Graphical Models

Aarti Singh
Slides Courtesy: Carlos Guestrin

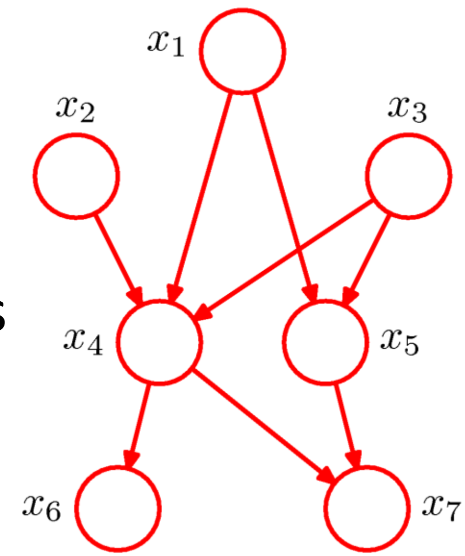
Machine Learning 10-701/15-781
Nov 15, 2010

Directed – Bayesian Networks

- Compact representation for a joint probability distribution
- Bayes Net = Directed Acyclic Graph (DAG) + Conditional Probability Tables (CPTs)
- distribution factorizes according to graph ≡ distribution satisfies local Markov independence assumptions

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

≡ x_k is independent of its non-descendants given its parents pa_k



Directed – Bayesian Networks

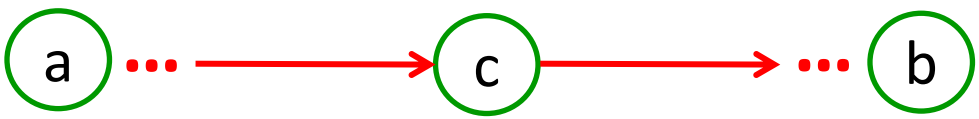
- Graph encodes local independence assumptions (local Markov Assumptions)
- Other independence assumptions can be read off the graph using d-separation
- distribution factorizes according to graph ≡ distribution satisfies all independence assumptions found by d-separation

$$F \iff I$$

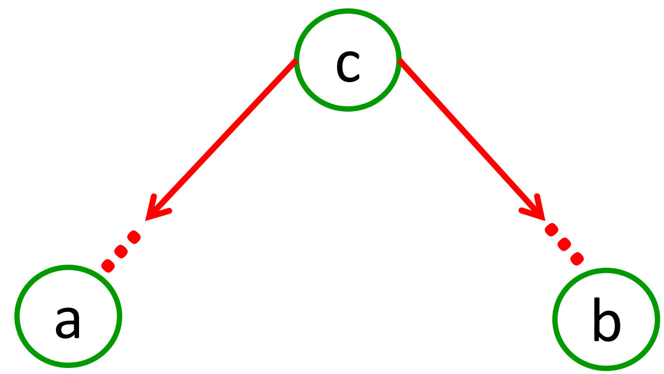
- Does the graph capture all independencies? Yes, for *almost all* distributions that factorize according to graph. More in 10-708

D-separation

- a is D-separated from b by c ≡ $a \perp b | c$
- Three important configurations

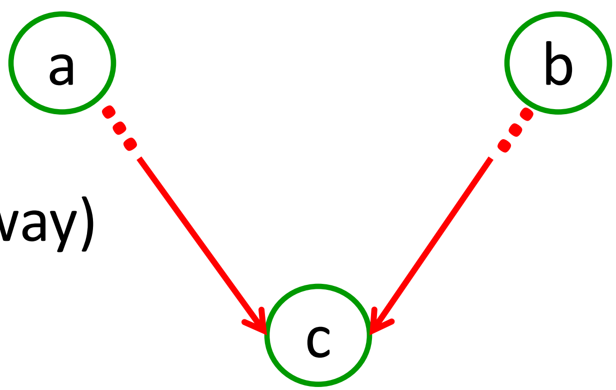


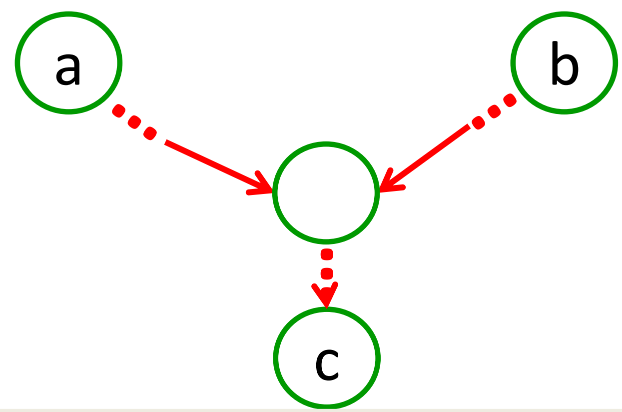
Causal direction $a \perp b | c$



Common cause $a \perp b | c$

V-structure
(Explaining away)
$a \not\perp b | c$



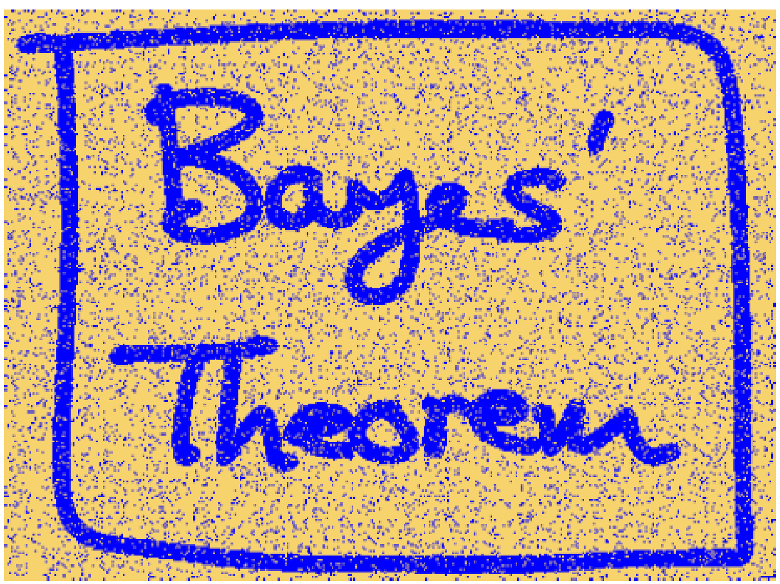


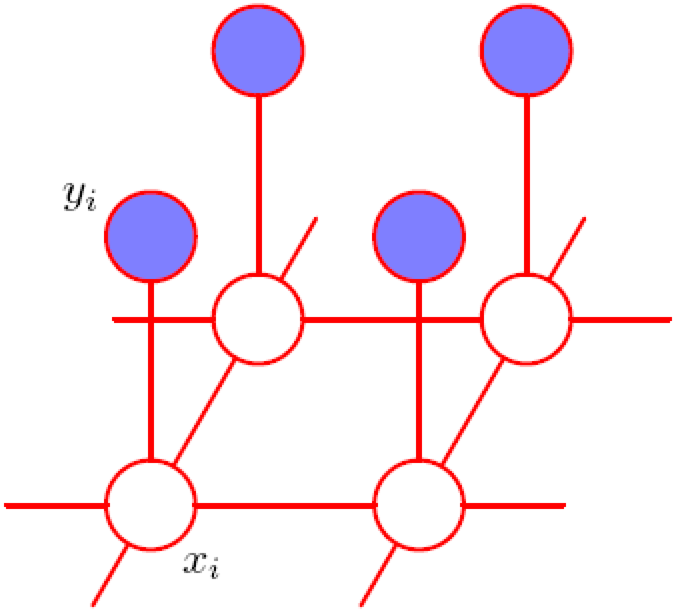
Undirected – Markov Random Fields

- Popular in statistical physics, computer vision, sensor networks, social networks, protein-protein interaction network
- Example – Image Denoising

