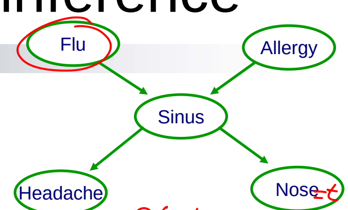
Readings:
K&F: 8.1, 8.2, 8.3, 8.7.1

Variable Elimination

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

October 11th, 2006

General probabilistic inference



$P(F \mid N=t)$

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

- Using def. of cond. prob.:

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

normalization constant

$P(X=t, e) = 0.4$

$P(X=f, e) = 0.1$

- Normalization:

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

normalize

$P(X=t \mid e) = \frac{0.4}{0.5} = 0.8$

Marginalization

I know $P(F,S,N) = P(F)\cdot P(S|F)\cdot P(N|S)$

$P(S=t|N=t) = \frac{.4}{.4+.2} = \frac{2}{3}$

first focus on $P(S=t, N=t) = .4$

$P(S=f, N=t) = .2$

$P(S, N=t)$

computing table for each value of S

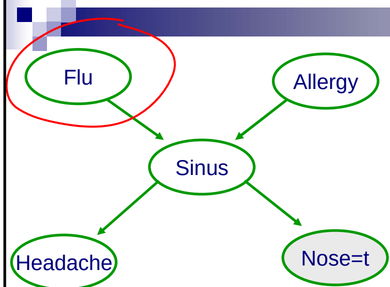
marginalization

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

$$= \sum_{f\in\{t,f\}} P(F=f)\cdot P(S|F=f)\cdot P(N=t|S)$$

$$= P(F=true)\cdot P(S|F=true)\cdot P(N=t|S) + P(F=false)\cdot P(S|F=false)\cdot P(N=t|S)$$

Probabilistic inference example



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

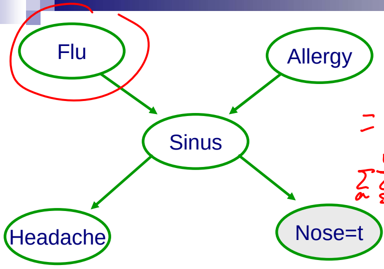
exponential blow-up

2^3

if marginalize out k vars, go through 2^k assignments

Inference seems exponential in number of variables!

Fast probabilistic inference example – Variable elimination



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

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

multiplications:

$\sum_a \sum_s \sum_h \left(\underset{2}{|F|} . \underset{2}{|A|} . \underset{2}{|S|} \cdot 3 = 24 \right.$

$= \sum_{a,s} P(f)\, P(a)\, P(s|f,a) . P(N=t|s) \cancel{\sum_h P(h|s)}$ 1

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

$|F|.|A||S|$

$g_1: 2 \times 2 \times 2 \cdot 1 = 8$

$g_1(f,a) =$

f \ a	t	f
t	..	$\sum_s P(s|f=t,A=t).P(N=t|s)$
f	..	..

$= \sum_a P(F) . P(a)\; g_1(f,a)$

$|A|.|F|$

$g_2: 2 \times 2 . 1 = 4$

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

$P(F).g_2(f)$: $|F|$ $2 \times 1 = 2$

14

(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity

$\underbrace{a}_{g}(\underbrace{b+c}_{g_1}) = ab + ac$

1 multiplication, 1 addition

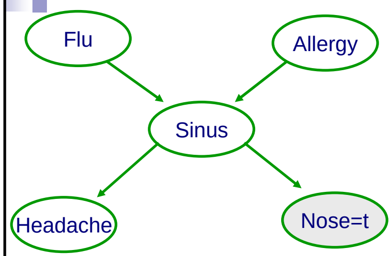
2 mult., 1 add

$P(F, N=t) = P(F) . P(S=t|F) . P(N=t|S=t) +$
$P(F) . P(S=f|F) . P(N=t|S=f)$

$= P(F) . \big(\underbrace{P(S=t|F) . P(N=t|S=t) + P(S=f|F) . P(N=t|S=f)}_{g_1(F)} \big)$

$= P(F) . g_1(F)$

Understanding variable elimination – Order can make a HUGE difference

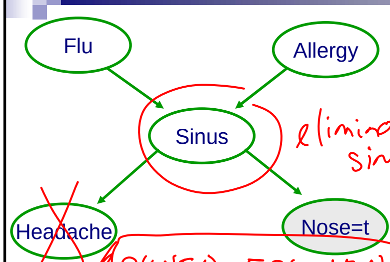


$$P(F, N=t) = \sum_{a,s,h} P(F) \cdot P(a)\, P(s|F,a) \cdot P(h|s) \cdot P(N=t|s)$$

