Reading:
Chapters 5&6 of Koller&Friedman

Variable elimination

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

September 26th, 2005

Announcements

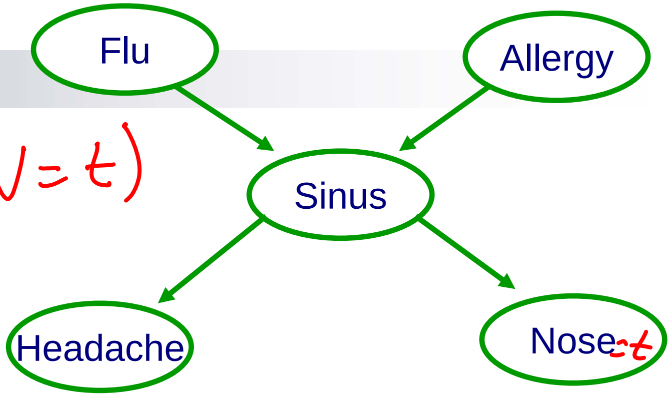
- Waiting List
  - Anyone still wants to be registered?

Inference in BNs hopeless?

- In general, yes!
  - Even approximate!
- In practice
  - Exploit structure
  - Many effective approximation algorithms (some with guarantees)
- For now, we'll talk about exact inference
  - Approximate inference later this semester

$P(e) = \sum_x P(x,e)$, give you $P(X,e)$, normalize by $\sum_x P(x,e) = P(e)$

General probabilistic inference



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

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

- Using def. of cond. prob.:

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

doesn't depend on X

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

- Normalization:

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

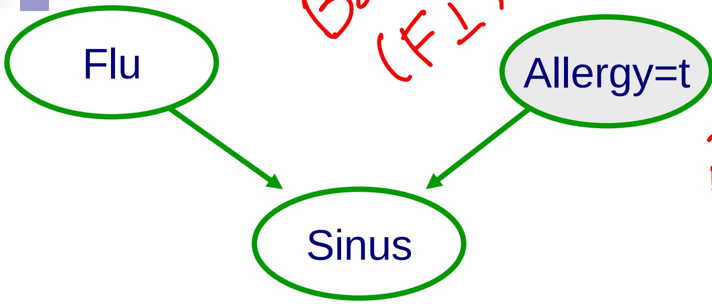
compute $P(F=t, N=t)$, $P(F=f, N=t)$ normalize

$P(F \mid N=t)$

Marginalization

Bad example ($F \perp A$)

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



$$P(F|A=t) \propto P(F,A=t)$$

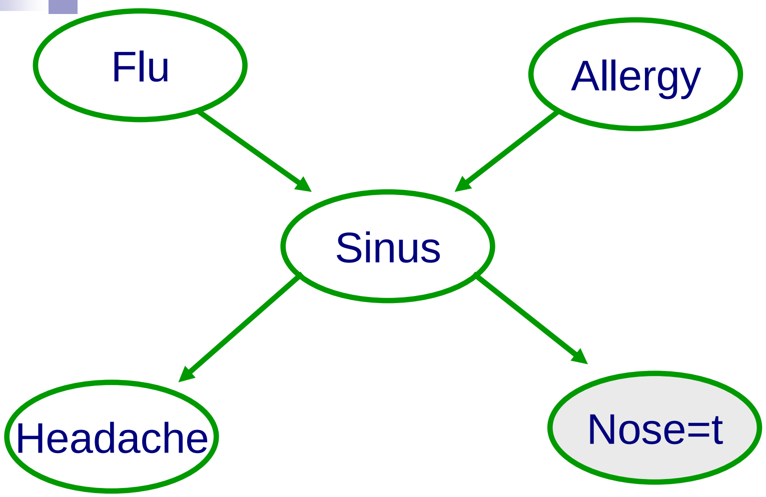
$$P(F,A=t) = \sum_{s} P(F,S=s,A=t)$$

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

$$\left.\begin{array}{l} P(F=t,A=t) = 0.3 \\ P(F=f,A=t) = 0.5 \end{array}\right\} \quad P(F=t|A=t) = \frac{0.3}{0.3+0.5}$$

Probabilistic inference example

16 sums
64 multiplies



$$P(F, N=t) = \sum_{\underbrace{a,s,h}_{2^3}} P(F, A, S, H, N=t)$$

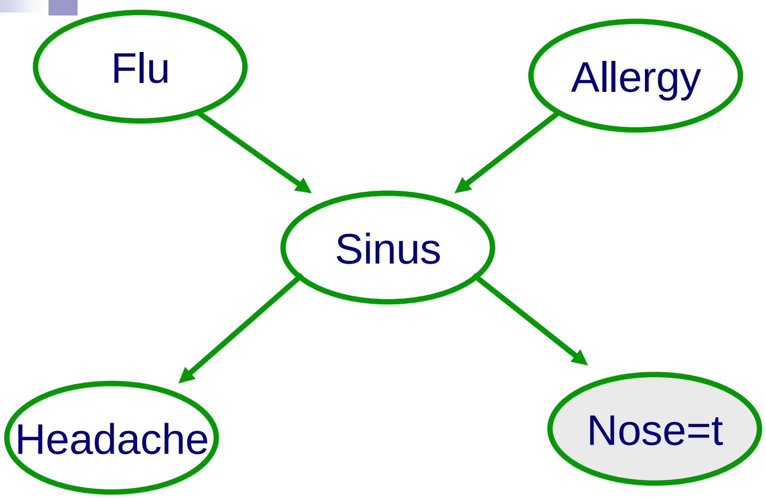
$$= \sum_{a,s,h} P(F)\,P(A)\,P(S|F,A)\,P(H|S)\,P(N=t|S)$$

in general

$$P(X_1, X_n=t) = \sum_{\underbrace{x_2 \dots x_{n-1}}_{k^{n-2}}} P(X_1, \dots, X_{n-1}, X_n=t)$$

Inference seems exponential in number of variables!

