Review of Variable Elimination

Compute $\Pr(C)$: eliminate $A$, then $B$

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\Theta_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>.6</td>
</tr>
<tr>
<td>false</td>
<td>.4</td>
</tr>
</tbody>
</table>

| $A$  | $B$  | $\Theta_{B|A}$ |
|------|------|----------------|
| true | true | .9             |
| true | false| .1             |
| false| true | .2             |
| false| false| .8             |

| $B$  | $C$  | $\Theta_{C|B}$ |
|------|------|----------------|
| true | true | .3             |
| true | false| .7             |
| false| true | .5             |
| false| false| .5             |
Review of Variable Elimination

Compute $\Pr(C)$: eliminate $A$, then $B$

Two factors mention $A$:
$\Theta_A$, $\Theta_{B|A}$
Multiplying $\Theta_A$ and $\Theta_{B|A}$

| $A$  | $B$  | $\Theta_{B|A}$ | $B$  | $C$  | $\Theta_{C|B}$ |
|------|------|---------------|------|------|---------------|
| true | true | .9            | true | true | .3            |
| true | false| .1            | true | false| .7            |
| false| true | .2            | false| true | .5            |
| false| false| .8            | false| false| .5            |

| $A$  | $B$  | $\Theta_A \Theta_{B|A}$ |
|------|------|--------------------------|
| true | true | .54                      |
| true | false| .06                      |
| false| true | .08                      |
| false| false| .32                      |
Review of Variable Elimination

Multiply $\Theta_A$ and $\Theta_{B|A}$

| $A$ | $B$ | $\Theta_{B|A}$ | $B$ | $C$ | $\Theta_{C|B}$ |
|-----|-----|----------------|-----|-----|----------------|
| true | true | .9             | true | true | .3             |
| true | false| .1             | true | false| .7             |
| false| true | .2             | false| true | .5             |
| false| false| .8             | false| false| .5             |

Sum out $A$

| $B$ | $\sum_A \Theta_A \Theta_{B|A}$ |
|-----|--------------------------------|
| true| .62 = .54 + .08                |
| false| .38 = .06 + .32               |
Review of Variable Elimination

Two factors left, $\Theta_{C|B}$ & $\sum_A \Theta_A \Theta_{B|A}$, multiply

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\Theta_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>.6</td>
</tr>
<tr>
<td>false</td>
<td>.4</td>
</tr>
</tbody>
</table>

| $A$ | $B$ | $\Theta_{B|A}$ |
|-----|-----|----------------|
| true| true| .9             |
| true| false| .1             |
| false| true| .2             |
| false| false| .8             |

| $B$ | $C$ | $\Theta_{C|B}$ |
|-----|-----|----------------|
| true| true| .3             |
| true| false| .7             |
| false| true| .5             |
| false| false| .5             |

| $B$ | $C$ | $\Theta_{C|B} \sum_A \Theta_A \Theta_{B|A}$ |
|-----|-----|---------------------------------------------|
| true| true| .186                                        |
| true| false| .434                                        |
| false| true| .190                                        |
| false| false| .190                                        |
Inference by Factor Elimination

Elimination Trees
Separators and Clusters
Message Passing
The Jointree Algorithm

Review of Variable Elimination

Two factors left, $\Theta_{C|B}$ &
$\sum_A \Theta_A \Theta_{B|A}$, multiply

| $B$   | $C$   | $\Theta_{C|B} \sum_A \Theta_A \Theta_{B|A}$ |
|-------|-------|------------------------------------------|
| true  | true  | .186                                     |
| true  | false | .434                                     |
| false | true  | .190                                     |
| false | false | .190                                     |

Sum out $B$

| $C$  | $\sum_B \Theta_{C|B} \sum_A \Theta_A \Theta_{B|A}$ |
|------|-------------------------------------------------|
| true | .376                                            |
| false| .624                                            |
Review of Variable Elimination

Two factors left, $\Theta_{C|B}$ & $\sum_A \Theta_A \Theta_{B|A}$, multiply

| B  | C  | $\Theta_{C|B} \sum_A \Theta_A \Theta_{B|A}$ |
|----|----|--------------------------------------------|
| true true | .186 |
| true false | .434 |
| false true | .190 |
| false false | .190 |

Sum out $B$

| C  | $\sum_B \Theta_{C|B} \sum_A \Theta_A \Theta_{B|A}$ |
|----|---------------------------------------------|
| true | .376 |
| false | .624 |
Review of Variable Elimination

- To compute marginal over variables $Q$, eliminate other variables, one at a time

- Eliminate variable $X$ by multiplying factors containing $X$ and summing out $X$

- $\prod$ of remaining factors give marginal over $Q$
Factor Elimination

- To compute marginal over variables $Q$, eliminate all factors except one that contains $Q$.

