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

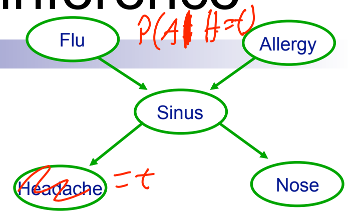
Variable Elimination

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



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

- Using def. of cond. prob.:

$$P(X \mid e) = \frac{P(X,e)}{P(e)} \propto P(X,e) \; : \; \text{compute } \forall x \Rightarrow P(X=x, E=e)$$

- Normalization:

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

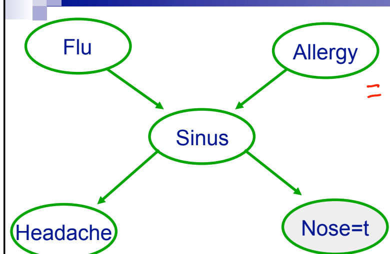
normalize

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

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

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

Probabilistic inference example



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

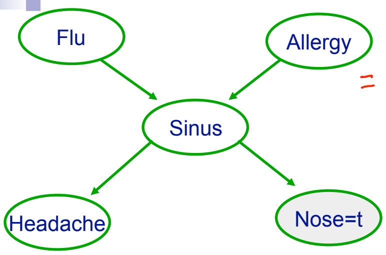
$$= \sum_f \sum_s \sum_h P(A, f, s, h, N=t)$$

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

Inference seems exponential in number of variables!

Fast probabilistic inference example – Variable elimination

$\boxed{\text{(Potential for) Exponential reduction in computation!}}$



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

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

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

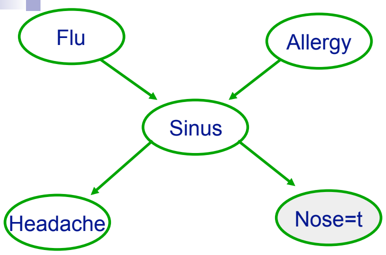
$$= \sum_f \sum_s P(f)\,P(A)\,P(s|f,A)\,P(N=t|s) \cancelto{1}{\sum_h P(h|s)}$$

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

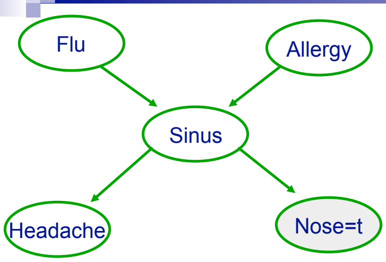
$$= P(A) \underbrace{\sum_f P(f)\; g_1(f,A)}_{g_2(A)} = P(A)\,g_2(A) = P(A,N=t)$$

Understanding variable elimination – Exploiting distributivity

Understanding variable elimination – Order can make a HUGE difference

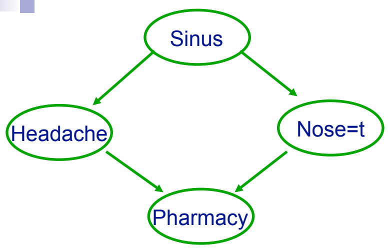


Understanding variable elimination – Intermediate results

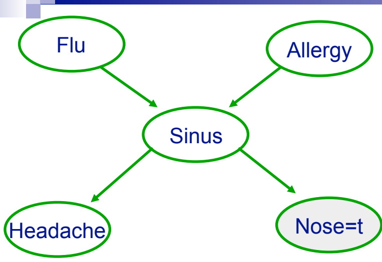


Intermediate results are probability distributions

Understanding variable elimination – Another example



Pruning irrelevant variables



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**
- Prune non-active vars for $\{X,\mathbf{e}\}$

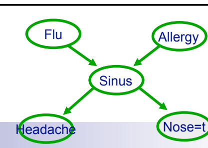
IMPORTANT!!!

- 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

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

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

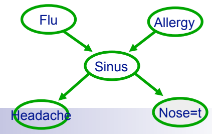
Operations on factors



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

Multiplication:

Operations on factors



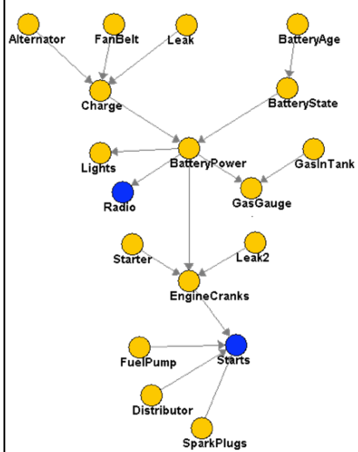
$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

Marginalization:

Complexity of VE – First analysis

- Number of multiplications:
- 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

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

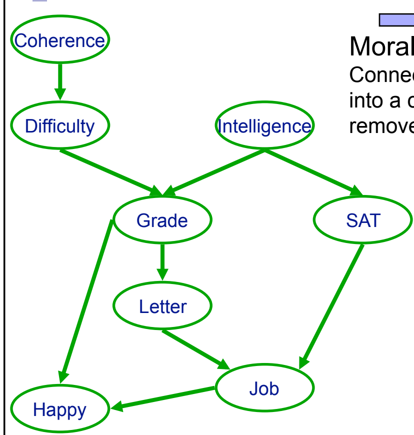
Announcements

- 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

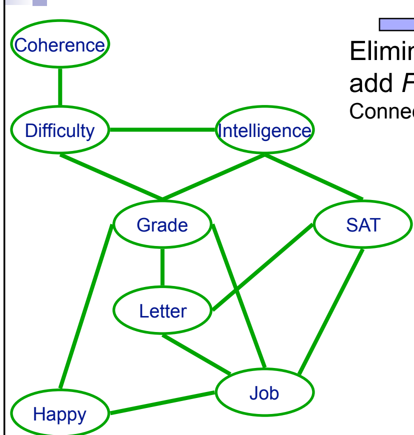
Complexity of variable elimination – Graphs with loops



