Readings:
K&F: 8.1, 8.2, 8.3, 8.4

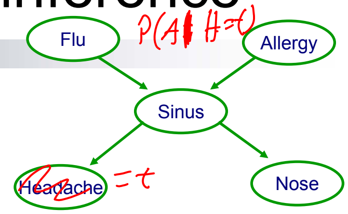
Variable Elimination

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University
October 15th, 2008

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

General probabilistic inference



$P(A \mid H=t)$ $=t$

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

- Using def. of cond. prob.:

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

: compute $\forall x$ $\Rightarrow P(X=x, E=e)$

- Normalization:

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

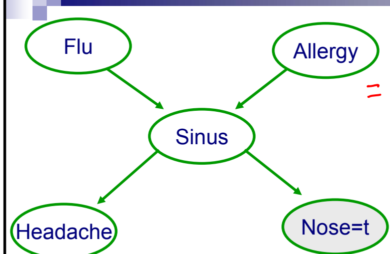
normalize

$P(A=t, H=t) = 0.2$

$P(A=f, H=t) = 0.1$

$P(A=t \mid H=t) = 2/3$

Probabilistic inference example



$P(A \mid N=t) \propto P(A, N=t)$ ($A=t$)

$= \sum_f \sum_s \sum_h P(A, f, s, h, N=t)$

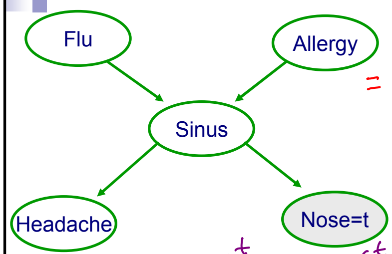
$= \underbrace{\sum_f \sum_s \sum_h}_{8 \text{ terms}} \underbrace{P(f)\, P(A)\, P(s \mid f, A)\, P(h \mid s)\, P(N=t \mid s)}_{4 \text{ multiplies}}$

$2^3 \leftarrow$ # eliminated vars   32 multiplies

Inference seems exponential in number of variables!

Fast probabilistic inference example – Variable elimination

(Potential for) Exponential reduction in computation!



$$P(\overset{=t}{A}|N=t) \propto P(A, N=t)$$

$$= \sum_f \sum_s \sum_h P(\overset{=t}{A}, f, s, h, N=t)$$

Grand total
5 mult.

$$= \sum_f \sum_s \sum_h P(f)\, P(\overset{=t}{A})\, P(s|f,\overset{=t}{A})\, P(h|s)\, P(N=t|s)$$

$$= \sum_f \sum_s P(f)\, P(\overset{=t}{A})\, P(s|f,\overset{=t}{A})\, P(N=t|s) \underbrace{\sum_h P(h|s)}_{1}$$

← if you are smart, no work

$$= \sum_f P(f)\, P(\overset{=t}{A}) \underbrace{\sum_s P(s|f,\overset{=t}{A})\, P(N=t|s)}_{g_1(f,A)}$$

2 multiplies

1 mult.

$$= P(\overset{=t}{A}) \underbrace{\sum_f P(f)\, g_1(f,\overset{=t}{A})}_{g_2(A)} = P(\overset{=t}{A})\, g_2(\overset{=t}{A}) = P(\overset{=t}{A}, N=t)$$

← 2 mult.