- Eliminate factor $\Phi$ by summing out “private” variables and multiplying result into any other factor.

- Remaining factor, projected on $Q$, gives marginal over $Q$. 

Jinbo Huang

Reasoning with Bayesian Networks
Inference by Factor Elimination

Factor Elimination

Compute $\Pr(C)$: eliminate $\Theta_A$ and $\Theta_{B|A}$

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\Theta_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>.6</td>
</tr>
<tr>
<td>false</td>
<td>.4</td>
</tr>
</tbody>
</table>

| $A$ | $B$ | $\Theta_{B|A}$ |
|-----|-----|----------------|
| true| true| .9             |
| true| false| .1            |
| false| true| .2             |
| false| false| .8          |

| $B$ | $C$ | $\Theta_{C|B}$ |
|-----|-----|----------------|
| true| true| .3             |
| true| false| .7           |
| false| true| .5             |
| false| false| .5          |
Factor Elimination

Compute $\Pr(C)$: eliminate $\Theta_A$ and $\Theta_{B|A}$

Eliminate $\Theta_A$: Sum out private variables (none), multiply into any other factor (let’s choose $\Theta_{B|A}$)

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\Theta_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>.6</td>
</tr>
<tr>
<td>false</td>
<td>.4</td>
</tr>
</tbody>
</table>

| $A$ | $B$   | $\Theta_{B|A}$ |
|-----|-------|----------------|
| true | true | .9             |
| true | false| .1             |
| false| true | .2             |
| false| false| .8             |

| $B$ | $C$   | $\Theta_{C|B}$ |
|-----|-------|----------------|
| true| true | .3             |
| true| false| .7             |
| false| true | .5             |
| false| false| .5             |
Factor Elimination

Compute $\Pr(C)$: eliminate $\Theta_A$ and $\Theta_{B|A}$

Eliminate $\Theta_A$: Sum out private variables (none), multiply into any other factor (let’s choose $\Theta_{B|A}$)

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\Theta_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>.6</td>
</tr>
<tr>
<td>false</td>
<td>.4</td>
</tr>
</tbody>
</table>

| $A$ | $B$ | $\Theta_{B|A}$ |
|-----|-----|----------------|
| true| true| .9             |
| true| false| .1             |
| false| true| .2             |
| false| false| .8             |

| $B$ | $C$ | $\Theta_{C|B}$ |
|-----|-----|----------------|
| true| true| .3             |
| true| false| .7             |
| false| true| .5             |
| false| false| .5             |

| $A$ | $B$ | $\Theta_A \Theta_{B|A}$ |
|-----|-----|-------------------------|
| true| true| .54                     |
| true| false| .06                     |
| false| true| .08                     |
| false| false| .32                     |
Factor Elimination

Eliminate $\Theta_A \Theta_B|_A$: Sum out private variables ($A$)

| $B$    | $\sum_A \Theta_A \Theta_B|_A$ |
|--------|-------------------------------|
| true   | $0.62 = 0.54 + 0.08$         |
| false  | $0.38 = 0.06 + 0.32$         |

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\Theta_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>0.6</td>
</tr>
<tr>
<td>false</td>
<td>0.4</td>
</tr>
</tbody>
</table>

| $A$  | $B$  | $\Theta_B|_A$ |
|------|------|--------------|
| true | true | 0.9          |
| true | false| 0.1          |
| false| true | 0.2          |
| false| false| 0.8          |

| $B$    | $C$    | $\Theta_C|B$ |
|--------|--------|--------------|
| true   | true   | 0.3          |
| true   | false  | 0.7          |
| false  | true   | 0.5          |
| false  | false  | 0.5          |
Factor Elimination

Eliminate $\Theta_A\Theta_B|A$: Sum out private variables (A)

\[
\begin{array}{c|c}
B & \sum_A \Theta_A\Theta_B|A \\
\hline
\text{true} & .62 = .54 + .08 \\
\text{false} & .38 = .06 + .32 \\
\end{array}
\]

