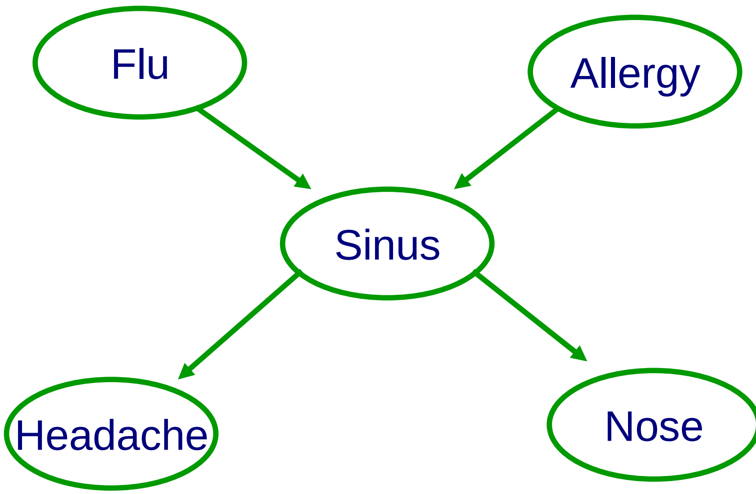
$$x_i^* \in \operatorname*{argmax}_{x_i} \prod_{j=1}^{k} f_j$$

Stochastic simulation – Obtaining a sample from the joint distribution

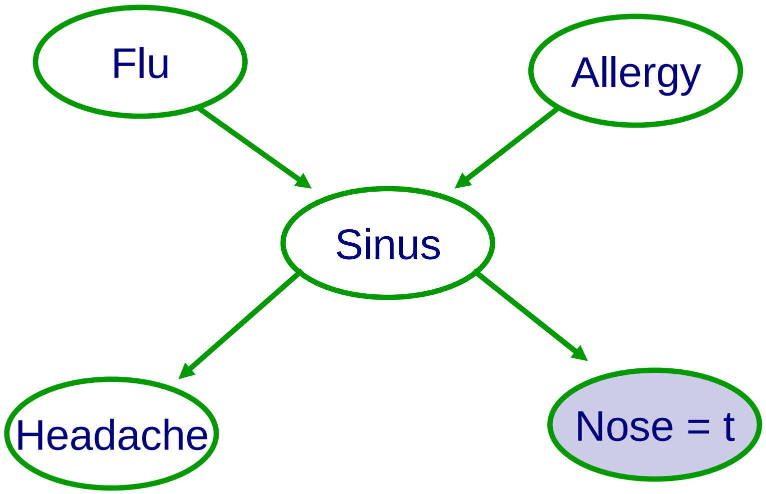


Using stochastic simulation (sampling) to compute P(X)

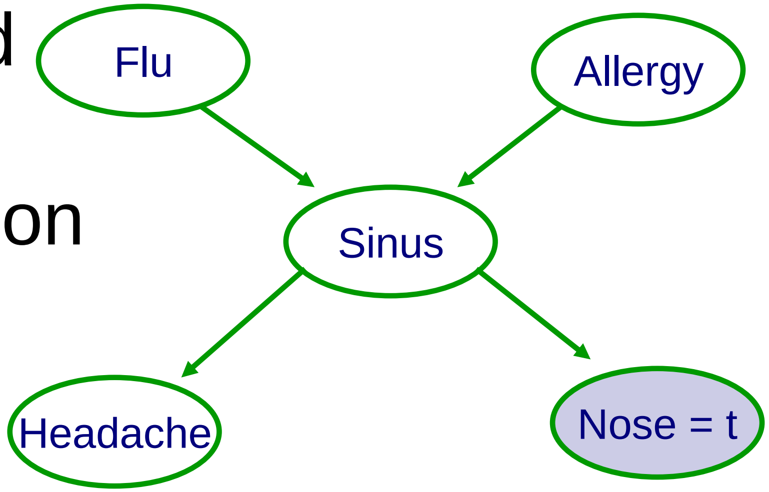
- Given a BN, a query P(X), and number of samples *m*
- Choose a **topological** ordering on variables, e.g., $X_1, \ldots, X_n$
- For j = 1 to m
  - $\{x_1^j, \ldots, x_n^j\}$ will be j^{th} sample
  - For i = 1 to n
    - Sample x_i^j from the distribution $P(X_i|\mathbf{Pa}_{X_i})$, where parents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

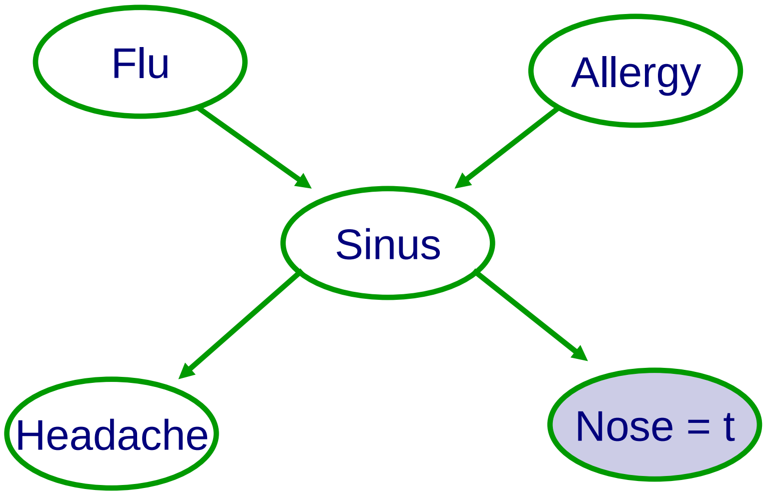


Using rejection sampling to compute P(X|e)

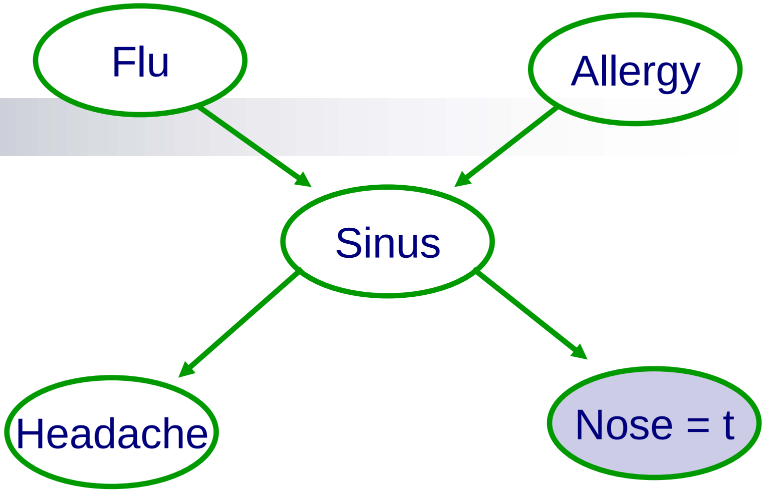


- Given a BN, a query P(X|e), and number of samples *m*
- Choose a **topological** ordering on variables, e.g., $X_1, \ldots, X_n$
- j = 0
- While j < m
  - $\{x_1^j, \ldots, x_n^j\}$ will be j^{th} sample
  - For i = 1 to n
    - Sample x_i^j from the distribution $P(X_i|\mathbf{Pa}_{X_i})$, where parents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