Understanding variable elimination – Exploiting distributivity

$$a(b+c) = ab + ac$$

commutativity $ab = ba$

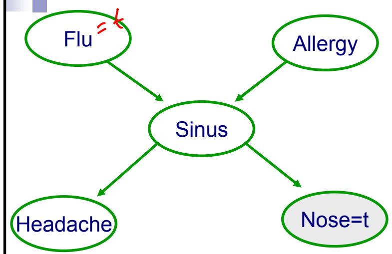
$$P(F=t, N=t) = \sum_s P(F=t) \cdot P(s|F=t) \cdot P(N=t|s)$$

$$= P(F=t) \sum_s P(s|F=t)\, P(N=t|s)$$

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

$$= P(F=t) \Big[\underbrace{P(S=t|F=t)\, P(N=t|S=t) + P(S=f|F=t)\, P(N=t|S=f)}_{\sum_s P(s|F=t)\, P(N=t|s)} \Big]$$

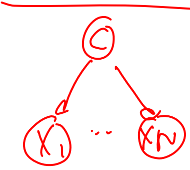
Understanding variable elimination – Order can make a HUGE difference



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

$\sum_h \sum_a \sum_s P(F=t)\, P(a)\, P(s|F=t,a)\, P(h|s) \cdot P(N=t|s)$

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



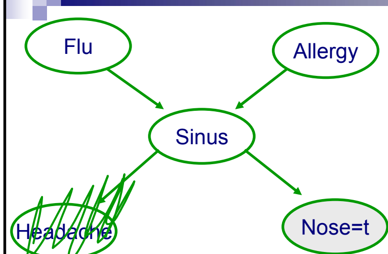
eliminate C first

$\underbrace{\sum_c P(c) \prod_i P(x_i|c)}_{g_1(x_1,\dots,x_n)} \leftarrow$ exponentially large

eliminate X_1

$P(c) \prod_{i \neq 1} P(x_i|c) \sum_{x_1} P(x_1|c)$

Understanding variable elimination – Intermediate results



$P(F, N=t) =$

$\sum_a \sum_s P(F)\, P(a)\, P(s|F,a)\, P(N=t|s)$

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

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

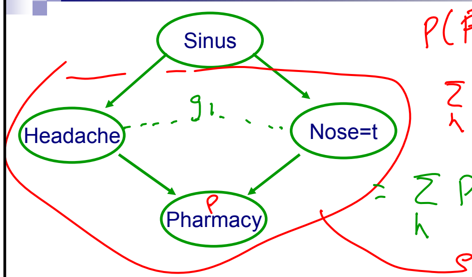
chain rule

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

$N \perp \{F, A\} | S$

$\boxed{\text{Intermediate results are probability distributions}}$

Understanding variable elimination – Another example



$P(P \mid N=t) \propto P(P, N=t)$

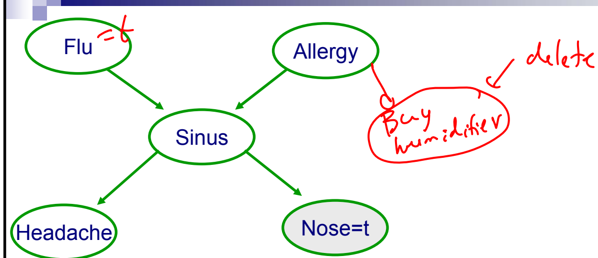
$= \sum_h \sum_s \underbrace{P(s)\, P(N=t \mid s)\, P(h \mid s)}\, P(P \mid h, N=t)$

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

As I eliminate vars ⇒ creating a graphical model for remaining variables ⇒ create/add extra terms ⇒ more edges

True i.p. of P ⊇ idep. of original ⊇ model after vars eliminated
I-map

Pruning irrelevant variables



when summed out, got $g_1 = 1$
could just delete H from model before doing VE

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

Variable elimination algorithm

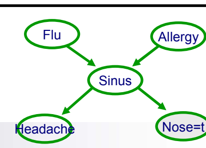
F N=t

- Given a BN and a query $P(X|e) \propto P(X,e)$
- Instantiate evidence **e** , N=t
- Prune non-active vars for {X,**e**} ← option – $\boxed{\text{IMPORTANT}}$ (N,F,A)
- 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) $f_i(X_i, Pa_{X_i})$
- For i = 1 to n, If $X_i \notin \{X,\mathbf{E}\}$ ← must be eliminated
  - 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! ← add g to set of factors
- Normalize P(X,**e**) to obtain P(X|**e**)