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

Connect nodes that appear together in an initial factor

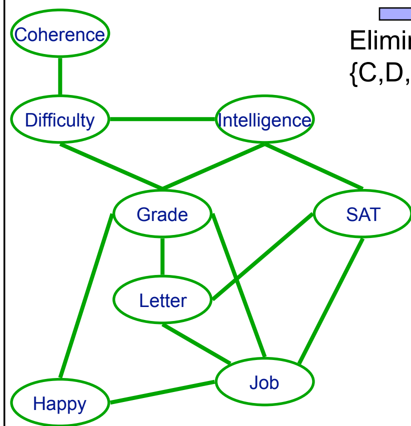
Eliminating a node – Fill edges



Eliminate variable
add *Fill Edges*:
Connect neighbors

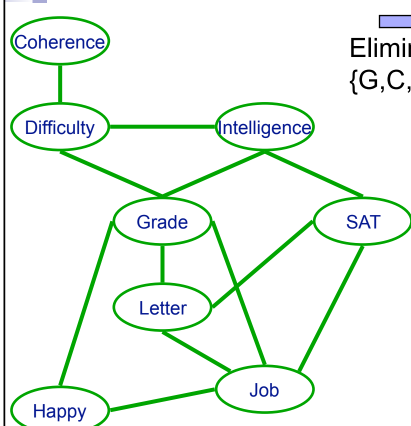
Induced graph

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



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

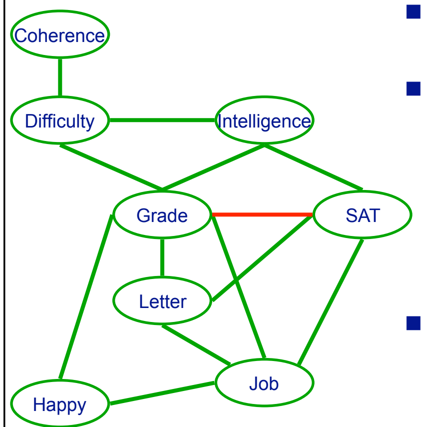
Different elimination order can lead to different induced graph



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

Induced graph and complexity of VE

Read complexity from cliques in induced graph



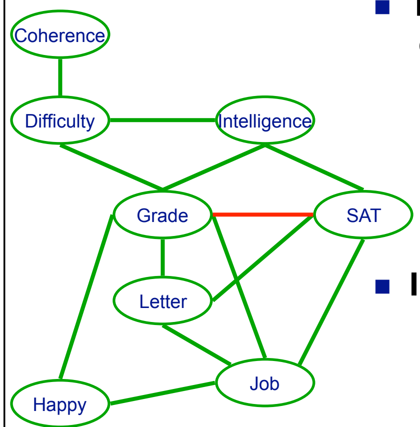
Elimination order:
{C,D,I,S,L,H,J,G}

- 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 ·

Example: Large induced-width with small number of parents

Compact representation $\not\to$ Easy inference ☹

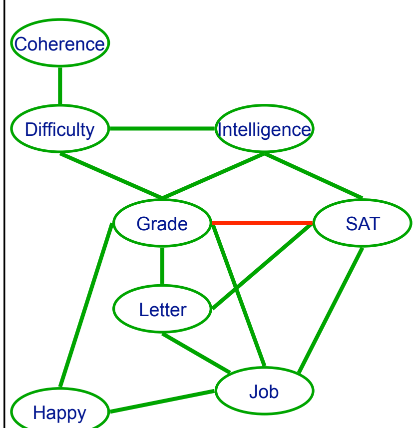
Finding optimal elimination order



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

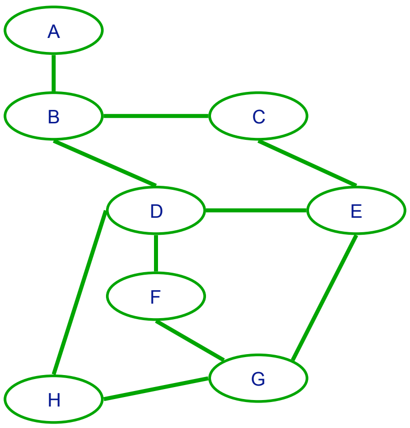
- **Theorem**: Finding best elimination order is NP-complete:
  - Decision problem: Given a graph, determine if there exists an elimination order that achieves induced width ≤ K
- **Interpretation**:
  - Hardness of finding elimination order in addition to hardness of inference
  - Actually, can find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

Induced graphs and chordal graphs



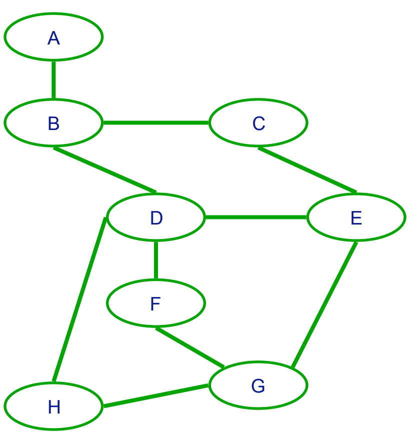
- **Chordal graph**:
  - Every cycle $X_1 - X_2 - \ldots - X_k - X_1$ with $k \geq 3$ has a chord
    - Edge $X_i - X_j$ for non-consecutive i & j
- **Theorem**:
  - Every induced graph is chordal
- "Optimal" elimination order easily obtained for chordal graph

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 ← unmarked var with most **marked** neighbors
  - ◁(X) ← 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 ← unmarked var whose elimination adds fewest edges
  - ◁(X) ← 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