x_i – value at pixel i

y_i – observed noisy value

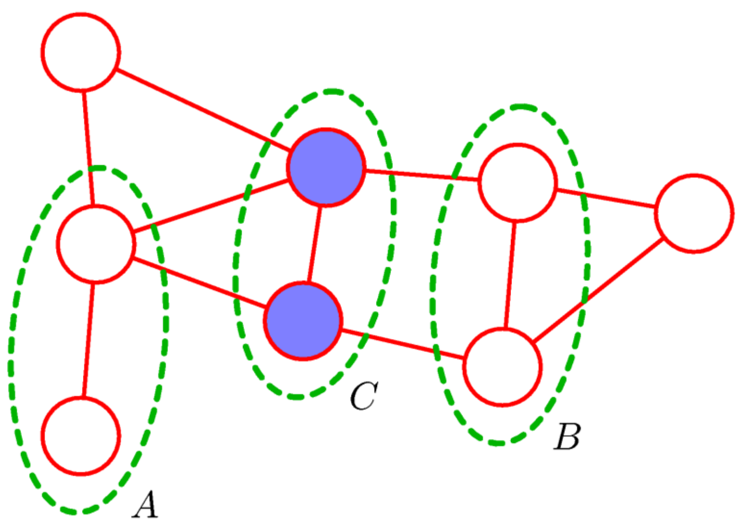




Conditional Independence properties

- No directed edges
- Conditional independence ≡ graph separation
- A, B, C – non-intersecting set of nodes
- $A \perp B | C$ if all paths between nodes in A & B are "blocked"

  i.e. path contains a node z in C.



Factorization

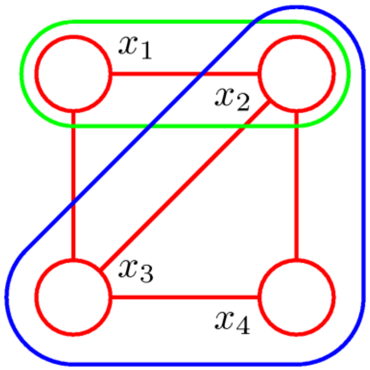
- Joint distribution factorizes according to the graph

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

$\mathcal{C}$ is the set of maximal cliques in the graph

$\psi_C(x_C)$ is a potential function on the clique x_C

Arbitrary positive function



Clique, $x_C = \{x_1, x_2\}$

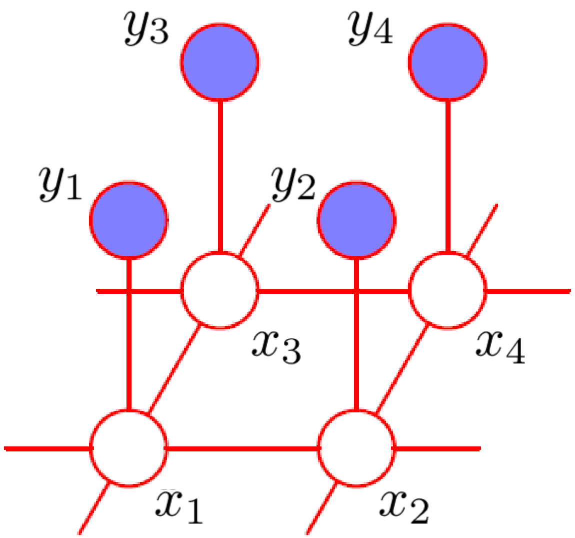
Maximal clique $x_C = \{x_2, x_3, x_4\}$

normalization factor

$$Z = \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

typically NP-hard to compute

MRF Example



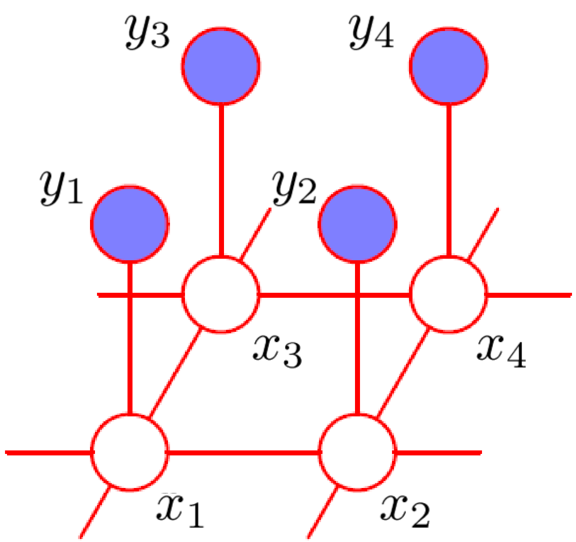
$$P(x,y) \propto \Psi(x_1,x_2)\Psi(x_1,x_3)\Psi(x_2,x_4)\Psi(x_3,x_4)\prod_{i=1}^{4}\Psi(x_i,y_i)$$

Often $\psi_C(\mathbf{x}_C) = \exp\{-E(\mathbf{x}_C)\}$

Energy of the clique (e.g. lower if variables in clique take similar values)

$$p(\mathbf{x}) = \prod_{C\in\mathcal{C}} \exp\{-E(\mathbf{x}_C)\} = \exp\{-\sum_{C\in\mathcal{C}} E(\mathbf{x}_C)\}$$

MRF Example



Ising model:

cliques are edges $x_C = \{x_i, x_j\}$
binary variables $x_i \in \{-1,1\}$

$$\psi_C(\mathbf{x}_C) = \exp\{\beta \underbrace{x_i x_j}\}$$

1 if $x_i = x_j$
-1 if $x_i \neq x_j$

$$p(\mathbf{x}) = \prod_{(i,j)\in E} \exp\{\beta x_i x_j\} = \exp\{\sum_{(i,j)\in E} \beta x_i x_j\}$$

Probability of assignment is higher if neighbors x_i and x_j are same

Hammersley-Clifford Theorem

- Set of distributions that factorize according to the graph - $\mathcal{F}$
- Set of distributions that respect conditional independencies implied by graph-separation – $\mathcal{I}$

$$\mathcal{I} \Rightarrow \mathcal{F}$$

Important because: **Given independencies of *P* can get MRF structure *G***

$$\mathcal{I} \Leftarrow \mathcal{F}$$

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

What you should know…

- Graphical Models: Directed Bayesian networks, Undirected Markov Random Fields
  - A compact **representation** for large probability distributions
  - Not an algorithm
- Representation of a BN, MRF
  - Variables
  - Graph
  - CPTs
- Why BNs and MRFs are useful
- D-separation (conditional independence) & factorization

Topics in Graphical Models

- Representation
  - Which joint probability distributions does a graphical model represent?
- Inference
  - How to answer questions about the joint probability distribution?
    - Marginal distribution of a node variable
    - Most likely assignment of node variables
- Learning
  - How to learn the parameters and structure of a graphical model?

Inference

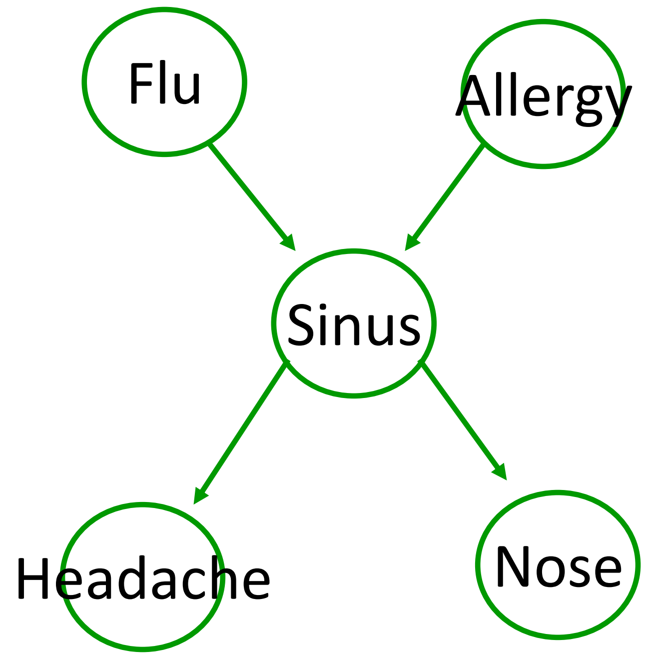
- Possible queries:

1) Marginal distribution e.g. P(S)

   Posterior distribution e.g. P(F|H=1)

2) Most likely assignment of nodes

   $$\arg\max_{f,a,s,n} P(F=f,A=a,S=s,N=n|H=1)$$

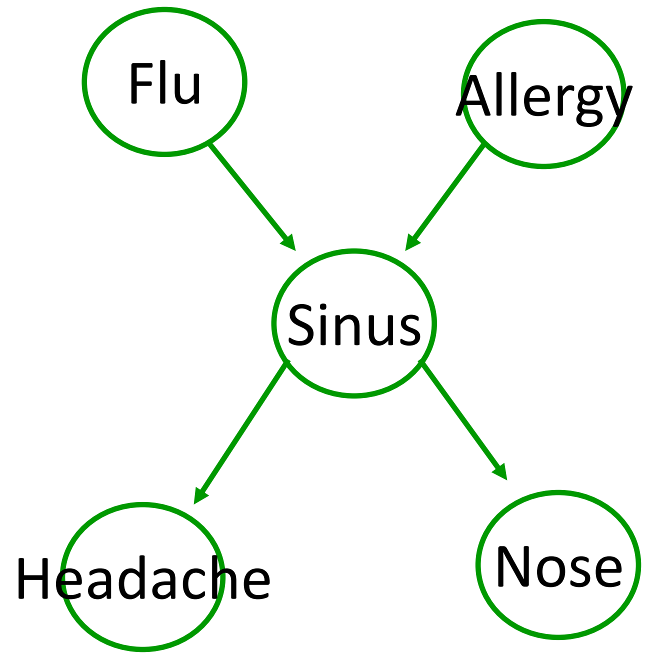


Inference

- Possible queries:

1) Marginal distribution e.g. $P(S)$

   Posterior distribution e.g. $P(F|H=1)$



$P(F|H=1)$?

$$P(F|H=1) = \frac{P(F, H=1)}{P(H=1)}$$

$$= \frac{P(F, H=1)}{\sum_f P(F=f, H=1)}$$

$$\propto P(F, H=1)$$

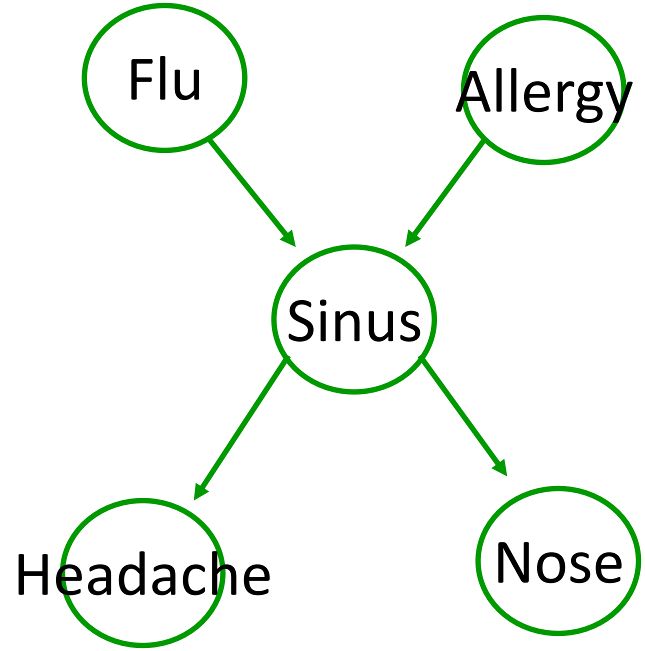
will focus on computing this, posterior will follow with only constant times more effort

Marginalization

Need to marginalize over other vars

$$P(S) = \sum_{f,a,n,h} P(f,a,S,n,h)$$

$$P(F,H=1) = \underbrace{\sum_{a,s,n}}_{2^3 \text{ terms}} P(F,a,s,n,H=1)$$

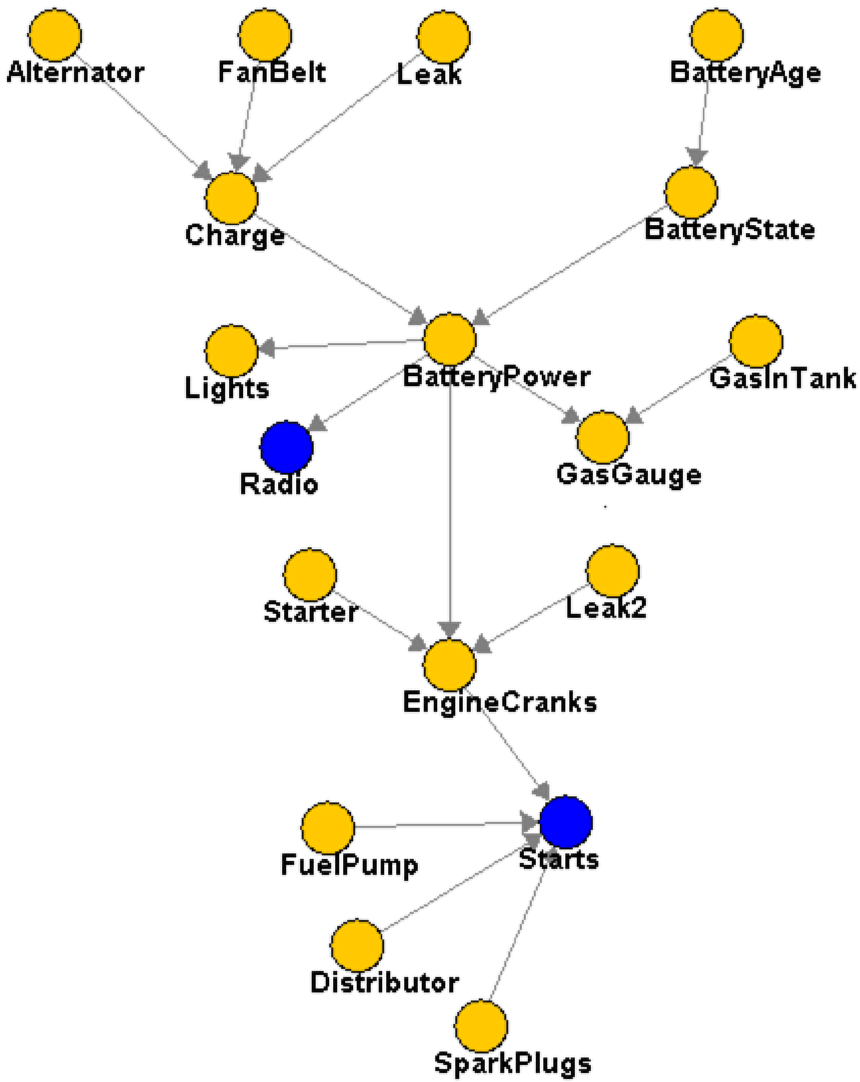


To marginalize out n binary variables,

need to sum over 2^n terms

Inference seems exponential in number of variables!
Actually, inference in graphical models is NP-hard ☹

Bayesian Networks Example



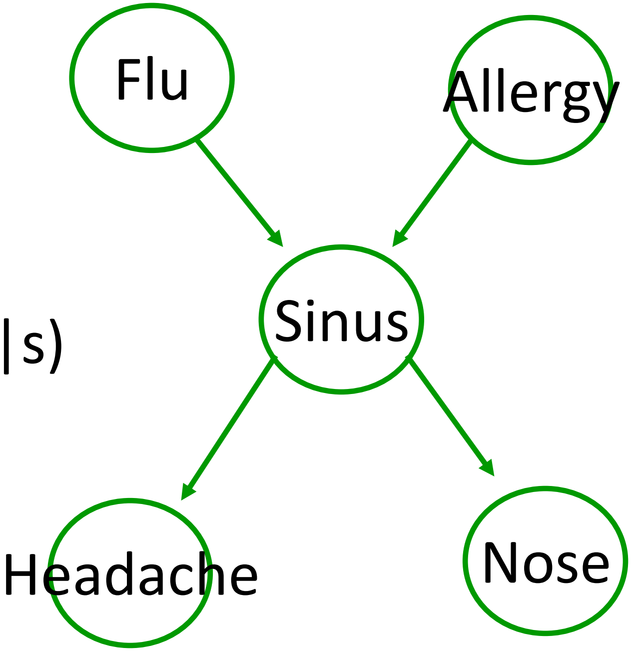
- 18 binary attributes
- Inference
  - P(BatteryAge|Starts=f)
- need to sum over 2^{16} terms!
- Not impressed?
  - HailFinder BN – more than 3^{54} = 58149737003040059690390169 terms