Fast probabilistic inference example – Variable elimination



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

$$= \sum_{a,s,h} P(F) \cdot P(A=a) \cdot P(S=s \mid A=a, F) \cdot P(H=h \mid S=s) \cdot P(N=t \mid S=s)$$

$$= \sum_{a,s} P(F)\, P(a) .\, P(s \mid a,F)\, P(N=t \mid s) \cancel{\sum_h P(h \mid s)} \rightarrow 1$$

no work

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

$\leftarrow$ 4 (1 sum, 2 multi

$$= P(F) . \underbrace{\sum_a P(a)\, g_1(F,a)}_{g_2(F)} = \underbrace{P(F) . g_2(F)}_{}$$

$g_2(F) \leftarrow$ 2 (1 sums, 1 multi) 2 mult

total:

6 sums

14 multiply

(Potential for) Exponential reduction in computation!

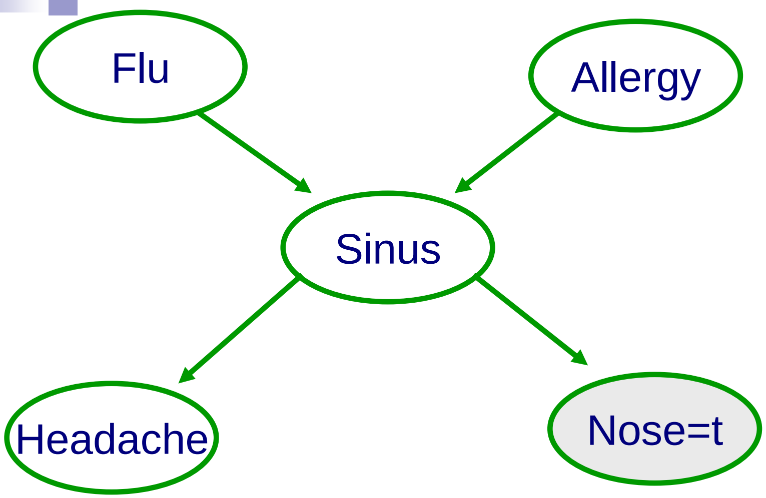
Understanding variable elimination – Exploiting distributivity

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

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

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

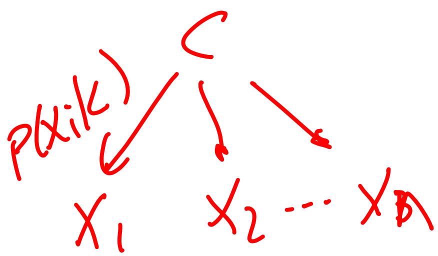
Understanding variable elimination – Order can make a HUGE difference



$$P(F \mid N=t) = \sum_{a,h} P(F)\,P(a) \underbrace{\sum_{s} P(s \mid F,a)\,P(h \mid s)\,P(N=t \mid s)}_{g_1(F,a,h)}$$

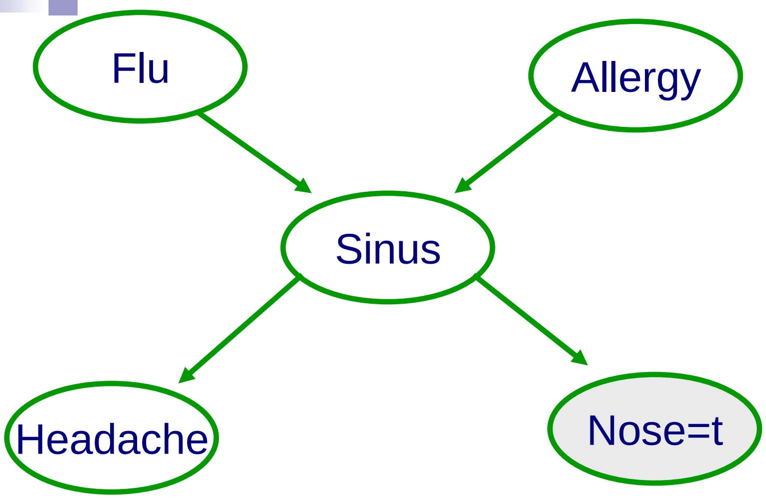
$$8\begin{pmatrix}\cancel{4}\ \text{mult}\\ 1\ \text{sum}\end{pmatrix}$$

Naive Bayes



$P(X_n)$ sum out C first

Understanding variable elimination – Intermediate results



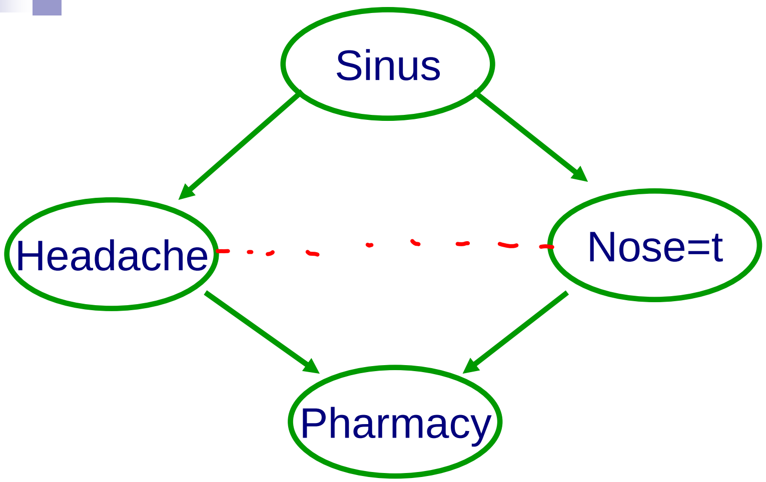
$$P(F, N=t) = \sum_{a,h} P(F), P(a) \sum_{s} P(s|F,a)\, P(h|s)\, P(N=t|s)$$