Multiply into $\Theta_C|B$

\[
\begin{array}{c|c|c}
B & C & \Theta_C|B \sum_A \Theta_A\Theta_B|A \\
\hline
\text{true} & \text{true} & .186 \\
\text{true} & \text{false} & .434 \\
\text{false} & \text{true} & .190 \\
\text{false} & \text{false} & .190 \\
\end{array}
\]
### Factor Elimination

#### Sum out B

| C     | $\sum_B \Theta_{C|B} \sum_A \Theta_A \Theta_{B|A}$ |
|-------|-------------------------------------------------|
| true  | .376                                           |
| false | .624                                           |

#### Factor Elimination Table

<table>
<thead>
<tr>
<th>A</th>
<th>$\Theta_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>.6</td>
</tr>
<tr>
<td>false</td>
<td>.4</td>
</tr>
</tbody>
</table>

| A  | B   | $\Theta_{B|A}$ |
|----|-----|----------------|
| true | true | .9          |
| true | false | .1         |
| false | true  | .2         |
| false | false | .8         |

| B  | C   | $\Theta_{C|B}$ |
|----|-----|----------------|
| true | true | .3          |
| true | false | .7         |
| false | true  | .5          |
| false | false | .5         |
Elimination Trees

- Two choices per step: Which $\Phi$ to eliminate, which $\Phi$ to multiply into

- Specified by *elimination tree*
Elimination Trees

Winter? (A)

Sprinkler? (B)

Rain? (C)

Wet Grass? (D)

Slippery Road? (E)

Elimination Trees

1

2

3

f(A)

f(AB)

f(AC)

f(AC)

1

2

3

4

5

f(AB)

f(BCD)

f(CE)

f(AB)

f(BCD)

f(CE)

Jinbo Huang

Reasoning with Bayesian Networks
Elimination Trees

Eliminate $f_2$

Eliminate $f_1$

Eliminate $f_5$

Eliminate $f_4$
Separators

Intersection of variables on either side of edge

\[ f(A) \]
\[ AB \]
\[ f(AB) \]

\[ AB \]
\[ f(AC) \]
\[ BC \]
\[ C \]

\[ f(BCD) \]
\[ f(CE) \]

\[ f(AB) \]
Separators

Intersection of variables on either side of edge

Between 2 steps of elimination (summing out & multiplication), factor contains precisely separator variables
Clusters

Variables of node union all surrounding separators

1
\[ AB \]

2
\[ AB \]

3
\[ ABC \]

4
\[ BCD \]

5
\[ CE \]
Clusters

Variables of node union all surrounding separators

Before both steps of elimination (summing out & multiplication), factor contains precisely cluster variables
Clusters

Variables of node union all surrounding separators

Before both steps of elimination (summing out & multiplication), factor contains precisely cluster variables

width

\[= \max |\text{cluster}| - 1\]

Given elimination order, can generate elimination tree of same width
Factor Elimination as Message Passing

View eliminated factor as *message* to neighbor (to be multiplied)
View eliminated factor as *message* to neighbor (to be multiplied)

After one message per edge, result obtained at root
Factor Elimination as Message Passing

View eliminated factor as *message* to neighbor (to be multiplied)

Size of any factor computed $\leq \exp(\text{width})$

After one message per edge, result obtained at root
Multiple Queries and Message Reuse
Multiple Queries and Message Reuse
Multiple Queries and Message Reuse

Inward and outward passes

Two messages per edge answer all queries
Complexity of Message Passing

- Computing message involves $\prod$ and $\sum$
- Both $O(\exp(w))$, $w = width$ bounds size of biggest factor
- Assumes each node has bounded $\#$ of neighbors
  - Given BN with $n$ variables and treewidth $w$, $\exists$ elimination tree with $O(n)$ edges, width $w$, and bounded degree
- 2 messages/edge; total complexity $O(n \exp(w))$
- Compared with $O(n^2 \exp(w))$ for variable elimination for all marginals
The Polytree Algorithm

- If Bayesian network has polytree structure, can use that as elimination tree (after dropping directionality)

- Width $k = \max \# \text{ of parents of any node}$

- Linear complexity $O(n \exp(k))$ for bounded $k$
Jointree for DAG $G$ is a tree where each node $i$ is labeled with cluster $C_i$ such that:

- Each $C_i$ is a set of nodes of $G$
- Each family of $G$ is contained in some $C_i$
- $C_i \cap C_j$ is contained in every cluster on the path between nodes $i$ and $j$ (jointree property, running intersection property)
The Jointree Connection

Jointree for DAG $G$ is a tree where each node $i$ is labeled with cluster $C_i$ such that:

- Each $C_i$ is a set of nodes of $G$
- Each family of $G$ is contained in some $C_i$
- $C_i \cap C_j$ is contained in every cluster on the path between nodes $i$ and $j$ (*jointree property, running intersection property*)

Minimal jointree: ceases to be jointree with any variable removed from any cluster

$$\text{Width of jointree} \stackrel{\text{def}}{=} \max |C_i| - 1$$

For edge $i \rightarrow j$, separator $S_{ij} \stackrel{\text{def}}{=} C_i \cap C_j$
DAG and Its Jointree

Inference by Factor Elimination
Inference by Conditioning
Factor Elimination
Elimination Trees
Separators and Clusters
Message Passing
The Jointree Algorithm

Jinbo Huang
Reasoning with Bayesian Networks
Jointrees and Elimination Trees

- Theorem: Clusters of elimination tree satisfy the three properties of jointree
  - Every elimination tree induces a jointree of same width
Jointrees and Elimination Trees

Theorem: Clusters of elimination tree satisfy the three properties of jointree
  - Every elimination tree induces a jointree of same width

Definition: Elimination tree \((\mathcal{T}, \phi)\) is embedded in jointree \((\mathcal{T}, \mathcal{C})\) if \(\text{vars}(\phi) \subseteq \mathcal{C}_i\)
  - To construct embedded elimination tree, adopt \(\mathcal{T}\) of jointree, assign \(\phi\) to node \(i\) such that \(\text{vars}(\phi) \subseteq \mathcal{C}_i\)
Jointrees and Elimination Trees

- **Theorem:** Clusters of elimination tree satisfy the three properties of jointree
  - Every elimination tree induces a jointree of same width

- **Definition:** Elimination tree $(\mathcal{T}, \phi)$ is *embedded* in jointree $(\mathcal{T}, \mathcal{C})$ if $\text{vars}(\phi_i) \subseteq \mathcal{C}_i$
  - To construct embedded elimination tree, adopt $\mathcal{T}$ of jointree, assign $\phi$ to node $i$ such that $\text{vars}(\phi) \subseteq \mathcal{C}_i$

- **Theorem:** $S_{ij}$ and $\mathcal{C}_i$ of embedded elimination tree $\subseteq$ jointree counterparts
  - Every jointree leads to an elimination tree of no greater width
The Jointree Algorithm

- Construct jointree for given Bayesian network

- Assign each CPT $\Theta_{X|U}$ to a cluster that contains $X$ and $U$ (guaranteed by properties of jointree)

- Two methods for message passing
  - Shenoy-Shafer: same as factor elimination
  - Hugin: uses division to compute messages
Inward messages toward root, outward messages from root, totaling 2 messages/edge

Node $x$ sends message to $m$ after receiving messages $x_1, \ldots, x_{m-1}$ from all other neighbors

$$x_m = \text{product of } x_1, \ldots, x_{m-1} \text{ and } \phi_x, \text{ projected on } S_{xm}$$
Shenoy-Shafer

- Time complexity $O(n \exp(w))$ as factor elimination, using appropriate jointree (with bounded degree)

- Stored messages (for reuse in outward pass) over separator variables; space complexity exponential only in separator size ($\leq w$)


Hugin

\[ x_m = \text{product of } \phi_x \text{ and } x_1, \ldots, x_m, \text{ projected on } S_{x_m}, \text{ divided by } x_m \]

Inward pass same as Shenoy-Shafer
Outward pass

- Root already has \( \prod \) of \( \phi_x \) and \( x_1, \ldots, x_m \); computes each message by projection and division
- Other node already has \( \prod \) of \( \phi_x \) and messages from all but one neighbor; last one received from direction of root
Hugin

- Factor in cluster gives $\Pr(C_i, e)$ as Shenoy-Shafer
- In addition, factor on edge (saved just before division is performed to compute message) gives $\Pr(S_{ij}, e)$
- Space complexity $O(n \exp(w))$, higher than Shenoy-Shafer
- Time complexity $O(n \exp(w))$, does not require bounded degree
Inference by Factor Elimination: Summary

