Graph-guided Sampling

Rina Dechter
Bren school of Information and Computer Sciences
University of California, Irvine

Joint work with Vibhav Gogate and Bozhena Bidyuk

HUJI Dec 2011
Outline

• Background: Bayesian networks
• Importance Sampling
• AND/OR search space for graphical Models
• AND/OR Sampling: exploiting structure
• AND/OR+cutset sampling
• UAI 2010 experience

HUJI Dec 2011
Directed graphical models: Bayesian networks
(Pearl 1988)

Variables: \{X_{l1}, \ldots, X_{ln}\}

\[ P(x) = \prod_{i \uparrow \leftarrow} P(x_{li} | pa(x_{li})) \]

Basic inference task: compute the probability of evidence (likelihood computation)

\[ P(E=e) = \sum_{Z \uparrow \leftarrow} P(Z=z, E=e) = \sum_{Z \uparrow \leftarrow} \prod_{i \uparrow \leftarrow} [P(z_{li} | pa(z_{li})) | E=e] \]

\[ P(S=s_{\uparrow 0}, D=d_{\uparrow 1}) = \sum_{l, b, r} P(s_{\uparrow 0}) Pl(s_{\uparrow 0}) Pb_{\uparrow 0} Pr_{\uparrow 0} s_{\uparrow 0} \ P(d | b, l) \]
Undirected graphical models: Markov networks

Node potentials: $\phi_i(x)$
Edge potentials: $\phi_{ij}(x)$

$P(x) = \frac{1}{Z} \prod_{i \uparrow} \phi_i(x) \prod_{i,j \uparrow} \phi_{ij}(x)$

$Z = \sum x \prod_{i \uparrow} \phi_i(x) \prod_{i,j \uparrow} \phi_{ij}(x)$

$Z$: Partition function

• Inference task: compute the partition function $Z$
  – Sum-product problem
  – Qualitatively same as computing $P(e)$ in Bayesian networks
    • Counting solutions of a SAT or a CSP
Other inference tasks

- Compute the conditional marginal probability given evidence (MAR task)
  \[ P(L=l0|S=s0,D=d1) \]
- Compute the most probable explanation (MPE)
  - Find a maximum probability assignment
  - Abductive inference
Inference vs conditioning-search

Inference

Exp(w*) time/space

Search

Exp(n) time
O(n) space

Search+inference:
Space: exp(w)
Time: exp(w+c(w))

W: user controlled
Approximation

- Since inference, search and hybrids are too expensive when graph is dense; (high treewidth) then:
  - **Bounding inference:**
    - mini-bucket and mini-clustering
    - Belief propagation

- **Bounding search:**
  - **Sampling**

- Goal: an anytime scheme
Outline

• Background: Bayesian networks
• Importance Sampling
  • AND/OR search space for graphical Models
  • AND/OR Sampling: exploiting structure
  • AND/OR+cutset sampling
• UAI 2010 experience

HUJI Dec 2011
Monte Carlo Estimate

- **Estimator:**
  - An estimator is a function of the samples.
  - It produces an estimate of the unknown parameter of the sampling distribution.

Given i.i.d. samples $S^1, S^2, \ldots S^T$ drawn from $P$, the Monte carlo estimate of $\mathbb{E}_P[g(x)]$ is given by:

$$\hat{g} = \frac{1}{T} \sum_{t=1}^{T} g(S^t)$$

So the key is to generate samples from $P$

HUJI Dec 2011
Logic sampling (example)

\[ P(X_1, X_2, X_3, X_4) = P(X_1) \times P(X_2 | X_1) \times P(X_3 | X_1) \times P(X_4 | X_2, X_3) \]

But with evidence we have rejection...
In general it may be hard to generate a sample, especially on Markov networks.