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

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

Intermediate results are probability distributions

Understanding variable elimination – Another example

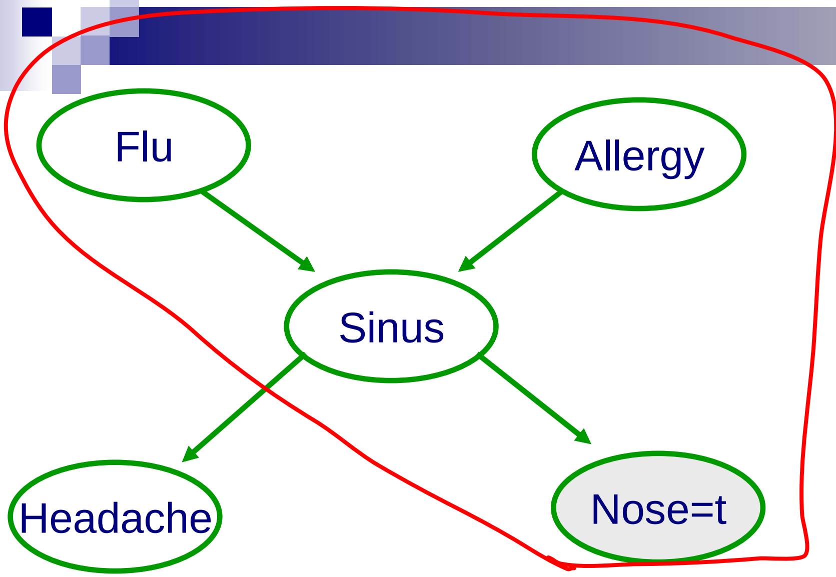


$$P(P, N=t) = \sum_{s,h} P(s) \cdot P(h|s) P(N=t|s) P(P|h,N=t)$$

$$= \sum_{h} P(P|h,N=t) \underbrace{\sum_{s} P(s) P(h|s) P(N=t|s)}_{g_1(h,\,N=t)}$$

$$g_1(H,N=t) = P(H, N=t)$$

Pruning irrelevant variables



$P(F|N=t)$

Prune all non-ancestors of query variables

More generally: Prune all nodes not on active trail between evidence and query vars

Variable elimination algorithm

- Given a BN and a query $P(X|e) \propto P(X,e)$
- Instantiate evidence **e** setting $N=t$
- Prune non-active vars for {X,**e**} (optional step) **IMPORTANT!!!**
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- Initial *factors* $\{f_1,\ldots,f_n\}$: $f_i = P(X_i|\mathbf{Pa}_{X_i})$ (CPT for X_i)
- For i = 1 to n, If $X_i \notin \{X,\mathbf{E}\}$; X_i is ith var in elim. order
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

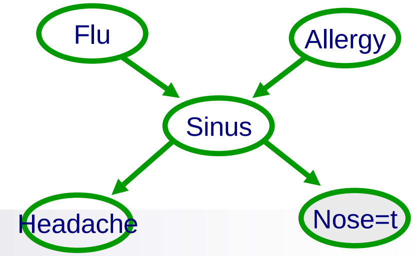
$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

remove $f_1 \ldots f_k$ from bag of factors

add g to bag

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

Operations on factors



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

Scope of f_j to be vars that f_j depends on

Multiplication:

$$h = \prod_{j=1}^{k} f_j \quad ; \quad \underbrace{Scope[h]}_{Y} = \bigcup_{j=1}^{k} \underbrace{Scope[f_j]}_{Y_j}$$

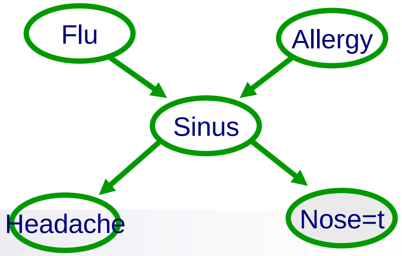
$$\forall y \quad h(y) = \prod_{j=1}^{k} f_j(y_j) \quad ,$$

where y is assignment to Y
y_j subset of y corresp. to Y_j

$C = |Val(Y)|$ ops: $C\,(K-1)$ multiplies

exponential in $|Y|$

Operations on factors



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

Marginalization:

$$g = \sum_{X_i} h \quad ; \quad Y = \text{Scope}(h)$$

$$\text{Scope}(g) = Y - \{X_i\}$$

$$\forall u \in Val(Y - X_i): \quad g(u) = \sum_{x_i} h(u, x_i)$$

$$\text{ops}: \quad \leq C \cdot (Val(X_i) - 1) \text{ sums}$$

Complexity of VE – First analysis

- Number of multiplications:

C_i factor g_i

K_i # factors depend on X_i

$\sum_i C_i K_i$ $\quad \forall i\; C_i \leq C$

$\leq C \underbrace{\sum_i K_i}_{\leq 2n} \leq C \cdot 2n$

- Number of additions: $O(C \cdot n)$

assumption n — num. vars

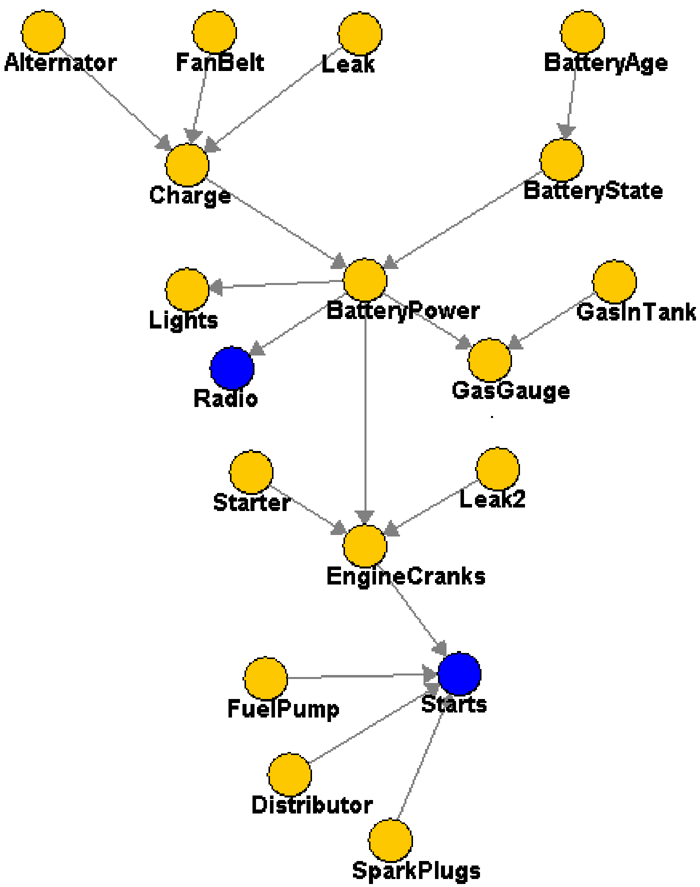
tractable

problem $C \leq$ exp. # vars in g's

Complexity of variable elimination – (Poly)-tree graphs

one trail between any x_i, x_j

Variable elimination order:
Start from "leaves" inwards:
- Start from skeleton!
- Choose a "root", any node
- Find topological order for root
- Eliminate variables in reverse order