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

now depends on headache (bigger table)

Understanding variable elimination – Intermediate results



eliminate sinus

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

$$= \sum_{a} P(F) \cdot P(a) \underbrace{\sum_{s} P(s|f,a) \cdot P(N=t|s)}_{g_1(f,a)}$$

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

$$g_1(f,a) = \sum_{S} P(s|f,a) \cdot P(N=t|s)$$

$$= P(N=t|f,a)$$

cond. prob. evidence given F, A

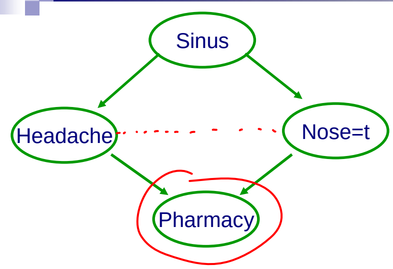
$$P(N,S|F,A) = P(S|F,A) \cdot P(N|S,F,A)$$

|| local markov

$$P(N|S)$$

Intermediate results are (conditional) probability distributions

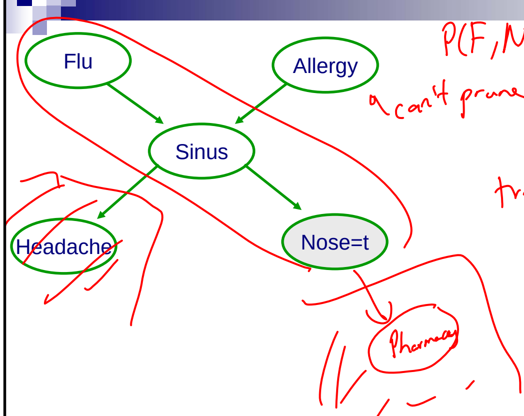
Understanding variable elimination – Another example



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

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

Pruning irrelevant variables



$P(F, N=t)$ ← headache irrelevant

Pharmacy irrelevant

a can't prune

trail between evidence an query

- remove all descendants

Prune all non-ancestors of query variables
More generally: Prune all nodes not on active trail between evidence and query vars

Variable elimination algorithm

$P(F,A \mid N=t, H=f)$
set vars / set of vars / X / e

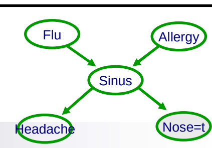
- Given a BN and a query $P(X|\mathbf{e}) \propto P(X,\mathbf{e})$
- Instantiate evidence **e** ← fix value in CPTs — all CPTs that include N, H, set N=t H=f
- Prune non-active vars for {X,**e**} (optional) **IMPORTANT!!!**
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- Initial *factors*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

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

-add g to the bag of factors
-remove $f_1, \ldots f_k$ from bag
marginalize, multiply

  - Variable X_i has been eliminated!
- Normalize $P(X,\mathbf{e})$ to obtain $P(X|\mathbf{e})$

Operations on factors



$f_1(a,b) =$

a\b	t	f
t	.1	.2
f	.3	.4

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

$f_2(b,c) =$

b\c	t	f
t	.01	.02
f	.03	.04

Multiplication:

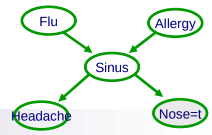
eliminating B

$f(a,b,c) = f_1 . f_2 =$ $f(a\,b\,c) =$

bc\a	t	f
tt	.1x.01	
tf		
ft		.4x.03
ff		

$f(a,b,c,d) = f_1(a\,b) . f_2(b\,c) . f(b,d)$

Operations on factors



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

Marginalization:

$g(a,c) = \sum_b f(a,b,c)$

bc \ a	t	f
tt		
$f(abc) =$ tf		
ft		
ff		

$g(A=t, C=f)$ + ...

