Learning BN tutorial:
ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.pdf
TAN paper:
http://www.cs.huji.ac.il/~nir/Abstracts/FrGG1.html

Bayesian Networks – Inference (continued) Learning

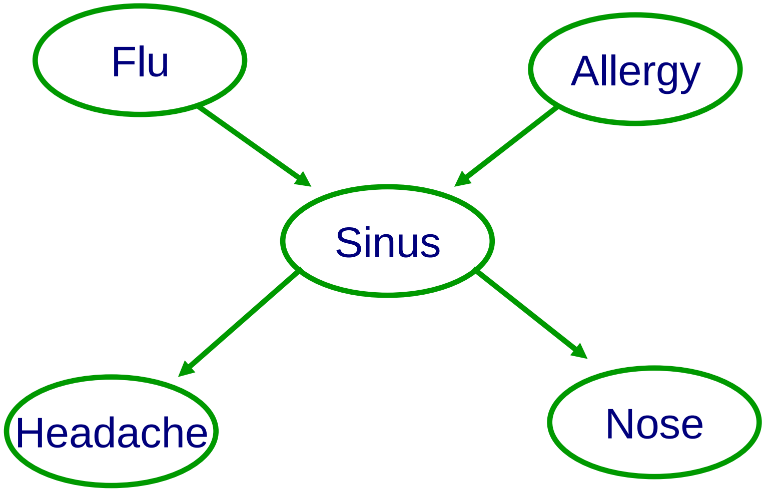
Machine Learning – 10701/15781

Carlos Guestrin
Carnegie Mellon University

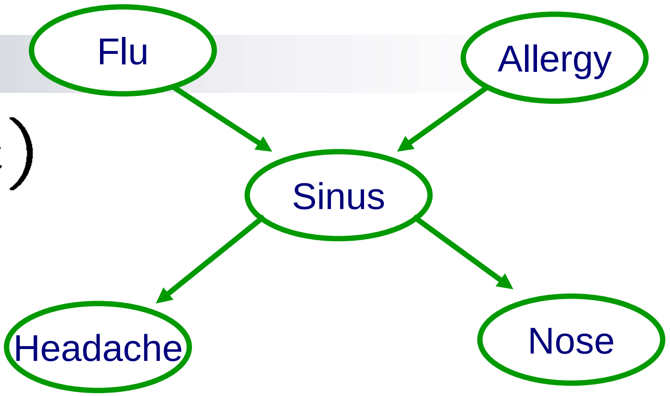
March 23rd, 2005

Review

- Bayesian Networks
  - Compact representation for probability distributions
  - Exponential reduction in number of parameters
- Fast probabilistic inference using variable elimination
  - Compute P(X|e)
  - Time exponential in tree-width, not number of variables
- Today
  - Finding most likely explanation
  - Using sampling for approximate inference
  - Learn BN structure



Most likely explanation (MLE)



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

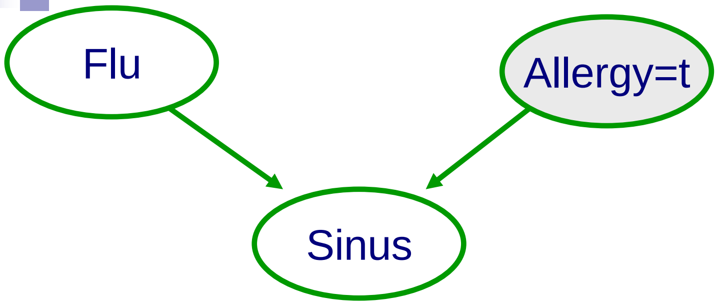
- Using Bayes rule:

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

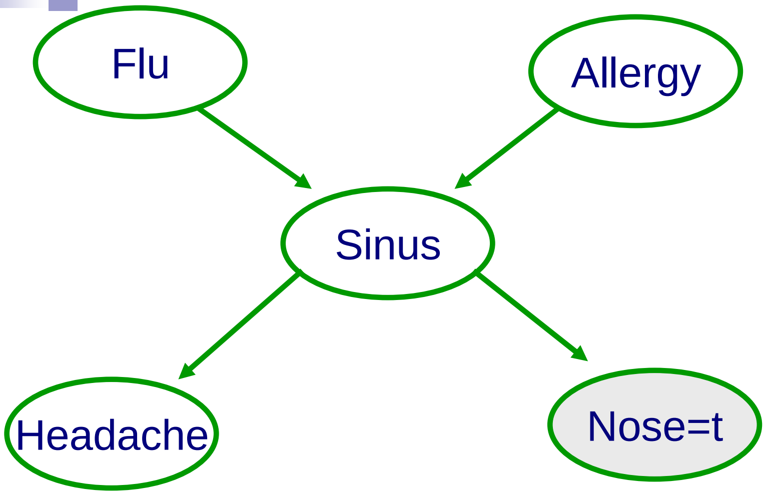
- Normalization irrelevant:

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

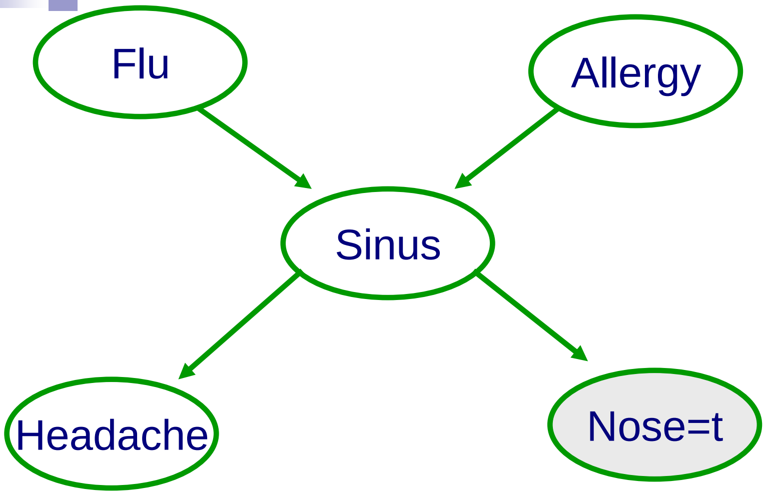
Max-marginalization



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

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

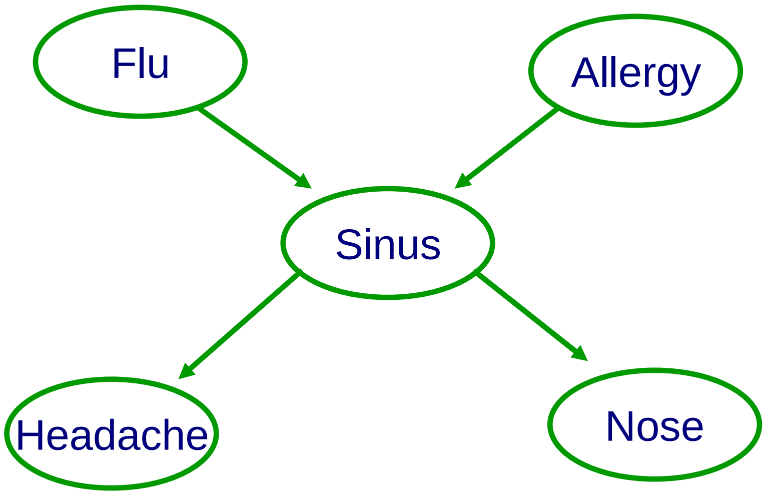
  - Variable X_i has been eliminated!

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:

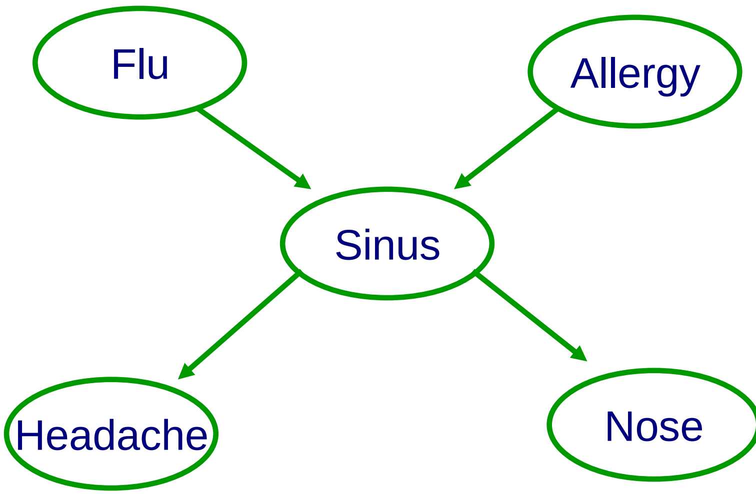
$$x_i^* \in \operatorname*{argmax}_{x_i} \prod_{j=1}^{k} f_j$$