HUJI Dec 2011
Importance Sampling: Overview

\[ P(e) = \sum_{X \in E} P_B(X, E = e) \]

\[ M = \sum_{z \in Z} f(z) = P(e) \text{ where } Z = X \setminus E \]

• Given a proposal or importance distribution \( Q(z) \) such that \( f(z) > 0 \) implies \( Q(z) > 0 \), rewrite

\[ M = \sum_{z \in Z} f(z) = \sum_{z \in Z} f(z) \frac{Q(z)}{Q(z)} = E_Q \left[ \frac{f(z)}{Q(z)} \right] \]

■ Given i.i.d. samples \( z_1, \ldots, z_N \) from \( Q(z) \),

\[ \hat{M} = \frac{1}{N} \sum_{j=1}^{N} \frac{f(z_j)}{Q(z_j)} = \frac{1}{N} \sum_{j=1}^{N} w(z_j) \quad E_Q[\hat{M}] = M = P(e) \]

HUJI Dec 2011
Variance reduction

• MSE of estimator $Z$ is equals its variance when bias is zero.

$$Var_Q[\hat{Z}] = Var_Q \left[ \frac{1}{N} \sum_{i=1}^{N} w(x_i) \right] = \frac{Var_Q[w(x)]}{N}$$

• Performance depends on the variance of the estimate
  • How close $Q(Z)$ is to $P(Z|E)$.
  • In practice, we can compute $Q$ from the output of Generalized Belief Propagation (UAI, 2005)

• The number of samples;
AND/OR sampling: motivation

\[ Z = \sum_{x} f(x) = \sum_{x_1, x_2, x_3, x_4} f_A(x_1, x_2, x_3) f_B(x_2, x_3, x_4) f_C(x_4) \]

Total disregard for the structure of \( f(x) \) while approximating it.

\[ Z = \sum_{x \in X} f(x) = \sum_{x \in X} f(x) \frac{Q(x)}{Q(x)} = E_Q \left[ \frac{f(x)}{Q(x)} \right] \]

Use AND/OR search to better exploit the structure
Outline

• Background: Bayesian networks
• Importance Sampling

• AND/OR search space for graphical Models
• AND/OR Sampling: exploiting structure
• AND/OR+w-cutset sampling
• UAI 2010 experience

HUJI Dec 2011
Conditioning generates the probability tree

\[ P(a, e = 0) = P(a) \sum_b P(b \mid a) \sum_c P(c \mid a) \sum_b P(d \mid a, b) \sum_{e=0} P(e \mid b, c) \]

Complexity of conditioning: exponential time, linear space
AND/OR search spaces for exact inference

(Dechter and Mateescu, 2007)

Graphical model

Chain Pseudo-tree

Pseudo-tree

AND/OR Tree

HUJI Dec 2011
The AND/OR search tree

A solution subtree is \((A=0, B=1, C=0, D=0, E=1, F=1)\)
AND/OR search spaces for exact reasoning

Task: compute the value of a root node

\[ P(E | A, B) \quad P(B | A) \quad P(C | A) \quad P(A) \]

\[
\begin{array}{c|cc}
A & B & E=0 \quad E=1 \\
\hline
0 & 0 & .4 \quad .6 \\
0 & 1 & .5 \quad .5 \\
1 & 0 & .7 \quad .3 \\
1 & 1 & .2 \quad .8 \\
\end{array}
\]

\[
\begin{array}{c|cc}
A & B=0 \quad B=1 \\
\hline
0 & .4 \quad .6 \\
1 & .7 \quad .3 \\
\end{array}
\]

\[
\begin{array}{c|cc}
A & C=0 \quad C=1 \\
\hline
0 & .2 \quad .8 \\
1 & .6 \quad .4 \\
\end{array}
\]

\[
\begin{array}{c|c}
A \quad P(A) \\
\hline
0 & .6 \\
1 & .4 \\
\end{array}
\]

Result: \[ P(D=1, E=0) \]

**Value** of node = updated belief for sub-problem below

\[ P(D | B, C) \]

\[
\begin{array}{c|cc}
B & C & D=0 \quad D=1 \\
\hline
0 & 0 & .2 \quad .8 \\
0 & 1 & .1 \quad .9 \\
1 & 0 & .3 \quad .7 \\
1 & 1 & .5 \quad .5 \\
\end{array}
\]

Evidence: \( D=1 \)

OR node: Marginalization operator (summation)