Complexity of VE – First analysis

- Number of multiplications:
- Number of additions:

Eliminating var X_i

$g(C_i)$ C_i clique variables

$C_i \subseteq X$

size of g_i: exponential in $|C_i|$

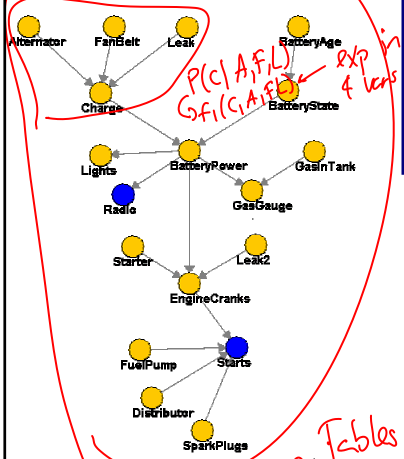
$|Val(C_i)|$, e.g., $k^{|C_i|}$ if all X_i $|Val(X_i)| = k$

generate n g_i's along the way

$O(n \cdot |Val[C_i]|)$

Complexity of variable elimination – (Poly)-tree graphs

Poly-tree



P(C|A,F,L) ← exp in 4 vars

$G f_1(C,A,F,L)$

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

running time linear in # vars

exponential in # parents

a. Tables

Linear in CPT sizes!!! (versus exponential)

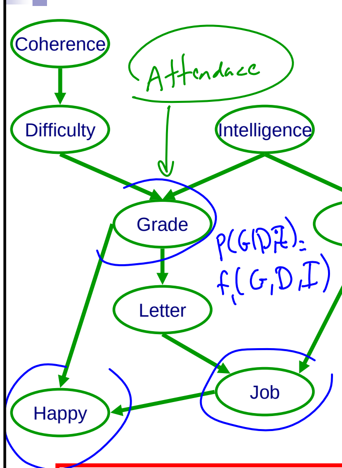
What you need to know about inference thus far

- Types of queries
  - probabilistic inference
  - most probable explanation (MPE)
  - maximum a posteriori (MAP)
    - MPE and MAP are truly different (don't give the same answer)
- Hardness of inference
  - Exact and approximate inference are NP-hard
  - MPE is NP-complete
  - MAP is much harder (NP^{PP}-complete)
- Variable elimination algorithm
  - Eliminate a variable:
    - Combine factors that include this var into single factor
    - Marginalize var from new factor
  - 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 with context-specific independence)
- Elimination order is important!
  - NP-complete problem
  - Many good heuristics

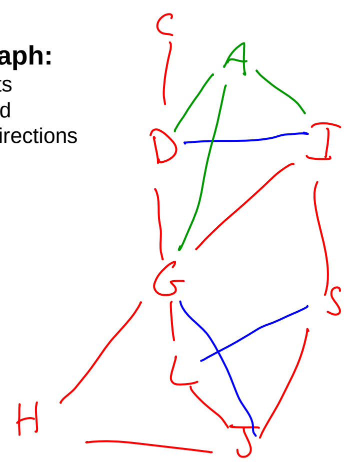
Announcements

- Recitation tomorrow:
  - Khalid on Variable Elimination
- Recitation on advanced topic:
  - Carlos on Context-Specific Independence
  - On Monday Oct 16, 5:30-7:00pm in Wean Hall 4615A

Complexity of variable elimination – Graphs with loops

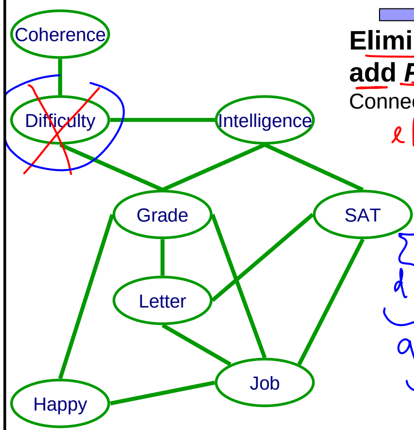


Moralize graph:
Connect parents into a clique and remove edge directions



Connect nodes that appear together in an initial factor

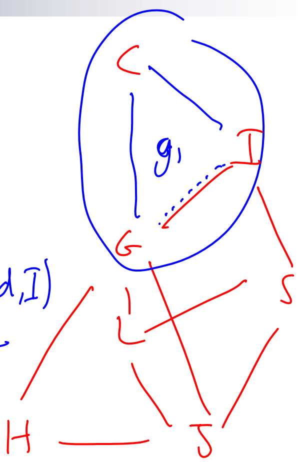
Eliminating a node – Fill edges



Eliminate variable add *Fill Edges*:
Connect neighbors

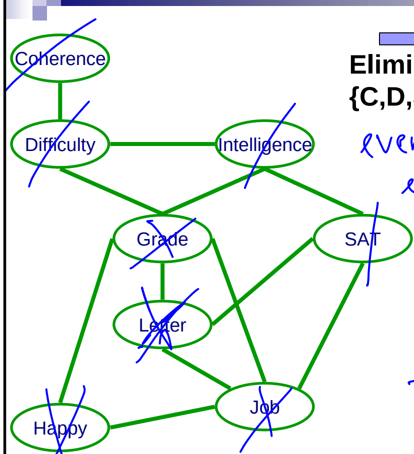
eliminate D :