$Y_1 \ldots Y_n$

X

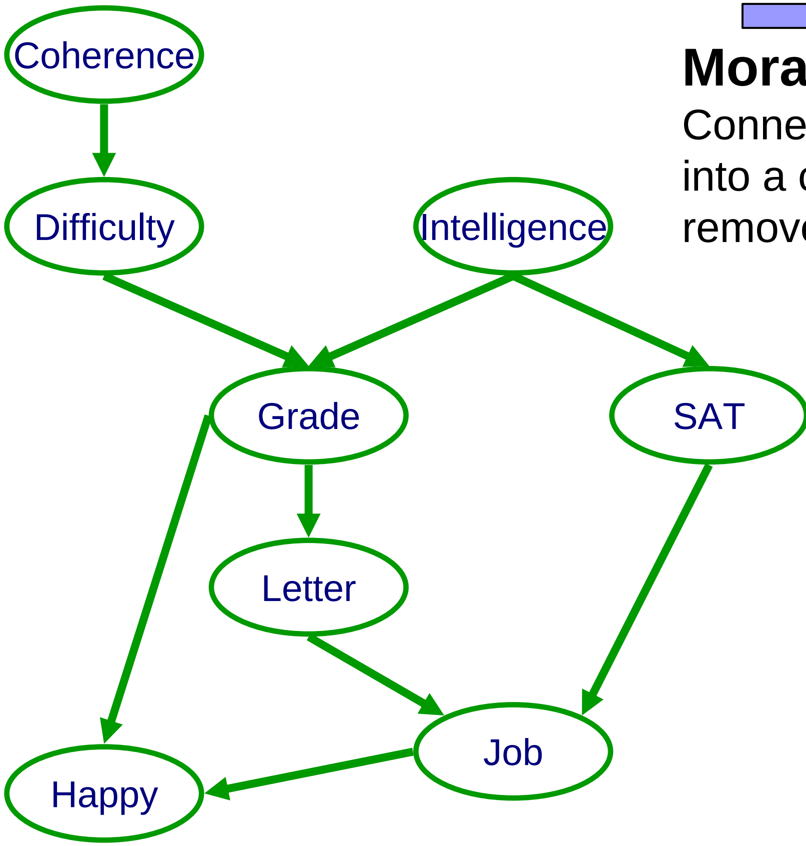
size CPT $P(X \mid Y_1, \ldots Y_n)$?

exponential

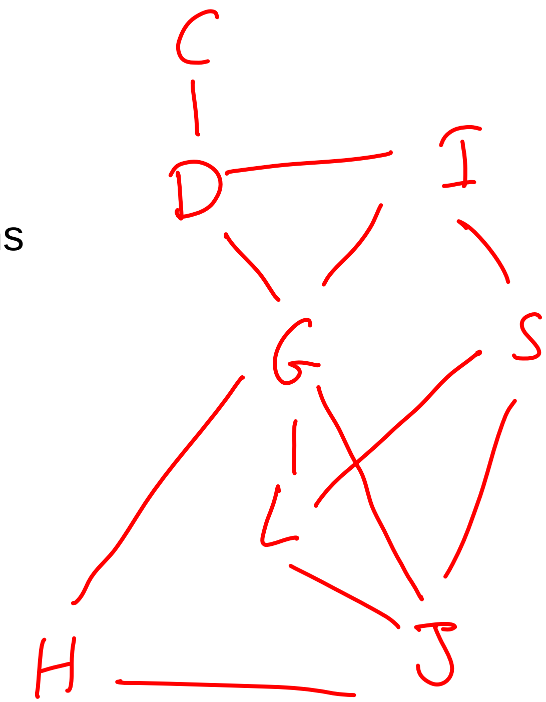
Linear in CPT sizes!!! (versus exponential)

Complexity of variable elimination – Graphs with loops

many trails!



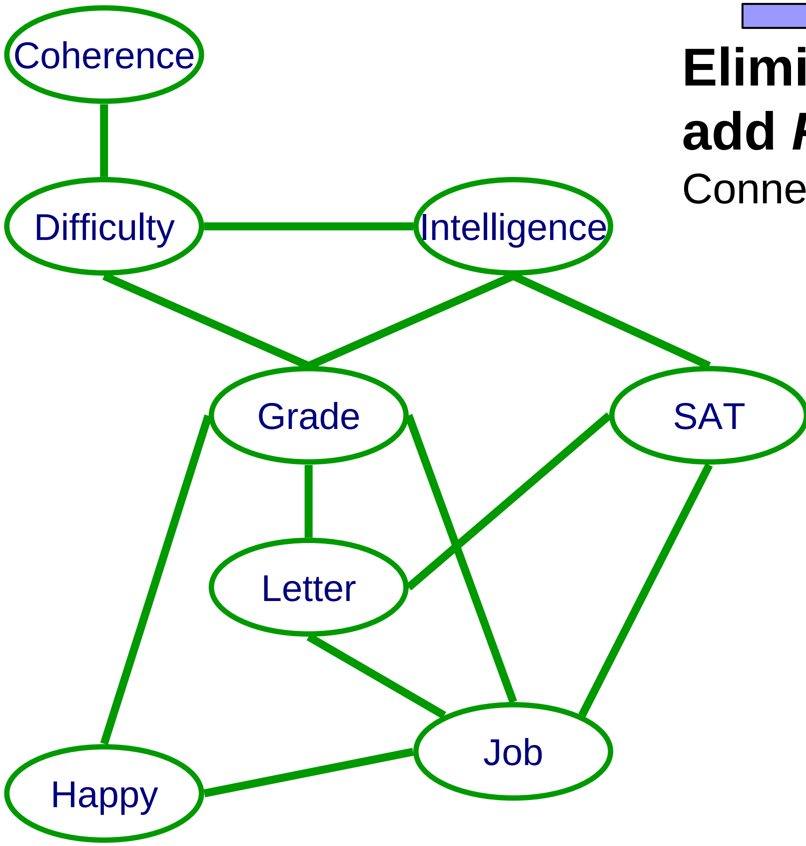
Moralize graph:
Connect parents into a clique and remove edge directions