AND node: Combination operator (product)
## Complexity of AND/OR tree search

<table>
<thead>
<tr>
<th></th>
<th>AND/OR tree</th>
<th>OR tree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space</strong></td>
<td>$O(n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>$O(n , d^t)$</td>
<td>$O(d^n)$</td>
</tr>
<tr>
<td></td>
<td>$O(n , d^{w*} \log n)$</td>
<td></td>
</tr>
</tbody>
</table>

(Freuder & Quinn85), (Collin, Dechter & Katz91), (Bayardo & Miranker95), (Darwiche01)

\(d\) = domain size  
\(t\) = depth of pseudo-tree  
\(n\) = number of variables  
\(w^*\) = treewidth
Outline

• Background: Bayesian networks
• Importance Sampling
• AND/OR Sampling: exploiting structure
• AND/OR+w-cutset sampling
• UAI 2010 experience

HUJI Dec 2011
AND/OR importance sampling
(general idea)

- Decompose Expectation

\[
P(d, f) = \sum_{a,b,c} P(a)P(c \mid a)P(b \mid a)P(d \mid b)P(f \mid c)
\]

\[
Q(A, B, C) = Q(A)Q(B \mid A)Q(C \mid A)
\]

\[
P(d, f) = \sum_{a,b,c} \frac{P(a)P(c \mid a)P(b \mid a)P(d \mid b)P(f \mid c)}{Q(a)Q(b \mid a)Q(c \mid a)} \frac{Q(a)Q(b \mid a)Q(c \mid a)}{Q(a)Q(b \mid a)Q(c \mid a)}
\]

\[
= E_Q \left[ \frac{P(a)P(c \mid a)P(b \mid a)P(d \mid b)P(f \mid c)}{Q(a)Q(b \mid a)Q(c \mid a)} \right]
\]
AND/OR importance sampling (general idea)

• Decompose Expectation

\[
P(d, f) = \sum_{a, b, c} \frac{P(a)P(c | a)P(b | a)P(d | b)P(f | c)}{Q(a)Q(b | a)Q(c | a)} Q(a)Q(b | a)Q(c | a)
\]

\[
P(d, f) = \sum_a \frac{P(a)}{Q(a)} Q(a) \sum_b \frac{P(b | a)P(d | b)}{Q(b | a)} Q(b | a) \sum_c \frac{P(c | a)P(f | c)}{Q(c | a)} Q(c | a)
\]

\[
P(d, f) = E_Q \left[ \frac{P(a)}{Q(a)} \right] E_Q \left[ \frac{P(b | a)P(d | b)}{Q(b | a)} \bigg| a \right] E_Q \left[ \frac{P(c | a)P(f | c)}{Q(c | a)} \bigg| a \right]
\]
AND/OR importance sampling (general idea)

\[ P(d, f) = E_Q \left[ \frac{P(a)}{Q(a)} E_Q \left[ \frac{P(b \mid a) P(d \mid b)}{Q(b \mid a)} \mid a \right] E_Q \left[ \frac{P(c \mid a) P(f \mid c)}{Q(c \mid a)} \mid a \right] \right] \]

- Compute all expectations separately
- How?
  - Record all samples
  - For each sample that has A=a
    - Estimate the conditional expectations separately using the generated samples
    - Combine the results
AND/OR tree importance sampling

\[ Z = E_Q \left[ \frac{f(a)}{Q(a)} \right] E_Q \left[ \frac{f(b, a)}{Q(b | a)} \right] E_Q \left[ \frac{f(c, a)}{Q(c | a)} \right] \]

\[ \hat{E} \left[ \frac{f(B, A = 0)}{Q(B | A = 0)} \right] \times \hat{E} \left[ \frac{f(C, A = 0)}{Q(C | A = 0)} \right] \]

Sample # | A | B | C
---|---|---|---
1 | 0 | 1 | 0
2 | 0 | 2 | 1
3 | 1 | 1 | 1
4 | 1 | 2 | 0

Pseudo-tree
AND/OR importance sampling

![Pseudo-tree diagram]

**Estimate of**

\[
E \left[ \frac{P(b | A = 0)P(d | b)}{Q(b | A = 0)} \mid A = 0 \right]
\]

= Average Weight of samples of B having \( A = 0 \)

\[
= \frac{w(B = 1, A = 0) + w(B = 2, A = 0)}{2}
\]
AND/OR sample tree

• Given a pseudo-tree and a set of samples, $S$
• 1. Generate the AND/OR search tree
• 2. Remove nodes/edges not appearing in $S$.
• 3. Assign weights $(w(n,m), #(n,m))$
  – $w(n,m) =$ the weights $(p/q)$
  – $#(n,m) =$ frequency of the partial sample in $S$
• The AOT estimate is the value of the root node.