Fast Probabilistic Inference

$$P(F,H=1) = \sum_{a,s,n} P(F,a,s,n,H=1)$$

$$= \sum_{a,s,n} P(F)P(a)P(s|F,a)P(n|s)P(H=1|s)$$

$$= P(F) \sum_{a} P(a) \sum_{s} P(s|F,a)P(H=1|s) \sum_{n} P(n|s)$$



Push sums in as far as possible

Distributive property: $\underbrace{x_1 z + x_2 z}_{\text{2 multiply}} = \underbrace{z(x_1+x_2)}_{\text{1 mulitply}}$

Fast Probabilistic Inference

$$P(F,H=1) = \sum_{a,s,n} P(F,a,s,n,H=1)$$

8 values x 4 multiplies

$$= \sum_{a,s,n} P(F)P(a)P(s|F,a)P(n|s)P(H=1|s)$$

$$= P(F) \sum_a P(a) \sum_s P(s|F,a)P(H=1|s) \cancelto{1}{\sum_n P(n|s)}$$

$$= P(F) \sum_a P(a) \sum_s P(s|F,a)P(H=1|s)$$

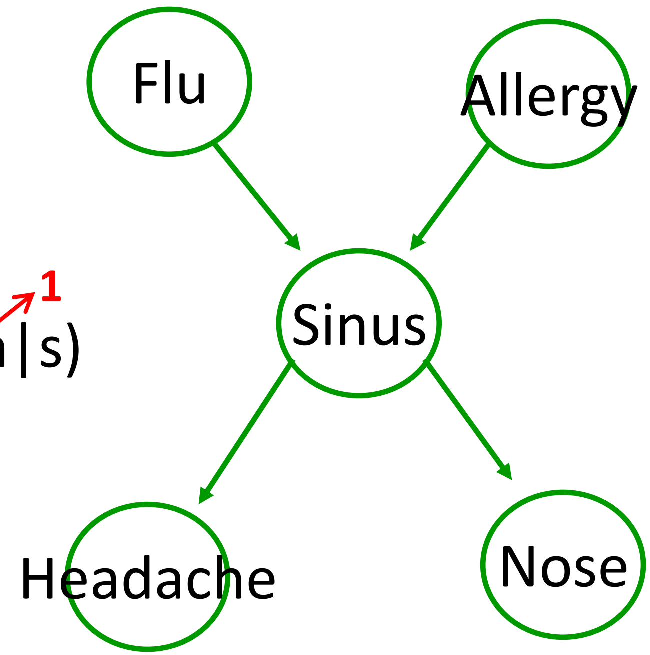
4 values x 1 multiply

$$= P(F) \sum_a P(a)\, g_1(F,a)$$

2 values x 1 multiply

$$= P(F)\, g_2(F)$$

1 multiply



32 multiplies vs. 7 multiplies

2^n vs. $n\ 2^k$

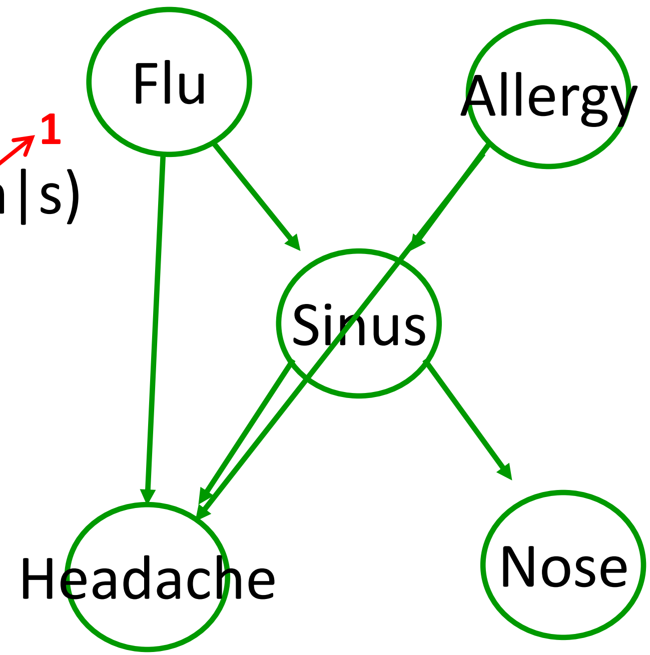
k – scope of largest factor

(Potential for) exponential reduction in computation!

Fast Probabilistic Inference – Variable Elimination

$$P(F,H=1) = \sum_{a,s,n} P(F)P(a)P(s|F,a)P(n|s)P(H=1|s)$$

$$= P(F) \underbrace{\sum_{a} P(a) \underbrace{\sum_{s} P(s|F,a)P(H=1|s)}_{P(H=1|F,a)} \cancelto{1}{\sum_{n} P(n|s)}}_{P(H=1|F)}$$

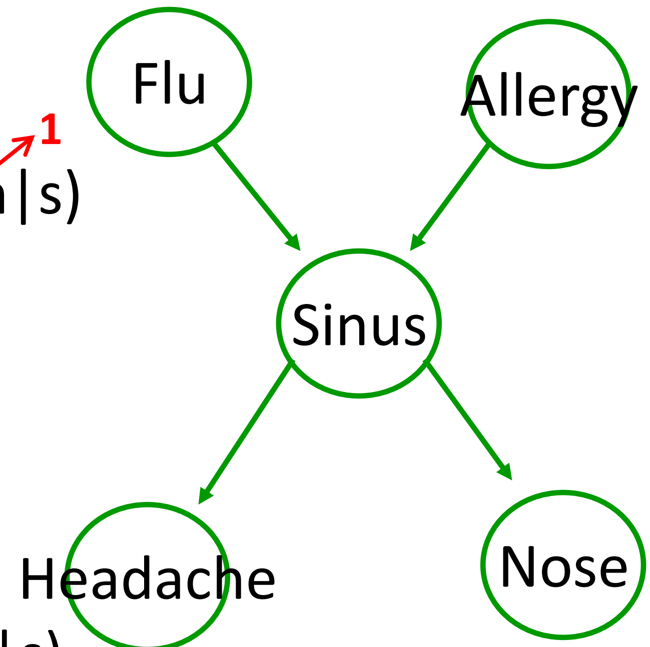


(Potential for) exponential reduction in computation!

Variable Elimination – Order can make a HUGE difference

$$P(F,H=1) = \sum_{a,s,n} P(F)P(a)P(s|F,a)P(n|s)P(H=1|s)$$

$$= P(F) \underbrace{\sum_a P(a) \underbrace{\sum_s P(s|F,a)P(H=1|s)}_{P(H=1|F,a)}}_{P(H=1|F)} \cancelto{1}{\sum_n P(n|s)}$$

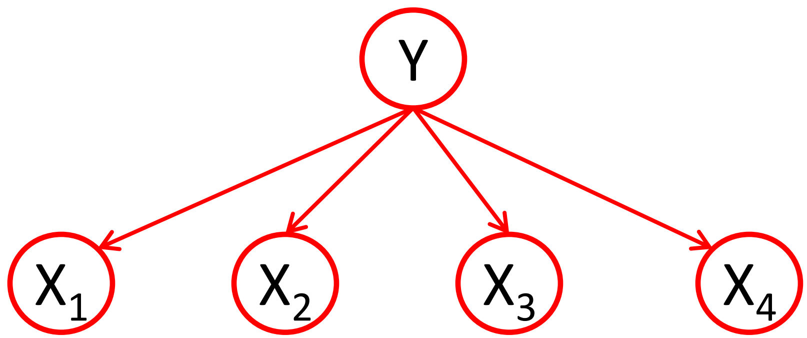


$$P(F,H=1) = P(F) \sum_a P(a) \sum_n \underbrace{\sum_s P(s|F,a)P(n|s)P(H=1|s)}_{g(F,a,n)}$$

3 – scope of largest factor

(Potential for) exponential reduction in computation!