Stochastic simulation – Obtaining a sample from the joint distribution

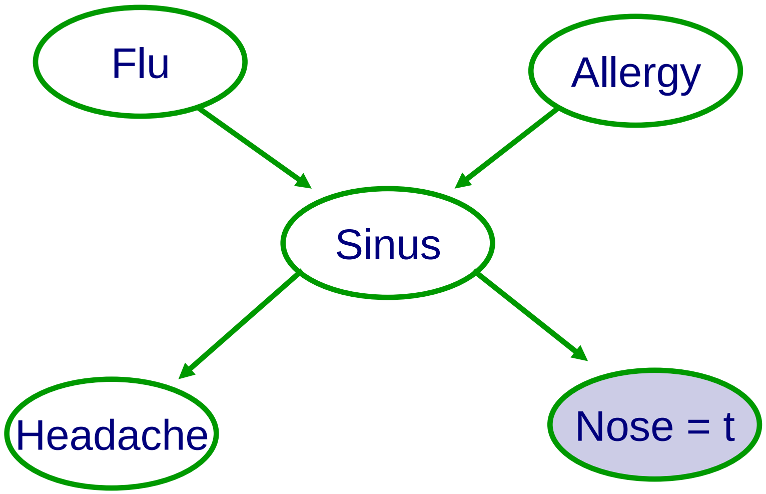


Using stochastic simulation (sampling) to compute P(X)

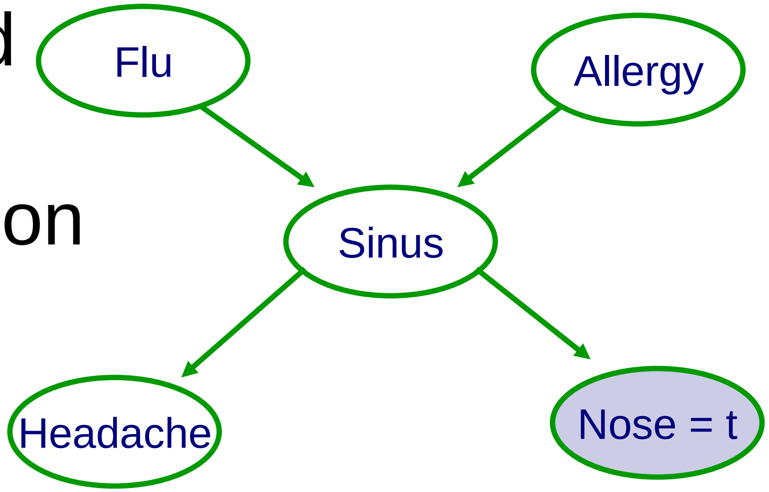
- Given a BN, a query P(X), and number of samples *m*
- Choose a **topological** ordering on variables, e.g., $X_1, \ldots, X_n$
- For j = 1 to m
  - $\{x_1^j, \ldots, x_n^j\}$ will be j^{th} sample
  - For i = 1 to n
    - Sample x_i^j from the distribution $P(X_i|\mathbf{Pa}_{X_i})$, where parents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

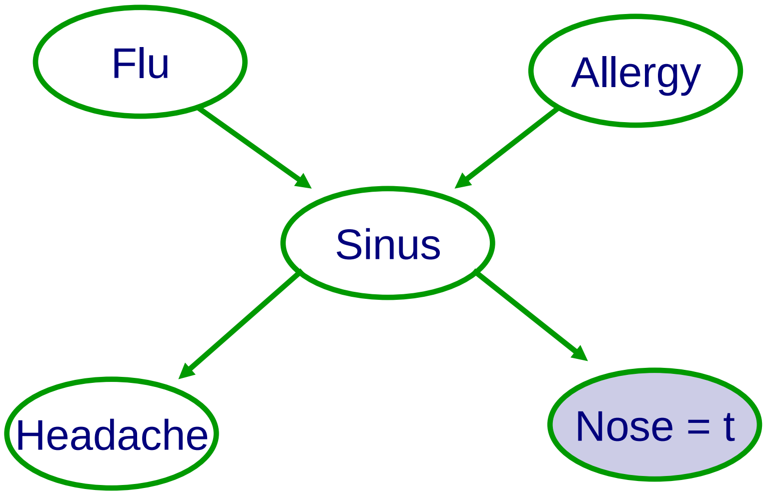


Using rejection sampling to compute P(X|e)