HUJI Dec 2011
Algorithm AND/OR importance sampling

1. Generate samples \( x_1, \ldots , x_N \) from \( Q \) along \( O \).
2. Build a AND/OR sample tree for the samples \( x_1, \ldots , x_N \) along the ordering \( O \).
3. FOR all leaf nodes \( i \) of AND-OR tree do
   1. IF AND-node \( v(i) = 1 \) ELSE \( v(i) = 0 \)
4. FOR every node \( n \) from leaves to the root do
   1. IF AND-node \( v(n) = \text{product of children} \)
   2. IF OR-node \( v(n) = \text{Average of children} \)
A Bayesian network

Figure 3: A Bayesian network and its CPTs

HUJI Dec 2011
Example of AND/OR sample tree

```
<table>
<thead>
<tr>
<th>Sample #</th>
<th>Z</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Figure 5: (a) Four samples drawn from a uniform proposal distribution $Q(X,Y,Z) = Q(X)Q(Y)Q(Z)$ where $Q(Z = 0) = Q(Z = 1) = 1/2$, $Q(X = 0) = Q(X = 1) = Q(X = 2) = 1/3$ and $Q(Y = 0) = Q(Y = 1) = Q(Y = 2) = 1/3$, (b) The samples in (a) arranged on a full AND/OR search tree. Dotted edges and nodes are not sampled. Each arc from an OR node to an AND node is labeled by its weight and frequency (see Definition 14).
Example of value cor

\[
\frac{(0.9945 \times 1.6 \times 2) + (0.207 \times 0.4 \times 2)}{2 + 2} = 0.12096
\]

Figure 6: Value computation on an AND/OR sample tree (see Definition 15). Each OR and AND node is annotated with its value. By definition, the value of a leaf AND node is 1 and is not shown to avoid clutter. The AND/OR sample tree mean (which equals the value of the root OR node Z) is 0.12096.
AND/OR sampling: more virtual samples

• Example: $A=0$, $B=2$, $C=0$ is not generated but still considered in the AND/OR space
• 8 virtual samples in AND/OR space versus 4 samples.
• In general, we get exponentially more samples.
Properties of AO-Tree Mean (AOTM)

- AOTM is an unbiased estimator of $Z$
- On a chain it reduces to IS estimator
- **Theorem:** Variance of AOTM is always smaller or equal to regular IS
- Time complexity: $O(nN)$, $N = |S|$
- Space complexity: $O(h)$, $h = \text{height of tree}$
From search trees to search graphs

- Any two nodes that root identical subtrees (subgraphs) can be merged
AND/OR search tree
(Belief Updating)

<table>
<thead>
<tr>
<th>E</th>
<th>A</th>
<th>B</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.4</td>
<td>.6</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.7</td>
<td>.3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.2</td>
<td>.8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>C</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.2</td>
<td>.8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.1</td>
<td>.9</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.7</td>
<td>.3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.6</td>
<td>.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Result: \( P(D=1, E=0) \)

Value of node = updated belief for sub-problem below

Evidence: E=0

Evidence: D=1

OR node: Marginalization operator (summation)
AND node: Combination operator (product)
**AND/OR search graph**

(BELief Updating)

\[ P(E \mid A, B) \quad P(B \mid A) \quad P(C \mid A) \quad P(A) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>E=0</th>
<th>E=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.4</td>
<td>.6</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.5</td>
<td>.5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.7</td>