- Marginals by factor elimination
- Elimination trees, separators, clusters, width
- Factor elimination as message passing
- Connection to inward and outward message passing on jointree
- Two flavors of jointree algorithm: Shenoy-Shafer and Hugin
Conditioning

- Also known as *case analysis*

\[ \Pr(\alpha) = \sum_{i=1}^{n} \Pr(\alpha \land \beta_i) \]

- Making assumption \( \beta_i \) can simplify problem

- Simplifying problem in different ways: *cutset conditioning* and *recursive conditioning*
Review of Network Pruning

- Consider computation of \( \Pr(Q, e) \) (includes prior marginals and probability of evidence as special cases)

- Pruning nodes: All leaves \( \notin Q \cup E \), interatively
  - Worst case: All leaves \( \in Q \cup E \), no pruning
  - Best case: All \( Q \cup E \) are roots, every node \( \notin Q \cup E \) pruned

- Pruning edges: For each edge \( U \rightarrow X, U \in E \)
  - Remove edge, shrink CPT \( \Theta_{X|U} \) by removing rows inconsistent with \( e \) and removing column \( U \)

- Effective treewidth is treewidth of pruned network given query
Try to reduce network to polytree
Pr(E, D = true)?
Try to reduce network to polytree

Pr(E, D = true)? No pruning available
Try to reduce network to polytree

Pr(\(E, D = true\))? No pruning available

How about Pr(\(E, D = true, B = true\))
+ Pr(\(E, D = true, B = false\))
Cutset Conditioning

Try to reduce network to polytree

Pr(E, D = true)? No pruning available

How about Pr(E, D = true, B = true) + Pr(E, D = true, B = false)
Nodes $C$ is a *loop-cutset* if removing edges outgoing from $C$ results in polytree
Cutset Conditioning

- Case analysis over loop-cutset
- Time complexity \( s = |\text{loop-cutset}| \)
  \[
  O(n \exp(k)) \cdot O(\exp(s)) = O(n \exp(k + s))
  \]
- Finding smallest loop-cutset is NP-hard
- Space complexity \( O(n \exp(k)) \); compares favorably with \( O(n \exp(w)) \) for elimination
Cutset Conditioning

Inefficient when loop-cutset is large
Recursive Conditioning

Different way to simplify problem: divide-and-conquer

A → B → C → D → E

B

A → B

C → D → E

C

C → D → E
Dtrees

Full binary tree whose leaves correspond to network CPTs
Dtrees

Cutset: $\cap$ of variables on either side minus acutset ($\cup$ of ancestor cutsets)
Recursive Conditioning: Basic Algorithm

- Case analysis on root cutset, considering subset of $\exp(|\text{cutset}|)$ cases compatible with evidence

- Each case returns $\prod$ of results from two recursive calls on left and right child

- Return $\sum$ of results from all cases
Inference by Factor Elimination
Inference by Conditioning

Complexity

- \# of recursive calls to non-root node \( \leq \text{acutset}\# \)
- Time \( O(n \exp(w \log n)) \), space \( O(wn) \), where \( w \) is cutset width, assuming balanced dtrees
  - Can turn elimination order of width \( w \) into dtree with cutset width \( \leq w + 1 \)
  - Can balance dtree while keeping cutset width \( \leq w + 1 \)

- Potentially more time, but less space, than variable elimination

- Not comparable to cutset conditioning
Redundant Computation

A → B → C → D → E → F → G → H

Cutsets

A → B → C → D → E → F → G → H

A

AB

BC

CD

DE

EF

FG

GH

A

AB

BC

CD

DE

EF

FG

GH

a-cutsets

A

AB

ABC

ABCDEF

GH

A

AB

ABC

ABCDEF

GH
Improving Time Complexity: Caching

Context: $\bigcap$ of variables and acutset

contexts

a-cutsets
Cache result for each instantiation of context

contexts

a-cutsets
Improving Time Complexity: Caching

\[
\text{# of recursive calls to node } T \leq \text{cutset}(T^p) \times \text{context}(T^p)
\]

contexts

\[ \text{AB} \quad \text{BC} \quad \text{CD} \quad \text{DE} \quad \text{EF} \quad \text{FG} \quad \text{GH} \]

cutsets

\[ \text{AB} \quad \text{BC} \quad \text{CD} \quad \text{DE} \quad \text{EF} \quad \text{FG} \quad \text{GH} \]
Full Caching: Complexity