any X_i, X_j that appear in same initial factor of VE connected in moral graph

Connect nodes that appear together in an initial factor

Eliminating a node – Fill edges

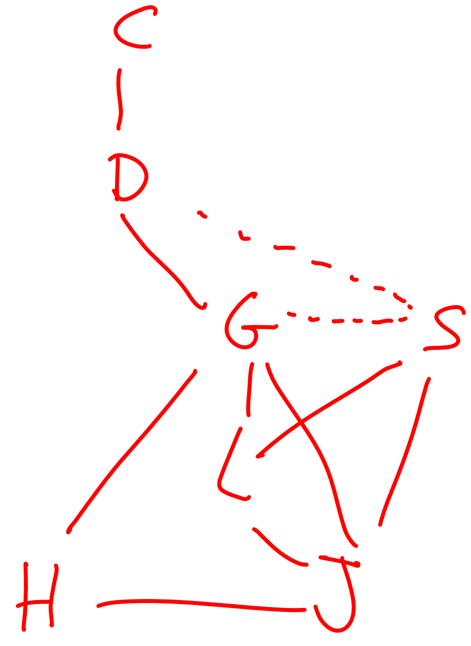


Eliminate variable add *Fill Edges*:
Connect neighbors

eliminate I first

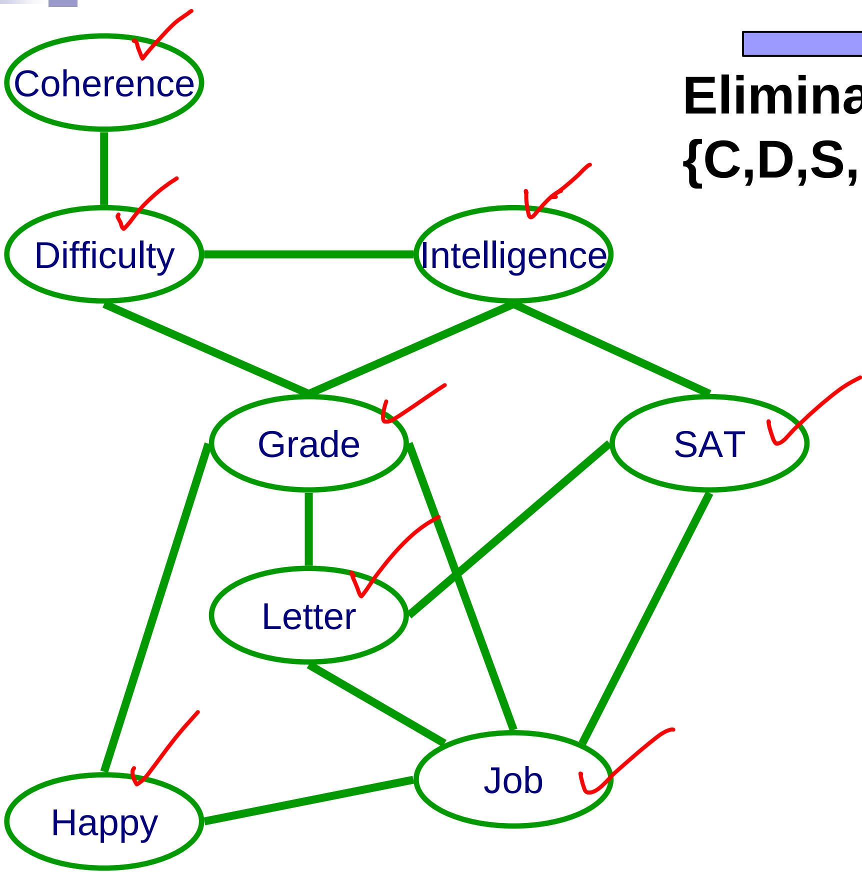
generate

$g_1(D,G,S)$

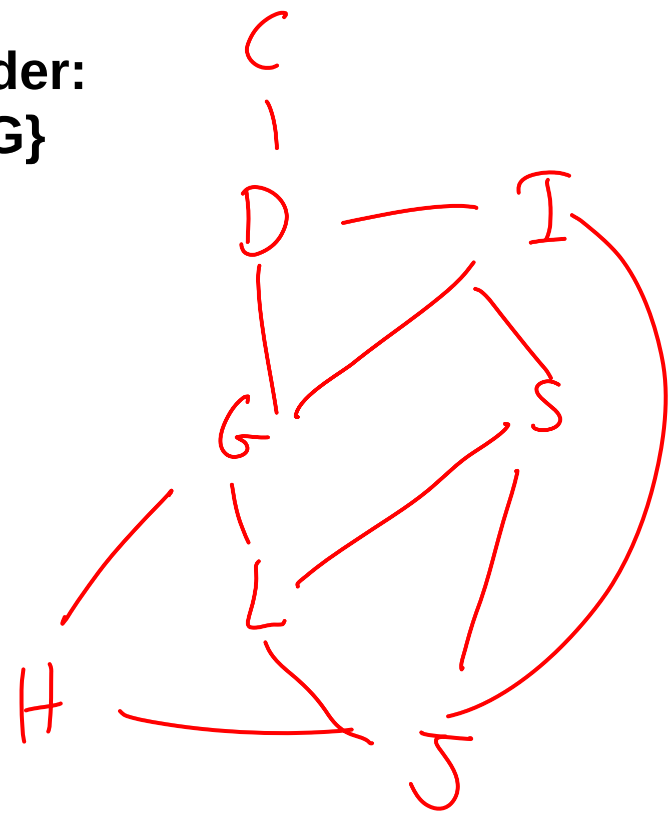


Induced graph

The **induced graph $I_{F\prec}$** for elimination order $\prec$ has an edge $X_i - X_j$ if X_i and X_j appear together in a factor generated by VE for elimination order $\prec$ on factors F