Variable Elimination – Order can make a HUGE difference



$$
\begin{aligned}
P(X_1) &= \sum_{Y,X_2,\ldots,X_n} P(Y)P(X_1|Y)\prod_{i=2}^{n} P(X_i|Y) \\
&= \sum_{Y,X_3,\ldots,X_n} P(Y)P(X_1|Y)\prod_{i=3}^{n} P(X_i|Y)\underbrace{\sum_{X_2} P(X_2|Y)}_{g(Y)}
\end{aligned}
$$

1 – scope of largest factor

$$
= \sum_{X_2,\ldots,X_n} \underbrace{\sum_{Y} P(Y)P(X_1|Y)\prod_{i=2}^{n} P(X_i|Y)}_{g(X_1,X_2,..,X_n)}
$$

n – scope of largest factor

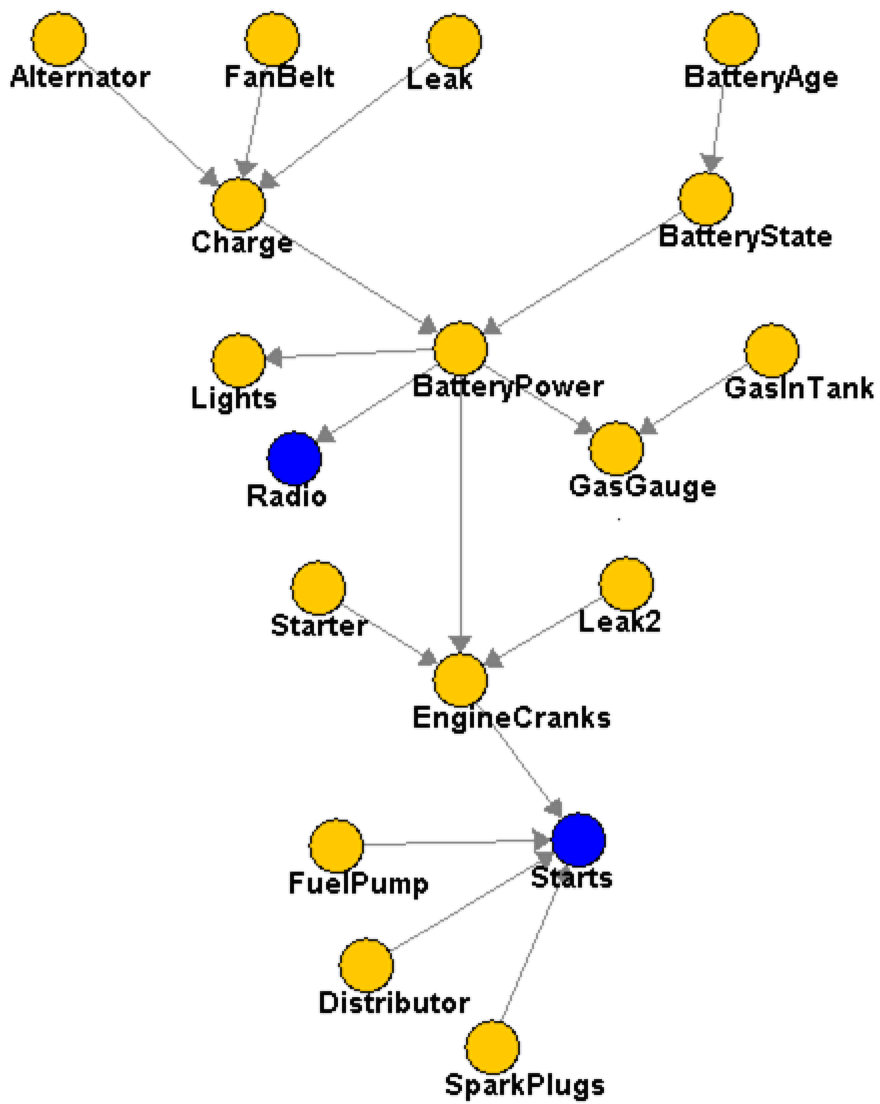
Variable Elimination Algorithm

- Given BN – DAG and CPTs (initial factors – $p(x_i|pa_i)$ for i=1,..,n)
- Given Query $P(X|e) \equiv P(X,e)$ X – set of variables
- Instantiate evidence e e.g. set H=1 **IMPORTANT!!!**
- Choose an ordering on the variables e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{X,e\}$
  - Collect factors $g_1,\ldots,g_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

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

  - Variable X_i has been eliminated!
  - Remove $g_1,\ldots,g_k$ from set of factors but add g
- Normalize $P(X,e)$ to obtain $P(X|e)$

Complexity for (Poly)tree graphs



Variable elimination order:

- Consider undirected version
- Start from "leaves" up
- find topological order
- eliminate variables in reverse order

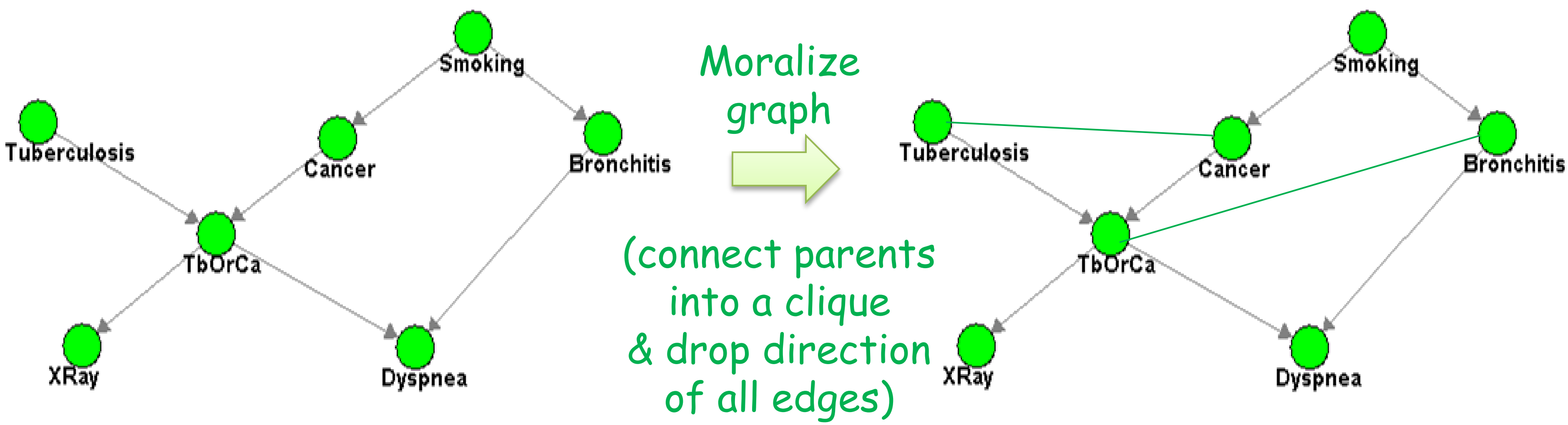
Does not create any factors bigger than original CPTs

For polytrees, inference is linear in # variables (vs. exponential in general)!