<td>.3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.2</td>
<td>.8</td>
</tr>
</tbody>
</table>

**Evidence:** E=0

\[
\text{Result: } P(D=1, E=0)
\]

**Cache table for D**

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.9</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.1</td>
</tr>
</tbody>
</table>

\[ P(D \mid B, C) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>D=0</th>
<th>D=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.2</td>
<td>.8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.1</td>
<td>.9</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.3</td>
<td>.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.5</td>
<td>.5</td>
</tr>
</tbody>
</table>

**Evidence:** D=1
The impact of the pseudo-tree

W=4, h=8

What is a good pseudo-tree? How to find a good one?

Min-Fill
(Kjaerulff90)

Hypergraph Partitioning
(h-Metis)

BGU 2011

(W=5, h=6)
AND/OR context-minimal graph

Variable Elimination

AND/OR Search
### Complexity of AND/OR graph search

<table>
<thead>
<tr>
<th></th>
<th>AND/OR graph</th>
<th>OR graph</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space</strong></td>
<td>$O(n , d^{w^*})$</td>
<td>$O(n , d^{pw^*})$</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>$O(n , d^{w^*})$</td>
<td>$O(n , d^{pw^*})$</td>
</tr>
</tbody>
</table>

- $d$ = domain size
- $n$ = number of variables
- $w^*$ = treewidth
- $pw^*$ = pathwidth

$w^* \leq pw^* \leq w^* \log n$
**AND/OR tree vs graph sampling**

Complexity:

\[
\text{AND/OR tree sample tree: } O(nN)
\]

\[
\text{AND/OR graph sample graph: } O(nwN)
\]

where \( w: \text{treewidth} \)
AND/OR Graph sampling

AND/OR sample tree
8 samples

AND/OR sample graph
12 samples
OR tree vs. A/O tree vs A/O-graph

- **AND/OR tree**
  - Time Complexity: $O(nN)$
  - Space Complexity: $O(h)$
  - **AOTM lower Variance than OR IS**

- **AND/OR Graph**
  - Time Complexity: $O(nwN)$
  - Space Complexity: $O(nN)$
  - **AOGM has Lower Variance than AOT**

- **Something in between Cache on “i” variables**
  - Time Complexity: $O(niN)$
  - Space Complexity: $O(nN)$
  - Variance in between.
**Experiments**

(a) Algorithms evaluated on benchmarks without Strong Deterministic Relationships

(b) Algorithms evaluated on benchmarks with Strong Deterministic Relationships

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Benchmarks</th>
<th>Algorithms</th>
<th>Benchmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>or-tree-IJGP-IS-hmetis</td>
<td>Alarm</td>
<td>or-tree-IJGP-SS-hmetis</td>
<td>Coding</td>
</tr>
<tr>
<td>ao-tree-IJGP-IS-hmetis</td>
<td>Grids</td>
<td>ao-tree-IJGP-SS-hmetis</td>
<td>Linkage</td>
</tr>
<tr>
<td>ao-graph-IJGP-IS-hmetis</td>
<td></td>
<td>ao-tree-IJGP-SS-hmetis</td>
<td>Graph Coloring</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>or-tree-IJGP-IS-minfill</td>
<td></td>
<td>or-tree-IJGP-SS-minfill</td>
<td></td>
</tr>
<tr>
<td>ao-tree-IJGP-IS-minfill</td>
<td></td>
<td>ao-tree-IJGP-SS-minfill</td>
<td></td>
</tr>
<tr>
<td>ao-graph-IJGP-IS-minfill</td>
<td></td>
<td>ao-graph-IJGP-SS-minfill</td>
<td></td>
</tr>
</tbody>
</table>

**Hypergraph decomposition**

**minfill**
AND/OR sampling: Some experimental results

Linkage analysis instance having ~500K variables

Error vs. Time in seconds

- SampleSearch
- AND/OR SampleSearch

(UAI, 2008; AIJ, 2011)
Grid instances

Figure 12: Log-relative error versus time plots for the two largest Grid instances with Deterministic ratio = 50%.