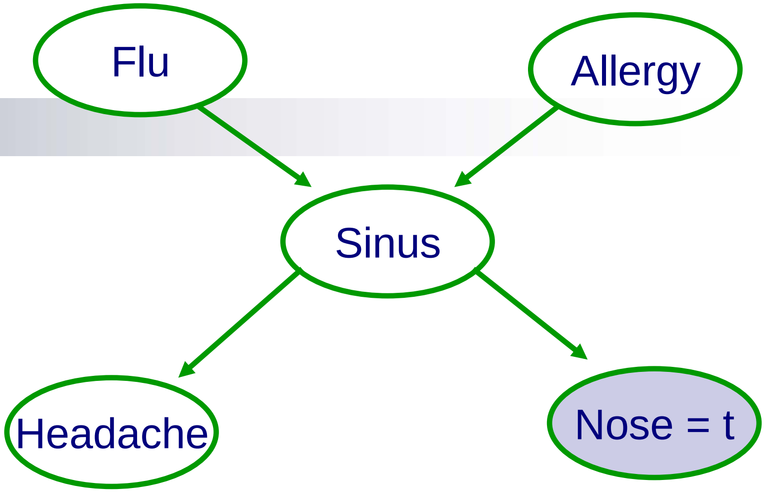


- Given a BN, a query P(X|e), and number of samples *m*
- Choose a **topological** ordering on variables, e.g., $X_1, \ldots, X_n$
- j = 0
- While j < m
  - $\{x_1^j, \ldots, x_n^j\}$ will be j^{th} sample
  - For i = 1 to n
    - Sample x_i^j from the distribution $P(X_i|\mathbf{Pa}_{X_i})$, where parents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

- 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

Where are we?

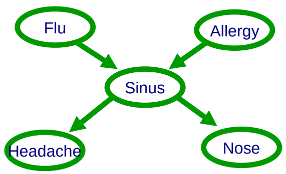
- Bayesian networks
  - Represent exponentially-large probability distributions compactly
- Inference in BNs
  - Exact inference very fast for problems with low tree-width
  - Many approximate inference algorithms, e.g., sampling
- Learning BNs
  - Given structure, estimate parameters
    - Using counts (maximum likelihood), MAP is also possible
  - What about learning structure?

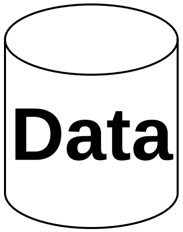
Maximum likelihood (ML) for learning BN structure

Possible structures

Learn parameters using ML

Score structure





$\langle x_1^{(1)},\ldots,x_n^{(1)}\rangle$

…

$\langle x_1^{(m)},\ldots,x_n^{(m)}\rangle$

How many graphs are there?

How many trees are there?

Nonetheless – Efficient optimal algorithm finds best tree

Scoring a tree

Chow-Liu tree learning algorithm

- For each pair of variables X_i, X_j
  - Compute empirical distribution:

$$\hat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$$

  - Compute mutual information:

$$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{P}(x_i, x_j) \log \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i)\hat{P}(x_j)}$$

- Define a graph
  - Nodes $X_1, \ldots, X_n$
  - Edge (i,j) gets weight $\hat{I}(X_i, X_j)$
- Optimal tree BN
  - Compute maximal spanning tree
  - Directions in BN: pick any node as root, breadth-first-search defines directions

Can we extend Chow-Liu

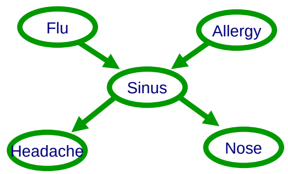
- Tree augmented naïve Bayes (TAN)
  - [Friedman et al. '97]
  - Same as Chow-Liu, but score edges with:

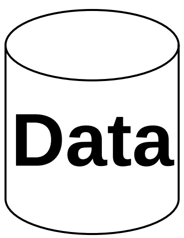
$$\hat{I}(X_i, X_j \mid C) = \sum_{c, x_i, x_j} \hat{P}(c, x_i, x_j) \log \frac{\hat{P}(x_i, x_j \mid c)}{\hat{P}(x_i \mid c)\hat{P}(x_j \mid c)}$$

- (Approximate learning) models with tree-width up to *k*
  - [Narasimhan & Bilmes '04]
  - But, $O(n^{k+1})$…

Scoring general graphical models – Model selection problem

What's the best structure?





$\langle x_1^{(1)},\ldots,x_n^{(1)}\rangle$

…

$\langle x_1^{(m)},\ldots,x_n^{(m)}\rangle$

The more edges, the fewer independence assumptions, the higher the likelihood of the data, but will overfit…

Bayesian score

- Given a structure, distribution over parameters

$$\log P(D \mid S) = \log \int_{\theta_S} P(D \mid S, \theta_S) P(\theta_S | S) d\theta_S$$

- Difficult integral, use Bayes information criterion (BIC) approximation

$$\log P(D \mid S) \approx \log P(D \mid S, \theta_S) - \frac{\text{NumberParameters}(S)}{2} \log m$$

- Note: regularize with MDL score
- Best BN under BIC still NP-hard

Learn BN structure using local search

Starting from Chow-Liu tree

Local search, possible moves:

- Add edge
- Delete edge
- Invert edge

Score using BIC

What you need to know about learning BNs

- Bayesian networks
  - A useful compact **representation** for large probability distributions
- Inference
  - Variable elimination algorithm
  - Sampling – Example of approximate inference
- Learning BNs
  - Maximum likelihood or MAP learns parameters
  - Best tree (Chow-Liu)
  - Best TAN
  - Other BNs, usually local search with BIC score

Acknowledgements

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