Operations on factors



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

$X_i = B$

Multiplication:

$f_1(A,B)$

A \ B	t	f
t	.5	.6
f	.7	.8

$f_2(B,C)$

B \ C	t	f
t	0.5	0.2
f	0.3	0.4

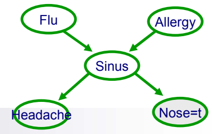
$h(A,B,C) = f_1(A,B) \cdot f_2(B,C)$

A \ BC	tt	tf	ft	ff
t				
f				

A=t, B=f, C=f

$0.6 \times 0.3 = 0.18$

Operations on factors



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

Marginalization:

$$g(A{=}t, C{=}f) = \sum_b h(A, b, C)$$

a t c

h:

A \ B	tt	tf	ft	ff
t		x		x
f				

Complexity of VE – First analysis

- Number of multiplications:

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

each f_j depends on C_j

each var has d assignments

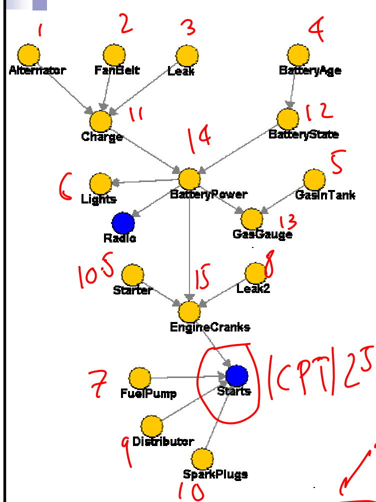
$$h\Big(\underbrace{\bigcup_j C_j}_{\text{size } k}\Big) \leftarrow \text{table has } d^k \text{ elements}$$

each requires m multiplies

exponential in # of vars in intermediate factors

- Number of additions:

Complexity of variable elimination – (Poly)-tree graphs



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

does not create factors any bigger than original CPTs

in trees, solve inference in linear time

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

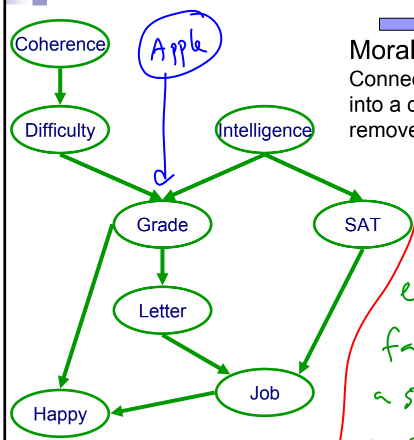
Announcements

- Recitation tomorrow
  - Be there!!
- Homework 3 out later today

What's next

- Thus far: Variable elimination
  - (Often) Efficient algorithm for inference in graphical models
- Next: Understanding complexity of variable elimination
  - Will lead to cool junction tree algorithm later

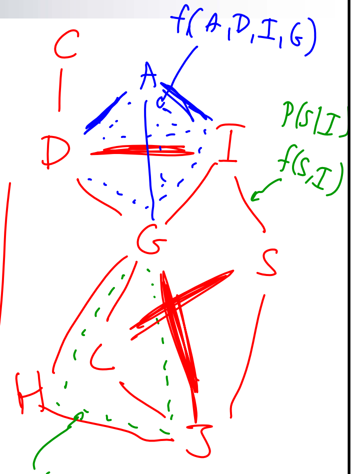
Complexity of variable elimination – Graphs with loops