Figure 13: Log-relative error versus time plots for the two largest Grid instances with Deterministic ratio = 75%.
Figure 16: Log-relative error versus time plots for four sample linkage instances from the UAI 2006 evaluation.
Figure 17: Log-relative error versus time plots for four sample linkage instances from the UAI 2008 evaluation.
Outline

- Background: Bayesian networks
- Importance Sampling
- AND/OR search space for graphical Models
- AND/OR Sampling: exploiting structure
- **AND/OR+w-cutset sampling**
- UAI 2010 experience

HUJI Dec 2011
Rao-Blackwellised w-cutset sampling

• Rao-Blackwellisation
  – Partition X into two sets K and R, such that we can compute $Z(R \mid k)$ efficiently.

\[
\tilde{Z} = \frac{1}{N} \sum_{i=1}^{N} \sum_{R} Z(R \mid k_i) Q(k_i)
\]

• Smaller variance

• w-cutset sampling: Select K such that the treewidth of R is bounded by “w” (Bidyuk and Dechter, 2007)
Algorithm AND/OR w-cutset sampling

Given an integer constant w

1. **Partition variables** into K and R, such that treewidth of R is <= w.

2. **AND/OR sampling on K**
   1. Construct a pseudo-tree of K and compute Q(K) consistent with K
   2. Generate samples from Q(K) and store them on an AND/OR tree

3. **Rao-Blackwellisation (Exact inference) at each leaf**
   1. For each leaf node of the tree compute $Z(R|g)$ where g is the assignment from the leaf to the root.

4. **Value computation**: Recursively from the leaves to the root
   1. At each AND node compute product of values at children
   2. At each OR node compute a weighted average over the values at children

5. Return the value of the root node
AND/OR w-cutset sampling:
Step 1: Partition the set of variables

Graphical model
AND/OR w-cutset sampling: Step 2: AND/OR sampling over \{A,B,C\}

Graphical model

Chain start pseudo-tree over \{A,B,C\}
Sample A, B and C without taking into account conditional independence

Pseudo-tree

A start pseudo-tree over \{A,B,C\} that takes into account conditional independence properties
AND/OR w-cutset sampling:
Step 2: AND/OR sampling over \{A,B,C\}

Samples: (C=0,A=0,B=1), (C=0,A=1,B=1), (C=1,A=0,B=0), (C=1,A=1,B=0)
AND/OR w-cutset sampling:

**Step 4: Value computation**

Value of $A$ given $C = 0$:
$$= \hat{E}[A, D, E \mid C = 0]$$

Value of $B$ given $C = 0$:
$$v(B = 1 \mid C = 0) = \sum_{f \in F} \sum_{g \in G} H(C = 0, g)H(B = 1, g)H(g, f)H(B = 1, f)$$

Value of $C$: Estimate of the partition function

$$= \hat{E}[B, F, G \mid C = 0]$$
Variance Hierarchy and Complexity

- IS
- O(nN)
- O(1)
- O(nN)
- O(h)
- O(nN)
- O(h)
- O(nN_t)
- O(nN)
- O(nN_t exp(w))
- O(nN + nexp(w))
- AND/OR Tree
- AND/OR w-cutset Tree
- AND/OR w-cutset Graph
- AND/OR Graph

t: the treewidth
w: a forced conditioned
Tree-width

HUJI Dec 2011
Experiments

• Benchmarks
  – Linkage analysis
  – Graph coloring

• Algorithms
  – OR tree sampling
  – AND/OR tree sampling
  – AND/OR graph sampling
  – w-cutset versions of the three schemes above

HUJI Dec 2011
## Results: Probability of Evidence

Linkage instances (UAI 2006 evaluation)

<table>
<thead>
<tr>
<th>Problem</th>
<th>( \langle n, k, E, t^*, c \rangle )</th>
<th>Exact</th>
<th>or-tree-IS ( \Delta )</th>
<th>ao-tree-IS ( \Delta )</th>
<th>ao-graph-IS ( \Delta )</th>
<th>or-wc-tree-IS ( \Delta )</th>
<th>ao-wc-tree-IS ( \Delta )</th>
<th>ao-wc-graph-IS ( \Delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN_69.uai</td>