Complexity for graphs with loops

- Loop – undirected cycle

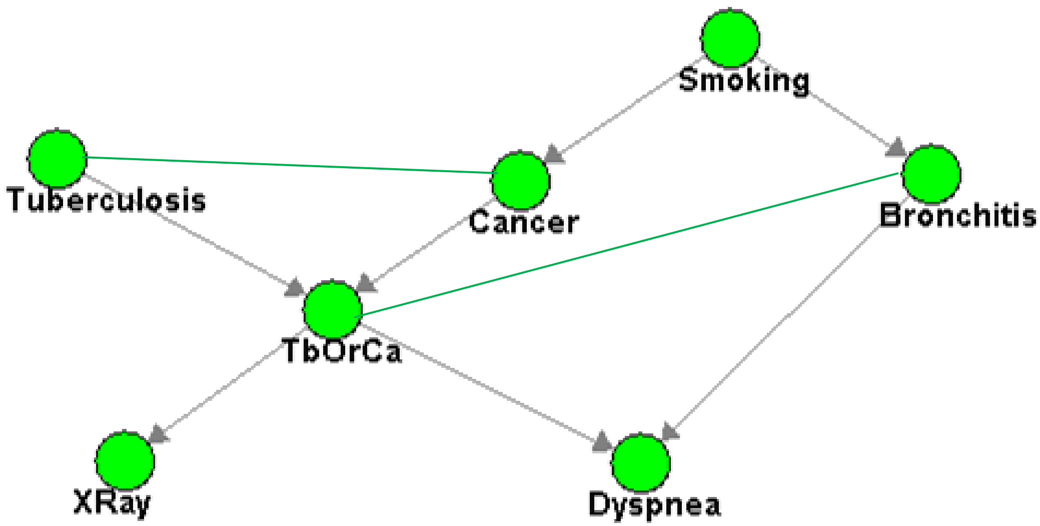
Linear in # variables but exponential in size of largest factor generated!



When you eliminate a variable, add edges between its neighbors

Complexity for graphs with loops

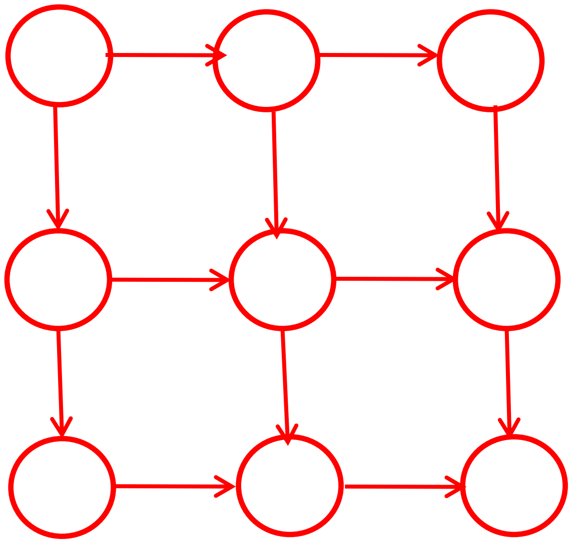
- Loop – undirected cycle



Var eliminated	Factor generated
S	$g_1(C,B)$
B	$g_2(C,O,D)$
D	$g_3(C,O)$
C	$g_4(T,O)$
T	$g_5(O,X)$
O	$g_6(X)$

Linear in # variables but exponential in size of largest factor generated ~ tree-width (max clique size-1) in resulting graph!

Example: Large tree-width with small number of parents



At most 2 parents per node, but tree width is O(√n)

Compact representation $\not\Rightarrow$ Easy inference ☹

Choosing an elimination order

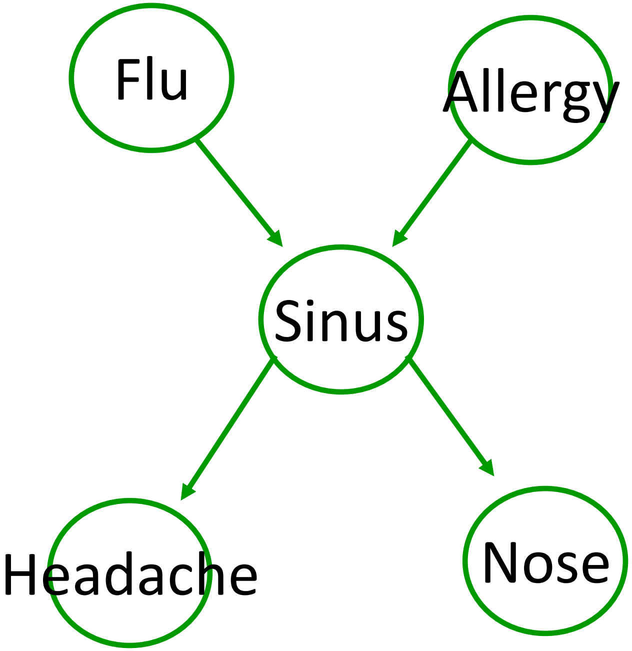
- Choosing best order is NP-complete
  - Reduction from MAX-Clique
- Many good heuristics (some with guarantees)
- Ultimately, can't beat NP-hardness of inference
  - Even optimal order can lead to exponential variable elimination computation
- In practice
  - Variable elimination often very effective
  - Many (many many) approximate inference approaches available when variable elimination too expensive

Inference

- Possible queries:

2) Most likely assignment of nodes

$$\arg\max_{f,a,s,n} P(F=f,A=a,S=s,N=n|H=1)$$



Use Distributive property:

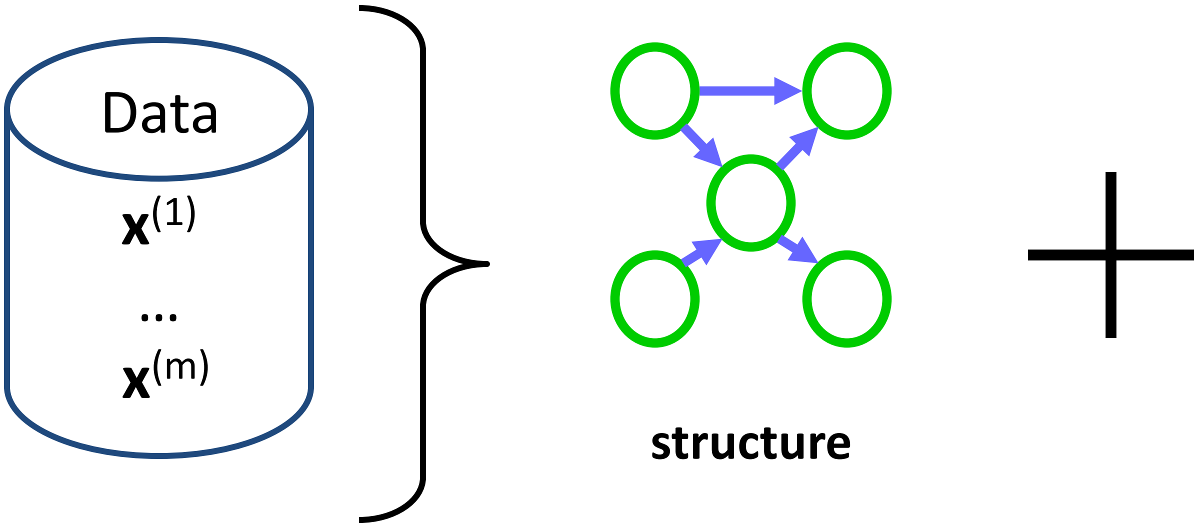
$$\max(x_1 z, x_2 z) = z \max(x_1,x_2)$$

2 multiply 1 mulitply

Topics in Graphical Models

- Representation
  - Which joint probability distributions does a graphical model represent?
- Inference
  - How to answer questions about the joint probability distribution?
    - Marginal distribution of a node variable
    - Most likely assignment of node variables
- Learning
  - How to learn the parameters and structure of a graphical model?