- \( Cluster \overset{\text{def}}{=} \text{variables, for leaf} \)
- \( Cluster \overset{\text{def}}{=} \text{cutset } \cup \text{ context, for internal node} \)
  - Cutset and context always disjoint

- Width of dtree is \( \max |cluster| - 1 \)

- Time and space complexity \( O(n \exp(w)) \), same as variable elimination
  - Elimination order of width \( w \) can be turned into dtree of no greater width
  - Space complexity actually exponential only in context width
Any-space Inference

Two extremes of recursive conditioning

- No caching: \(O(n \exp(w \log n))\) time, \(O(wn)\) space
- Full caching: \(O(n \exp(w))\) time and space

Can be anywhere in between

Cache factor, \(0 \leq cf(T) \leq 1\), is fraction of cache entries that will be filled

\(cf(T) = 0\) for all nodes: no caching
\(cf(T) = 1\) for all nodes: full caching
Any-space Inference: Complexity

Average \# of recursive calls to node $T$, $\text{ave}(T)$ \leq \text{cutset}(T^p)\#[\text{cf}(T^p)\text{context}(T^p)\# + (1 - \text{cf}(T^p))\text{ave}(T^p)]$

Exact for \textit{discrete} cache factors (0 or 1)

No caching
- $\text{ave}(T) \leq \text{cutset}(T^p)\#\text{ave}(T^p)$
- Solution: $\text{ave}(T) \leq \text{acutset}(T)\#$

Full caching
- $\text{ave}(T) \leq \text{cutset}(T^p)\#\text{context}(T^p)\#$
Decomposition Graphs

A *dgraph* is a set of dtrees that share structure.
Decomposition Graphs

RC on dgraph amounts to RC on every dtree
Decomposition Graphs: Multiple Marginals

- RC on dgraph more efficient than on separate dtrees, due to structure sharing

- Besides $\Pr(e)$, RC computes $\Pr(C, e)$ for root cutset $C$

- Hence RC on dgraph computes multiple marginals, similar to jointree
Decomposition Graphs: Construction

- Given dtree, can construct dgraph that allows computation of all family marginals
  - Each dtree leaf (network family) becomes child of one new dgraph root

- For internal node $X$, will have root cutset $XU$

- For leaf $X$, will have root cutset $U$
  - $Pr(XU, e) = Pr(U, e) Pr(X|U, e)$
  - $Pr(X|U, e) = Pr(X|U)$
Decomposition Graphs: Complexity

- Given dtree with $n \geq 5$ and width $w$, can construct such a dgraph with $(5n - 7)/2$ nodes and $4(n - 2)$ edges, and width $w$.

- RC with full caching on dgraph runs in $O(n \exp(w))$ time and space, same as jointree.
Cache Allocation

- No enough memory for full caching, what caches to keep?

- Caches aren’t equally useful
  - In particular, *dead cache* if context($T$) = cluster($T^p$)

- Consider discrete cache factors only

- Search for cache factors that minimize running time, computed by summing # of recursive calls to each node
Cache Allocation: Systematic Search

Depth-first search for assignment of $cf$ 0/1 to each node in dgraph

Diagram:

- $T_1$ with children $T_2$ and $T_3$
- $T_2$ with children $T_2$ and $T_3$
- $T_3$ with children $G_0$, $G_1$, $G_2$, and $G_3$
Cache Allocation: Systematic Search

Prune with lower bound computed by assigning 1 to all undecided nodes

Branch on all parents before their child
  - Lower bound monotonically improves
  - Lower bound computed in constant time

Variable ordering: choose node requiring largest cache (context\#)

Value ordering: try $cf = 1$ before $cf = 0$
Cache Allocation: Greedy Search

- Score for each node: reduced # of recursive calls (to any node) per cache entry

- Greedily assign $cf = 1$ to node with highest score

- Update scores after each assignment
  - $O(n^2)$ if done naively
  - Can be done in $O(n)$ by lazy updates

- Total complexity $O(n^2)$
Inference by Conditioning: Summary

- Simplify problem by case analysis
- Loop-cutsets and cutset conditioning: reduce to polytree
- Dtrees and recursive conditioning: divide-and-conquer
- Caching and time-space tradeoff
- Multiple marginals using dgraphs
- Systematic and greedy search for caching allocation