$$\underbrace{\sum_d P(d|c) \cdot P(G|d,I)}_{g_1(C,G,I)}$$



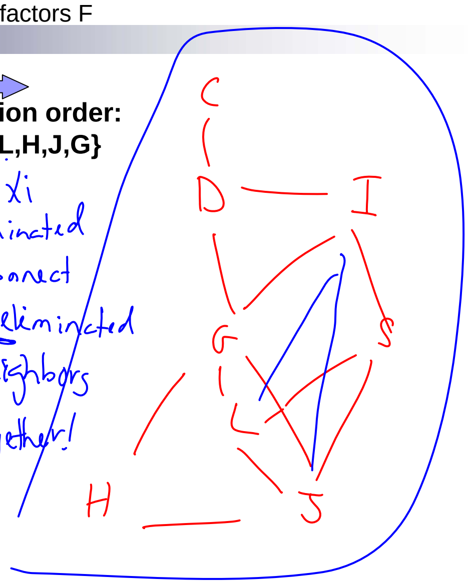
Induced graph

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

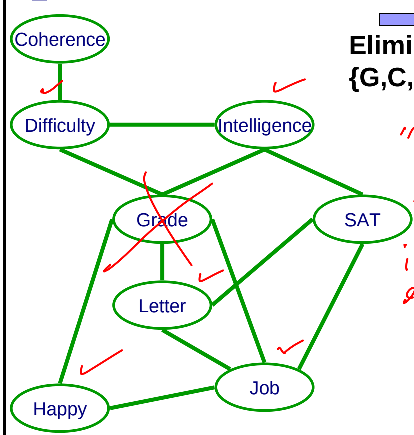


Elimination order: {C,D,S,I,L,H,J,G}

every X_i eliminated connect uneliminated neighbors together!

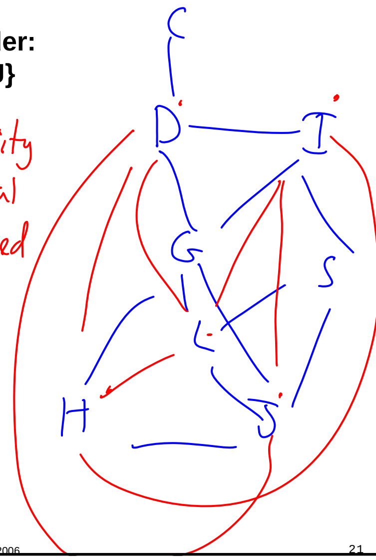


Different elimination order can lead to different induced graph



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

"complexity exponential in added edges"

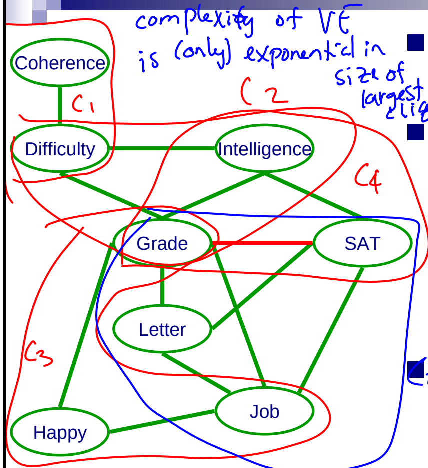


Induced graph and complexity of VE

Read complexity from cliques in induced graph

clique: set of nodes fully connected together
maximal clique: not subset of other cliques

complexity of VE is (only) exponential in size of largest clique



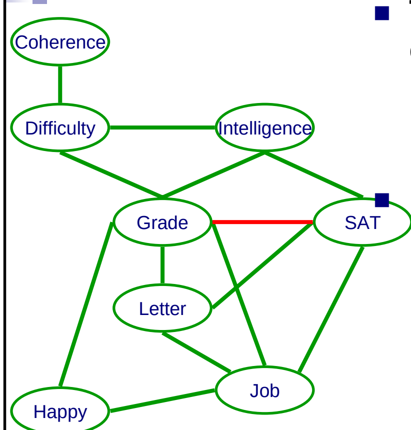
Elimination order:
{C,D,I,S,L,H,J,G}

- Structure of induced graph encodes complexity of VE!!!
- **Theorem**:
  - Every factor generated by VE subset of a maximal clique in $I_{F\prec}$ with ordering $\prec$
  - For every maximal clique in $I_{F\prec}$ corresponds to a factor generated by VE with ordering $\prec$
- C5 **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 ☹