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

initial factors
$f_i = P(X_i \mid Pa_{X_i})$

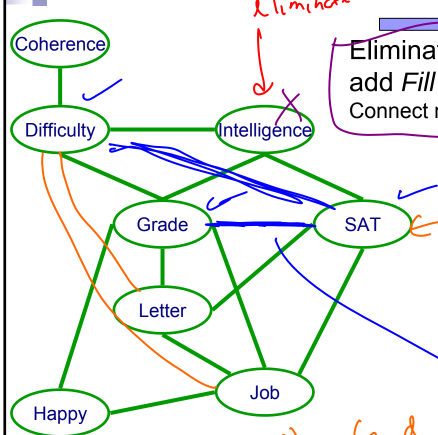
every initial factor f_i is a subset of a maximal clique in moralized graph



Connect nodes that appear together in an initial factor

Eliminating a node – Fill edges

eliminate



Eliminate variable
add *Fill Edges*:
Connect neighbors

$\underbrace{\sum_i P(i)\, \overbrace{P(g \mid d,i)}^{f_1(G,D,I)}\, p(s \mid i)}_{g_1(g,d,s)}$

what if I eliminate SAT next

not a subset of a maximal clique
⇓
add fill edges

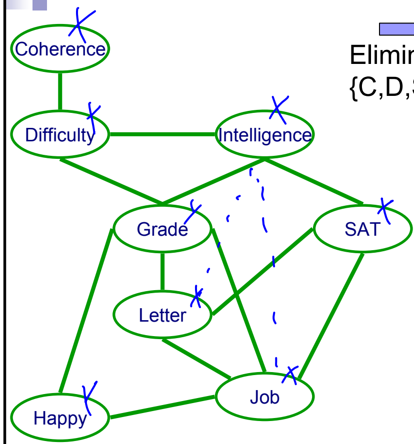
property traces

$\underbrace{\sum_s P(J \mid L,S)\, g_1(g,d,s)}_{g_2(J,L,d,G)}$ ⇐ connect D–L, J–D

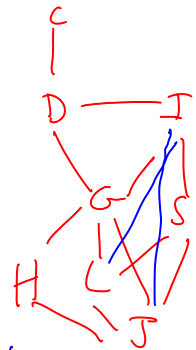
Induced graph

$\lhd \equiv \prec$

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



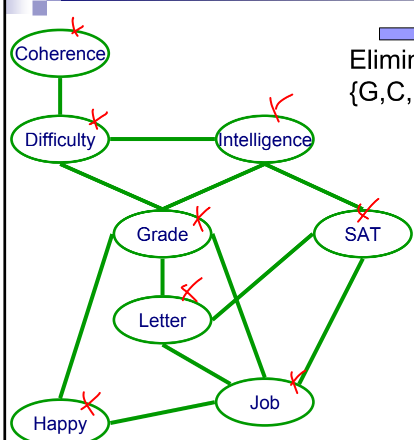
Elimination order:
{C,D,S,I,L,H,J,G}



each VE factor
subset of maximal clique

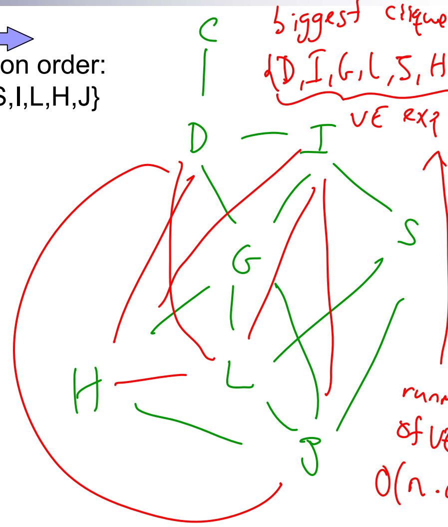
largest clique: $\{\underbrace{G, I, L, J}_{4}\} \Rightarrow$ VE exp. in 4