<td>( \langle 777, 7, 78, 47, 59 \rangle )</td>
<td>5.28E-54</td>
<td>2.26E-02</td>
<td>2.46E-02</td>
<td>2.43E-02</td>
<td>2.42E-02</td>
<td>2.34E-02</td>
<td>4.22E-03</td>
</tr>
<tr>
<td>BN_70.uai</td>
<td>( \langle 2315, 5, 159, 87, 98 \rangle )</td>
<td>2.00E-71</td>
<td>6.32E-02</td>
<td>7.25E-02</td>
<td>5.12E-02</td>
<td>8.18E-02</td>
<td>5.36E-02</td>
<td>2.62E-02</td>
</tr>
<tr>
<td>BN_71.uai</td>
<td>( \langle 1740, 6, 202, 70, 139 \rangle )</td>
<td>5.12E-111</td>
<td>6.74E-02</td>
<td>5.51E-02</td>
<td>2.35E-02</td>
<td>8.58E-02</td>
<td>9.46E-03</td>
<td>1.21E-02</td>
</tr>
<tr>
<td>BN_72.uai</td>
<td>( \langle 2155, 6, 252, 86, 88 \rangle )</td>
<td>4.21E-150</td>
<td>3.19E-02</td>
<td>4.61E-02</td>
<td>2.46E-03</td>
<td>6.12E-02</td>
<td>1.41E-03</td>
<td>2.63E-03</td>
</tr>
<tr>
<td>BN_73.uai</td>
<td>( \langle 2140, 5, 216, 101, 149 \rangle )</td>
<td>2.26E-113</td>
<td>1.18E-01</td>
<td>1.12E-01</td>
<td>4.55E-02</td>
<td>1.58E-01</td>
<td>3.54E-02</td>
<td>3.95E-02</td>
</tr>
<tr>
<td>BN_74.uai</td>
<td>( \langle 749, 6, 66, 45, 72 \rangle )</td>
<td>3.75E-45</td>
<td>5.34E-02</td>
<td>4.31E-02</td>
<td>2.87E-02</td>
<td>8.08E-02</td>
<td>2.83E-02</td>
<td>2.76E-02</td>
</tr>
<tr>
<td>BN_75.uai</td>
<td>( \langle 1820, 5, 155, 92, 131 \rangle )</td>
<td>5.88E-91</td>
<td>4.47E-02</td>
<td>8.15E-02</td>
<td>4.73E-02</td>
<td>7.28E-02</td>
<td>4.20E-02</td>
<td>7.60E-03</td>
</tr>
<tr>
<td>BN_76.uai</td>
<td>( \langle 2155, 7, 169, 64, 239 \rangle )</td>
<td>4.93E-110</td>
<td>1.07E-01</td>
<td>1.39E-01</td>
<td>6.95E-02</td>
<td>1.13E-01</td>
<td>5.03E-02</td>
<td>2.26E-02</td>
</tr>
<tr>
<td>BN_77.uai</td>
<td>( \langle 1020, 9, 135, 22, 97 \rangle )</td>
<td>6.88E-79</td>
<td>1.06E-01</td>
<td>9.38E-02</td>
<td>8.26E-02</td>
<td>1.24E-01</td>
<td>6.75E-02</td>
<td>3.27E-02</td>
</tr>
</tbody>
</table>

**Time Bound: 1hr**

HUJI Dec 2011
Outline

• Background: Bayesian networks
• Importance Sampling
• AND/OR search space for graphical Models
• AND/OR Sampling: exploiting structure
• AND/OR+w-cutset sampling
• UAI 2010 experience
2010 Approximate Inference Evaluation
(Gal Elidan, Amir Globerson)

Tasks:
- MAR: univariate marginals
- PR: partition function (probability of evidence)
- MPE: most probably explanation

Time frames:
- Lightning speed: 20 seconds
- Coffee break: 20 minutes
- Lunch break: 1 hour (reduced from 2 hours)

Anonymous solvers (on request)
## Network: The Big Picture

<table>
<thead>
<tr>
<th>Time</th>
<th>PR</th>
<th>MAR</th>
<th>MPE</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 sec</td>
<td>204</td>
<td>204</td>
<td>461</td>
<td>869</td>
</tr>
<tr>
<td>1200 sec</td>
<td>204</td>
<td>204</td>
<td>532</td>
<td>940</td>
</tr>
<tr>
<td>3600 sec</td>
<td>204</td>
<td>204</td>
<td>287</td>
<td>695</td>
</tr>
<tr>
<td>Total</td>
<td>612</td>
<td>612</td>
<td>1280</td>
<td>2504</td>
</tr>
</tbody>
</table>
## Networks – by domain (1 hour)

<table>
<thead>
<tr>
<th>Network</th>
<th>PR</th>
<th>MAR</th>
<th>MPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSP</td>
<td>8</td>
<td>8</td>
<td>55</td>
</tr>
<tr>
<td>Grids</td>
<td>20</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Image Alignment</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Medical Diagnosis</td>
<td>26</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Object Detection</td>
<td>96</td>
<td>96</td>
<td>92</td>
</tr>