Learning



CPTs –

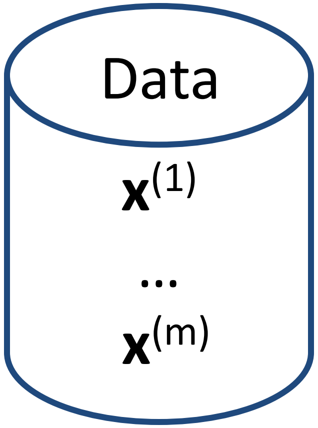
$P(X_i \mid \mathbf{Pa}_{X_i})$

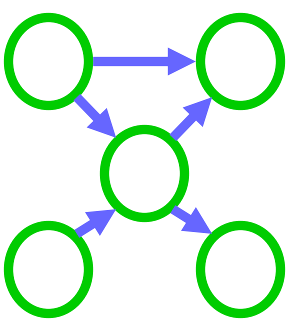
parameters

Given set of m independent samples (assignments of random variables),

find the best (most likely?) Bayes Net (graph Structure + CPTs)

Learning the CPTs (given structure)





For each discrete variable X_k

Compute MLE or MAP estimates for

$$p(x_k|\mathrm{pa}_k)$$

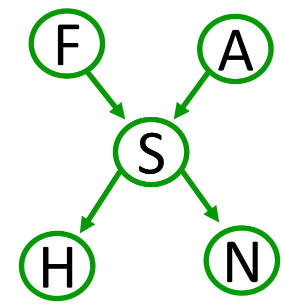
Recall

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

MAP: Add psuedocounts

MLEs decouple for each CPT in Bayes Nets

- Given structure, log likelihood of data



$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

$$= \log \prod_{j=1}^{m} P(f^{(j)})P(a^{(j)})P(s^{(j)}|f^{(j)},a^{(j)})P(h^{(j)}|s^{(j)})P(n^{(j)}|s^{(j)})$$

$$= \sum_{j=1}^{m} [\log P(f^{(j)}) + \log P(a^{(j)}) + \log P(s^{(j)}|f^{(j)},a^{(j)}) + \log P(h^{(j)}|s^{(j)}) + \log P(n^{(j)}|s^{(j)})]$$

$$= \underbrace{\sum_{j=1}^{m} \log P(f^{(j)})}_{\theta_F} + \underbrace{\sum_{j=1}^{m} \log P(a^{(j)})}_{\theta_A} + \underbrace{\sum_{j=1}^{m} \log P(s^{(j)}|f^{(j)},a^{(j)})}_{\theta_{F,A}} + \underbrace{\sum_{j=1}^{m} \log P(h^{(j)}|s^{(j)})}_{\theta_{H|S}} + \underbrace{\sum_{j=1}^{m} \log P(n^{(j)}|s^{(j)})}_{\theta_{N|S}}]$$

Depends only on

Can computer MLEs of each parameter independently!

Information theoretic interpretation of MLE

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log P\left(X_i = x_i^{(j)} \mid \mathbf{Pa}_{X_i} = \mathbf{x}_{\mathbf{Pa}_{X_i}}^{(j)}\right)$$

$$= \sum_{i=1}^{n} \sum_{x_i} \sum_{\mathbf{x}_{\mathbf{Pa}_{X_i}}} \text{count}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{x}_{\mathbf{Pa}_{X_i}}) \log P\left(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{x}_{\mathbf{Pa}_{X_i}}\right)$$

Plugging in MLE estimates: ML score

$$\log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log \widehat{P}\left(x_i^{(j)} \mid \mathbf{x}_{\mathbf{Pa}_{X_i}}^{(j)}\right)$$

$$= m \sum_{i=1}^{n} \sum_{x_i} \sum_{\mathbf{x}_{\mathbf{Pa}_{X_i}}} \widehat{P}(x_i, \mathbf{x}_{\mathbf{Pa}_{X_i}}) \log \widehat{P}\left(x_i \mid \mathbf{x}_{\mathbf{Pa}_{X_i}}\right)$$

Reminds of entropy

Information theoretic interpretation of MLE

$$\log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = m \sum_{i=1}^{n} \sum_{x_i} \sum_{\mathbf{x}_{\mathbf{Pa}_{X_i}}} \widehat{P}(x_i, \mathbf{x}_{\mathbf{Pa}_{X_i}}) \log \widehat{P}\left(x_i \mid \mathbf{x}_{\mathbf{Pa}_{X_i}}\right)$$

$$= -m \sum_{i=1}^{n} \widehat{H}(X_i \mid \mathbf{Pa}_{X_i})$$

$$= m \sum_{i=1}^{n} [\widehat{I}(X_i, \mathbf{Pa}_{X_i}) - \underbrace{\widehat{H}(X_i)}_{\text{Doesn't depend on graph structure } \mathcal{G}}]$$

ML score for graph structure $\mathcal{G}$

$$\arg\max_{\mathcal{G}} \log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \arg\max_{\mathcal{G}} \sum_{i=1}^{n} \widehat{I}(X_i, \mathbf{Pa}_{X_i})$$

ML – Decomposable Score

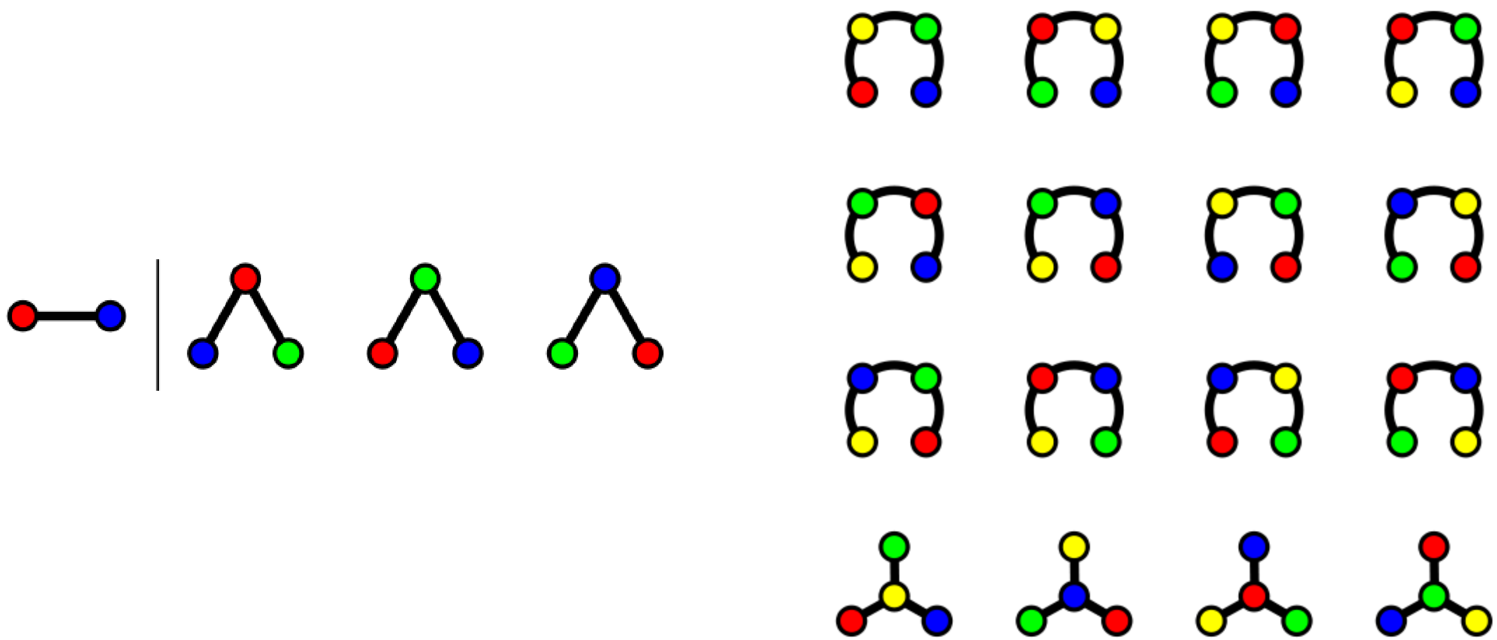
- Log data likelihood

$$\log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = m \sum_{i=1}^{n} [\widehat{I}(X_i, \mathbf{Pa}_{X_i}) - \widehat{H}(X_i)]$$

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!
  - Score(G : D) = $\sum_i$ FamScore(X_i | $\mathbf{Pa}_{X_i}$: D)

How many trees are there?