Different elimination order can lead to different induced graph



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

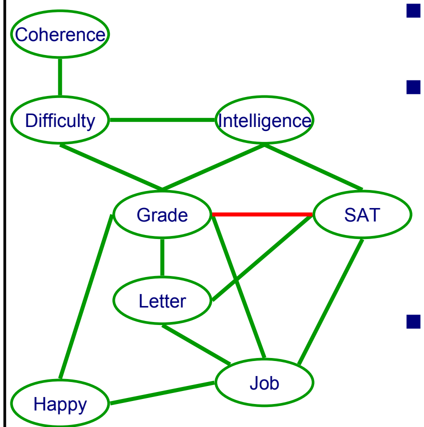
biggest clique:
$\underbrace{\{D, I, G, L, S, H\}}$ VE exp in 6



running of VE
$O(n \cdot d^6)$

Induced graph and complexity of VE

Read complexity from cliques in induced graph



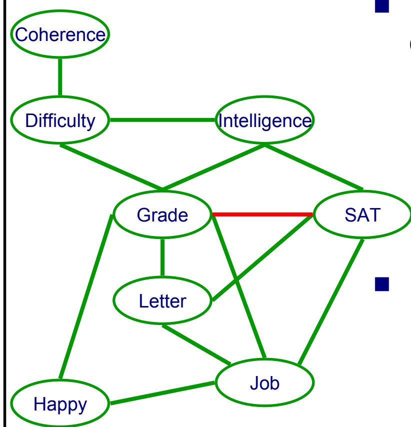
Elimination order:
{C,D,I,S,L,H,J,G}

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

Example: Large induced-width with small number of parents

Compact representation $\nRightarrow$ Easy inference ☹

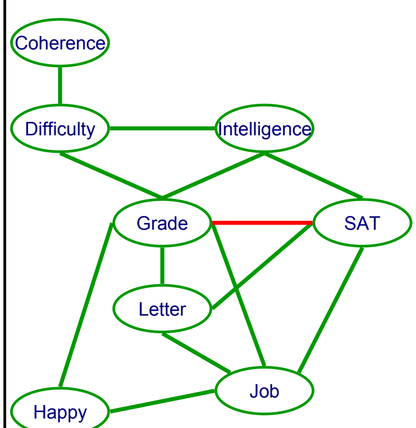
Finding optimal elimination order



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

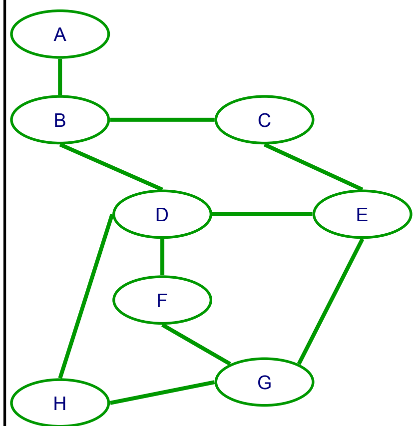
- **Theorem**: Finding best elimination order is NP-complete:
  - Decision problem: Given a graph, determine if there exists an elimination order that achieves induced width ≤ K
- **Interpretation**:
  - Hardness of finding elimination order in addition to hardness of inference
  - Actually, can find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

Induced graphs and chordal graphs



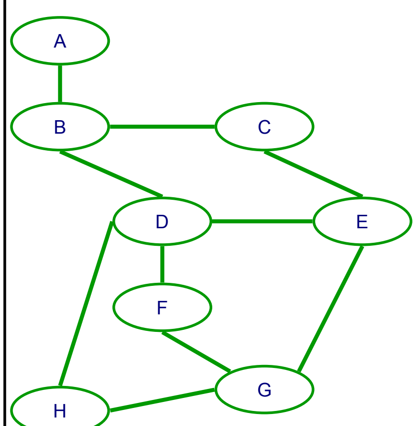
- **Chordal graph**:
  - Every cycle $X_1 - X_2 - \ldots - X_k - X_1$ with $k \geq 3$ has a chord
    - Edge $X_i - X_j$ for non-consecutive i & j
- **Theorem**:
  - Every induced graph is chordal
- "Optimal" elimination order easily obtained for chordal graph

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
  - ◁(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
  - ◁(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