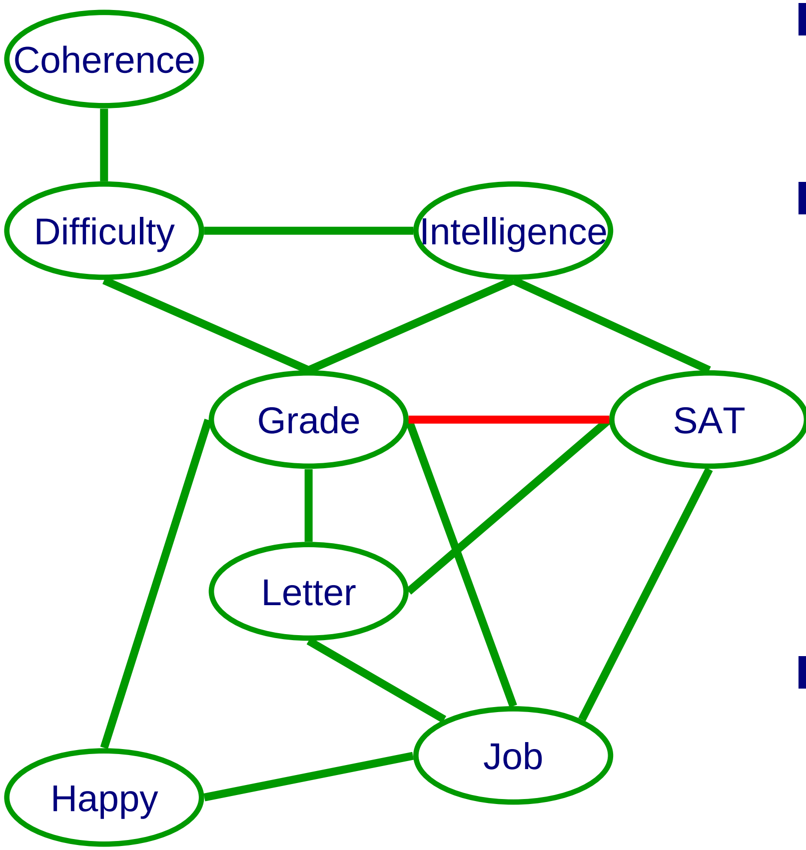


Elimination order: {C,D,S,I,L,H,J,G}



Induced graph and complexity of VE

Read complexity from cliques in induced graph



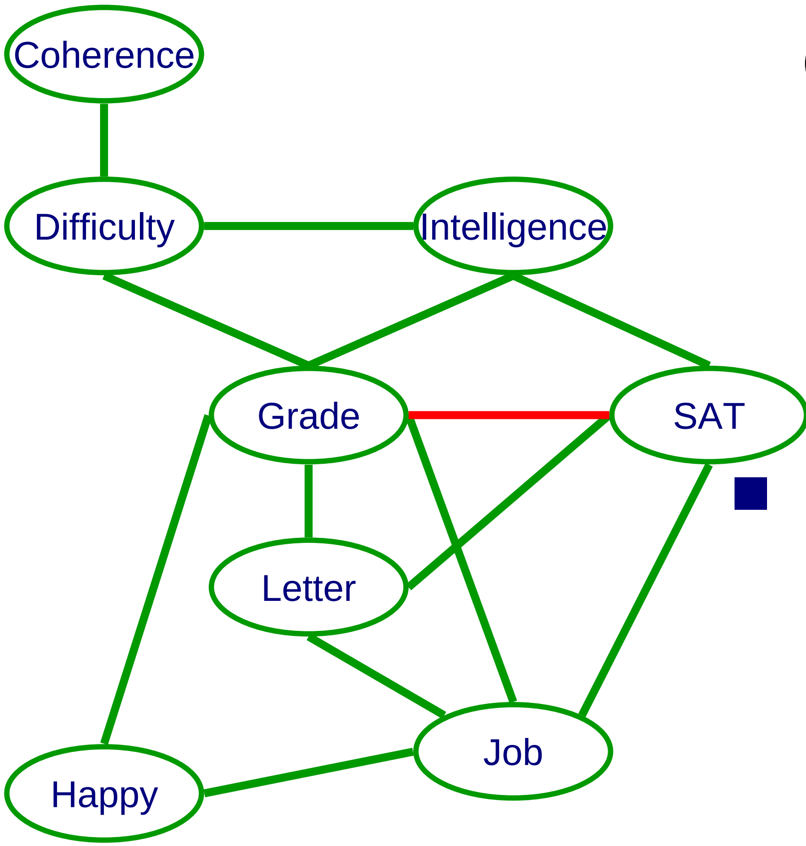
Elimination order: {C,D,I,S,L,H,J,G}

- Structure of induced graph encodes complexity of VE!!!
- **Theorem**:
  - Every factor generated by VE subset of a maximal clique in $I_{F\prec}$
  - For every maximal clique in $I_{F\prec}$ corresponds to a factor generated by VE
- **Induced width** (or treewidth)
  - Size of largest clique in $I_{F\prec}$ minus 1
  - *Minimal induced width* – induced width of best order $\prec$