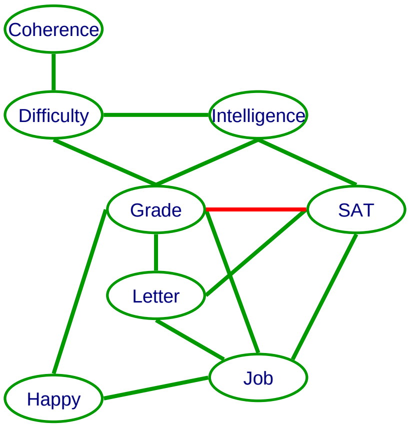
Finding optimal elimination order



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

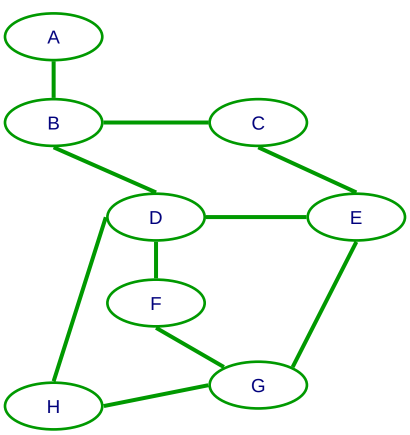
- **Theorem**: Finding best elimination order is NP-complete:
  - Decision problem: Given a graph, determine if there exists an elimination order that achieves induced width $\leq$ K
- **Interpretation**:
  - Hardness of finding elimination order in addition to hardness of inference
  - Actually, can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

Induced graphs and chordal graphs



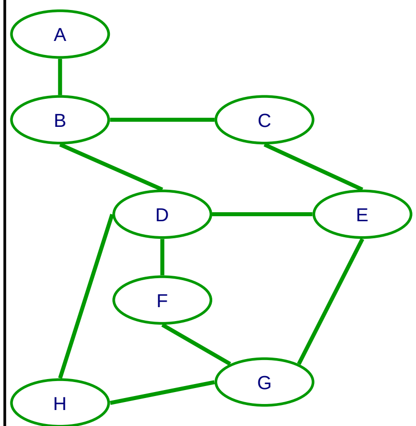
- **Chordal graph**:
  - Every cycle $X_1 - X_2 - \ldots - X_k - X_1$ with $k \geq 3$ has a chord
  - Edge $X_i - X_j$ for non-consecutive i & j
- **Theorem**:
  - Every induced graph is chordal
- "Optimal" elimination order easily obtained for chordal graph

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

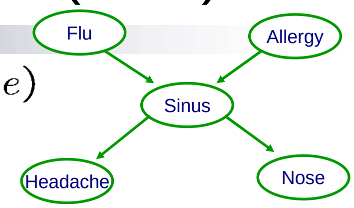


- Many more effective heuristics
  - see reading
- **Min (weighted) fill heuristic**
  - Often very effective
- Initialize unobserved nodes **X** as unmarked
- For k = 1 to |**X**|
  - $X \leftarrow$ unmarked var whose elimination adds fewest edges
  - $\prec(X) \leftarrow k$
  - Mark X
  - Add fill 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: $\operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n \mid e)$

- Using defn of conditional probs:

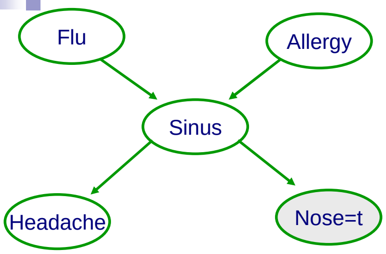
$$\operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n \mid e) = \operatorname*{argmax}_{x_1,\ldots,x_n} \frac{P(x_1,\ldots,x_n,e)}{P(e)}$$

- Normalization irrelevant:

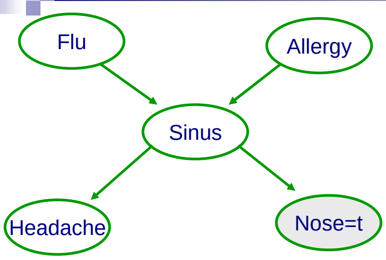
$$\operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n \mid e) = \operatorname*{argmax}_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,e)$$

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 $\max_{x_1,\ldots,x_n} P(x_1,\ldots,x_n,\mathbf{e})$
- Instantiate evidence **E**=**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:

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

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" (special recitation)
- 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"