- Trees – every node has at most one parent
- n^{n-2} possible trees (Cayley's Theorem)

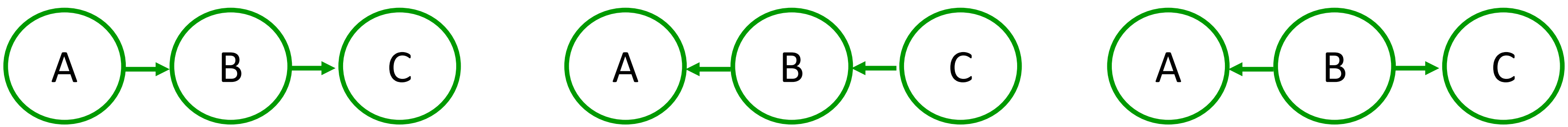


Nonetheless – Efficient optimal algorithm finds best tree!

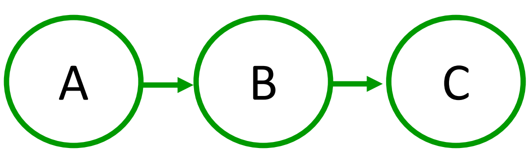
Scoring a tree

$$\arg\max_{\mathcal{G}} \log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \arg\max_{\mathcal{G}} \sum_{i=1}^{n} \widehat{I}(X_i, \mathbf{Pa}_{X_i})$$

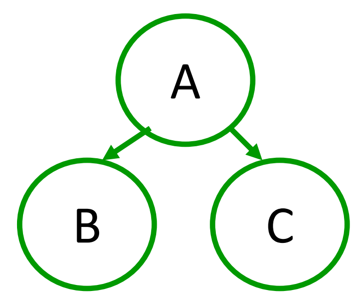
Equivalent Trees (same score): I(A,B) + I(B,C)



Score provides indication of structure:



I(A,B) + I(B,C)



I(A,B) + I(A,C)

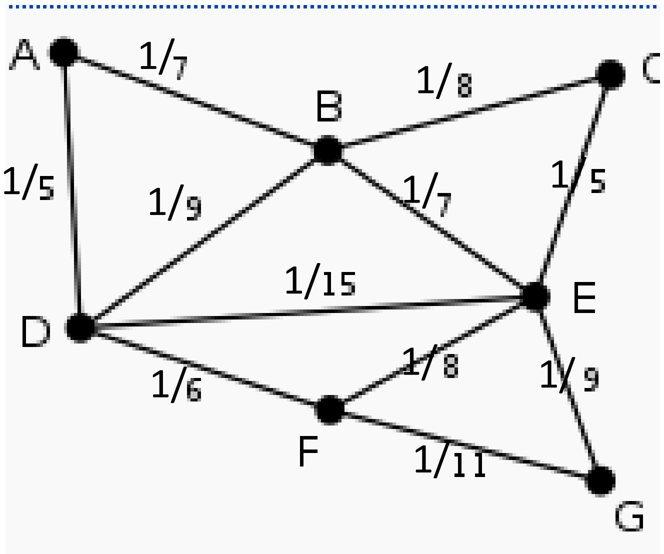
Chow-Liu algorithm

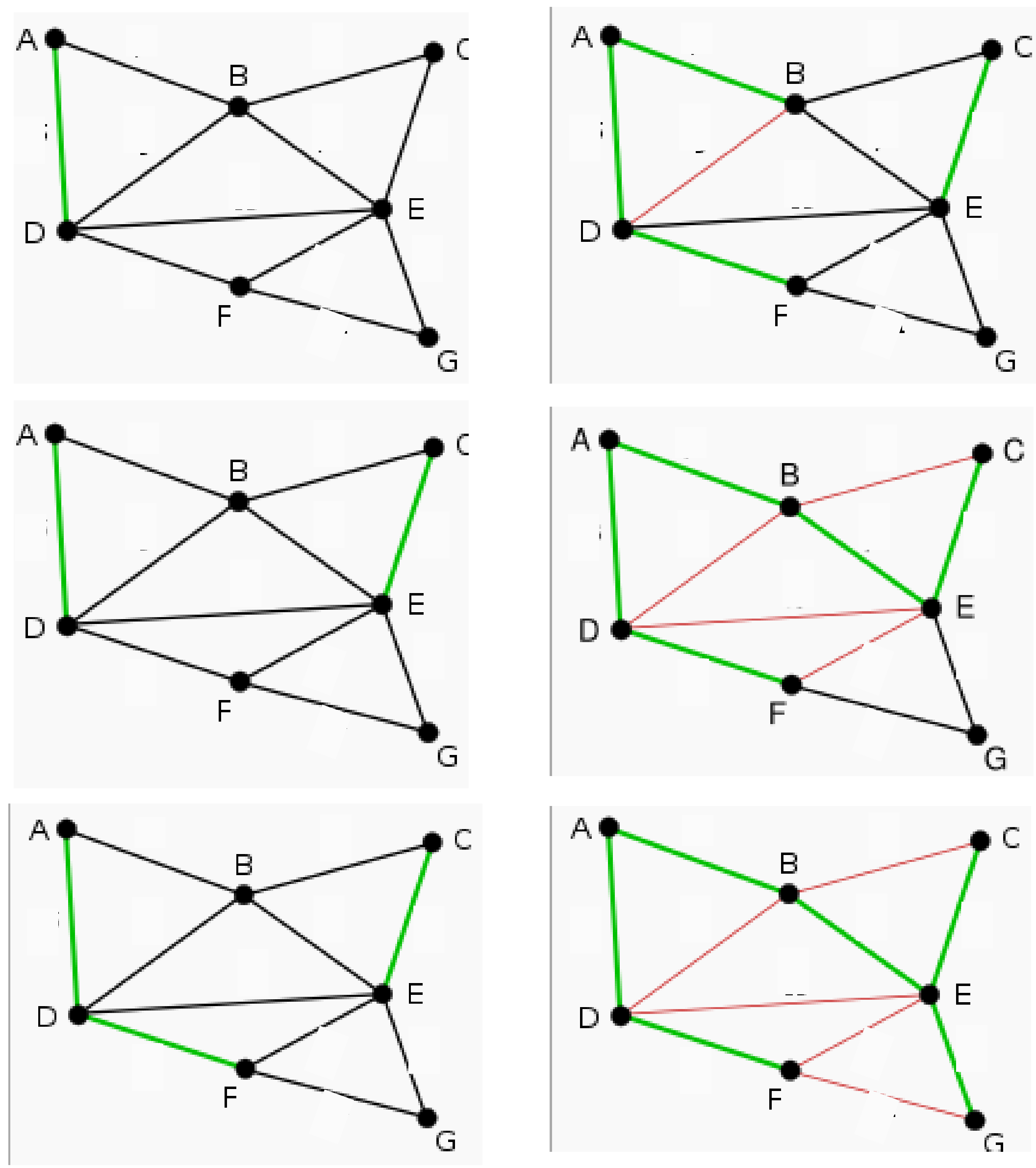
- For each pair of variables X_i, X_j
  - Compute empirical distribution: $\widehat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$
  - Compute mutual information:

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

- Define a graph
  - Nodes $X_1,\ldots,X_n$
  - Edge (i,j) gets weight $\widehat{I}(X_i, X_j)$
- Optimal tree BN
  - Compute maximum weight spanning tree (e.g. Prim's, Kruskal's algorithm O(nlog n))
  - Directions in BN: pick any node as root, breadth-first-search defines directions

Chow-Liu algorithm example





Scoring general graphical models

- Graph that maximizes ML score -> complete graph!
- Information never hurts

  $H(A|B) \geq H(A|B,C)$

- Adding a parent always increases ML score

  $I(A,B,C) \geq I(A,B)$

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

- Why does ML for trees work?

  Restricted model space – tree graph

Regularizing

- Model selection
  - Use MDL (Minimum description length) score
  - BIC score (Bayesian Information criterion)
- Still NP –hard
- Mostly heuristic (exploit score decomposition)
- Chow-Liu: provides best tree approximation to any distribution.
- Start with Chow-Liu tree. Add, delete, invert edges. Evaluate BIC score

What you should know

- Learning BNs
  - Maximum likelihood or MAP learns parameters
  - ML score
    - Decomposable score
    - Information theoretic interpretation (Mutual information)
  - Best tree (Chow-Liu)
  - Other BNs, usually local search with BIC score