Example: Large induced-width with small number of parents

Compact representation $\not\Rightarrow$ Easy inference ☹

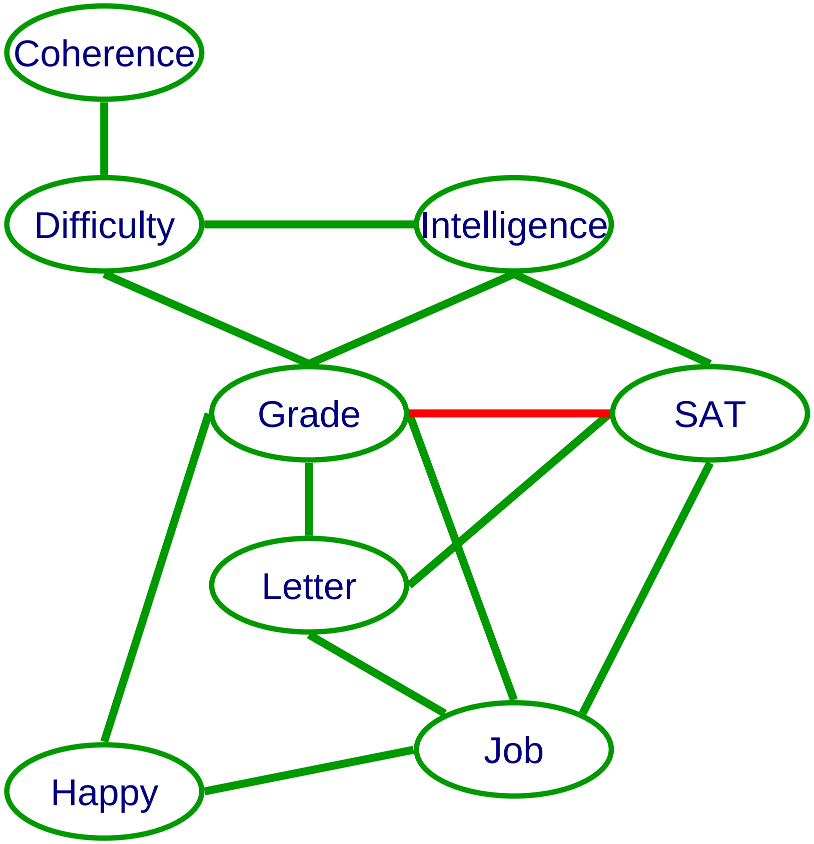
Finding optimal elimination order



Elimination order: {C,D,I,S,L,H,J,G}

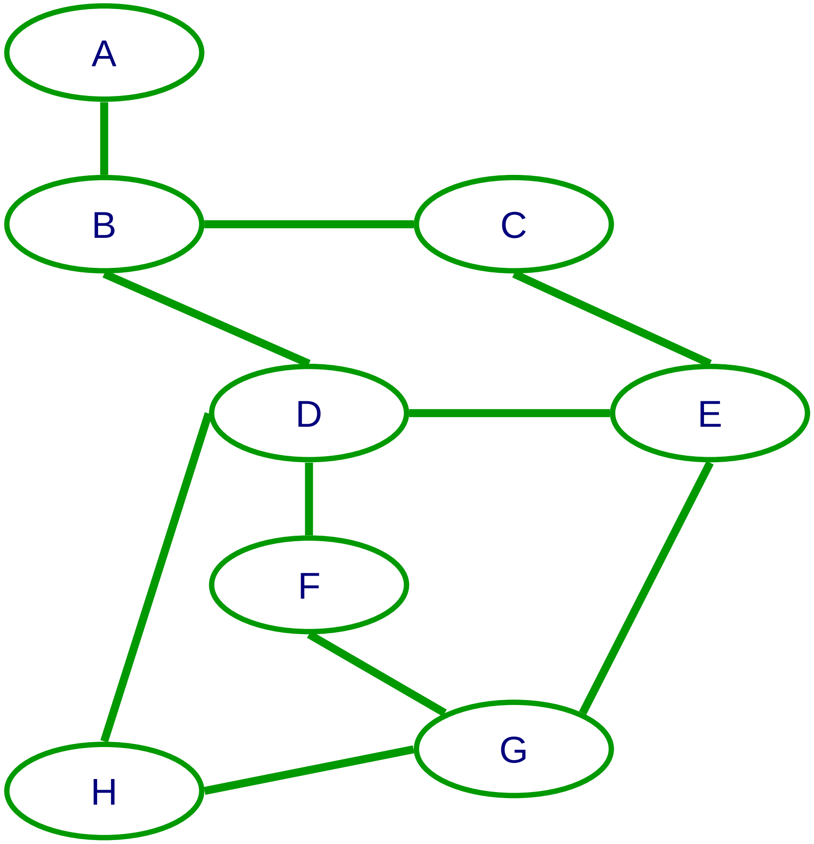
- **Theorem**: Finding best elimination order is NP-complete:
  - Decision problem: Given a graph, determine if there exists an elimination order that achieves induced width $\leq$ K
- **Interpretation**:
  - Hardness of elimination order "orthogonal" to hardness of inference
  - Actually, can find elimination order in time exponential in size of largest clique – same complexity as inference (next week)

Induced graphs and chordal graphs



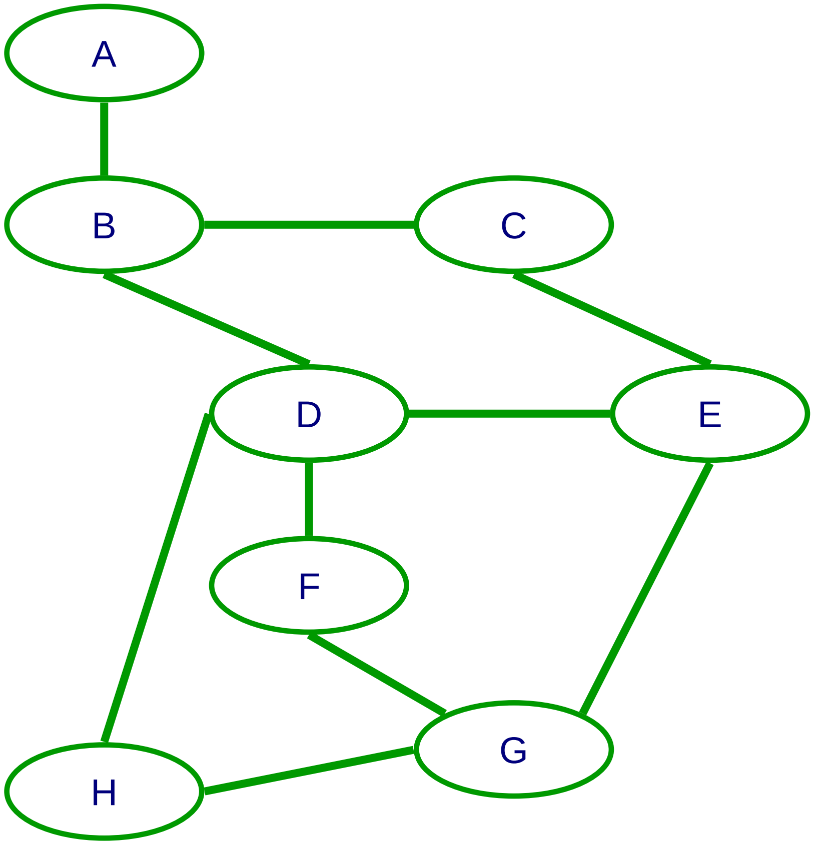
- **Chordal graph**:
  - Every cycle $X_1 - X_2 - \ldots - X_k - X_1$ with $k \geq 3$ has a chord
  - Edge $X_i - X_j$ for non-consecutive i & j
- **Theorem**:
  - Every induced graph is chordal
- "Optimal" elimination order easily obtained for chordal graph

Chordal graphs and triangulation



- **Triangulation**: turning graph into chordal graph
- **Max Cardinality Search**:
  - Simple heuristic
- Initialize unobserved nodes **X** as unmarked
- For k = |**X**| to 1
  - X $\leftarrow$ unmarked var with most marked neighbors
  - $\prec$(X) $\leftarrow$ k
  - Mark X
- **Theorem**: Obtains optimal order for chordal graphs
- Often, not so good in other graphs!

Minimum fill/size/weight heuristics

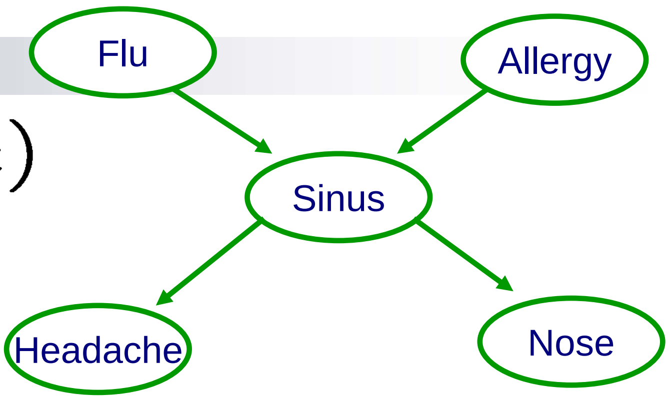


- Many more effective heuristics
  - page 262 of K&F
- **Min (weighted) fill heuristic**
  - Often very effective
- Initialize unobserved nodes **X** as unmarked
- For k = 1 to |**X**|
  - X $\leftarrow$ unmarked var whose elimination adds fewest edges
  - $\prec$(X) $\leftarrow$ k
  - Mark X
  - Add fill edges introduced by eliminating X
- Weighted version:
  - Consider size of factor rather than number of edges

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 approximate inference approaches build on ideas from variable elimination

Most likely explanation (MLE)



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

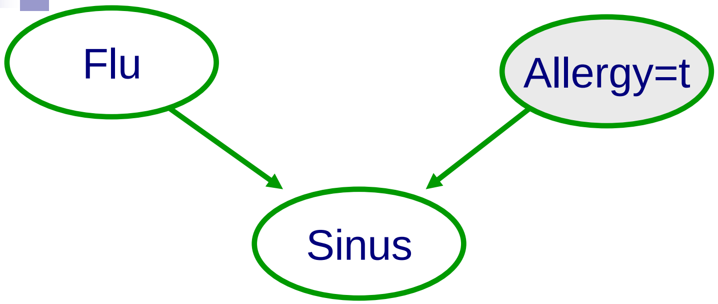
- Using Bayes rule:

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

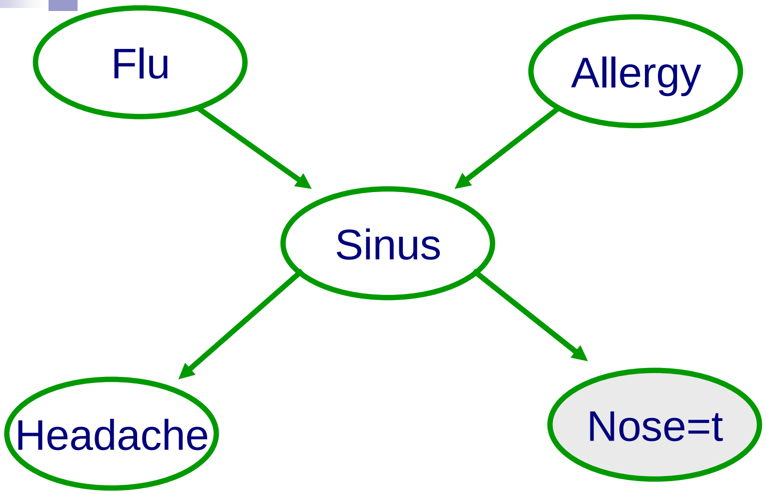
- Normalization irrelevant:

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

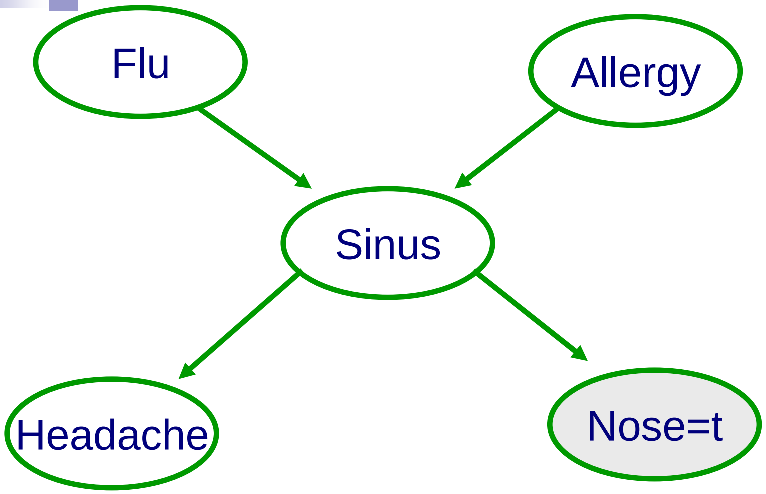
Max-marginalization



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 $\max_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,\mathbf{e})$
- Instantiate evidence $\mathbf{E}=\mathbf{e}$
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \mathbf{E}$
  - 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 $X_i \notin \mathbf{E}$
  - 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:

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

What you need to know

- Variable elimination algorithm
  - Eliminate a variable:
    - Combine factors that include this var into single factor
    - Marginalize var from new factor
  - Cliques in induced graph correspond to factors generated by algorithm
  - Efficient algorithm ("only" exponential in induced-width, not number of variables)
    - If you hear: "Exact inference only efficient in tree graphical models"
    - You say: "No!!! Any graph with low induced width"
    - And then you say: "And even some with very large induced-width" (next week)
- Elimination order is important!
  - NP-complete problem
  - Many good heuristics
- Variable elimination for MLE
  - Only difference between probabilistic inference and MLE is "sum" versus "max"