<tr>
<td>Pedigree</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Protein Folding</td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>Protein-Protein Interaction</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Segmentation</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Many thanks to: Kristian Kersting, Stephen Gould, Menachem Fromer, Ben Packer, Dan Geiger, Ariel Jaimovich, Farshid Moussavi

HUJI Dec 2011
<table>
<thead>
<tr>
<th>Seconds</th>
<th>PR</th>
<th>MAR</th>
<th>MPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Arthur Choi (UCLA)</td>
<td>Arthur Choi (UCLA)</td>
<td>Joris Mooij (Max Planck)</td>
</tr>
<tr>
<td>1200</td>
<td>Vibhav Gogate (UW+UCI)</td>
<td>Vibhav Gogate (UCI)</td>
<td>Thomas Schiex (INRA)</td>
</tr>
<tr>
<td>3600</td>
<td>Vibhav Gogate (UW+UCI)</td>
<td>Vibhav Gogate (UCI)</td>
<td>Joris Mooij (Max Planck)</td>
</tr>
</tbody>
</table>
Solver for PR (Gogate 2010)

0. Choose a good variable ordering (30 sec or random minfill)
1. IJGP-based IS
2. AOT w-cutset
3. SampleSearch (handles zeros)
4. Formula-based sampling (handles context specific independence, helped on ising and promedas only)
5. Rejection control and epsilon-correction (Replace very small probabilities (e.g., 0.0001) in the proposal distribution by epsilon (0.01), and normalize. Note that the proposal distribution is a Bayesian network.)
2 Solvers for MAR

**Sampling-based MAR algorithm:**
Step 1: Select values of w and i based on the problem and time-bound
Step 2: Select an ordering of variables by running min-fill 100 times or for 30 seconds whichever is earlier
Step 3: Select a w-cutset
Step 4: Run IJGP with i-bound=i
Step 5: If the problem has context specific independence and determinism
   - Run formula-based SampleSearch with proposal constructed from the output of IJGP
   Else:
   - Run IJGP-sampling

**Anytime IJGP algorithm:**
Step 1: Select an initial value of i based on the problem and time-bound
Step 2: Select an ordering of variables by running min-fill 100 times or for 30 seconds whichever is earlier
Step 3: For j = i to treewidth
   - Run IJGP with i-bound=j until 20 iterations or convergence
   Output the result
UAI 2010 approximate inference challenge:
Comparison on the hardest instances (20 mins)

Cumulative Error

Instances

Radboud Univ. team
UCLA team
UW/UCI team

UW/UCI team was substantially better on the hardest instances

Vibhav Gogate: Scaling up Probabilistic Inference
Winning Teams

- (MAR) **IJGP** by Vibahv Gogate, Andrew Gefland, Natasha Flerova and Rina Dechter (UCI): *Anytime* iterative GBP based algorithm
- (PR) **Vgogate** by Vibahv Gogate, Pedro Domingos (UW), Andrew Gefland and Rina Dechter (UCI): Formula based importance sampling
- (PR+MAR) **EDBP** by Arthur Choi, Adnan Darwiche, with support from Glen Lenker and Knot Pipatsrisawat (UCLA): *Anytime* BP based anytime thickening of structure
- (MAP) **libDAI** by Joris Mooij (Max Planck): junction tree, LBP/MP, double-loop GBP, Gibbs, decimation
- (MAP) **toulbar2** by Thomas Schiex et al (INRA) *Anytime* branch and bound weighted CSP solver
Software and UAI-2008-10 results

• AND/OR search algorithms
• Bucket-tree elimination
• Generalized belief propagation
• Samplesearch sampling

are available at:

– http://graphmod.ics.uci.edu/group
Conclusions

• Exploit structure


- Bozhena Bidyuk, Rina Dechter and Emma Rollon. "Active Tuples-based Scheme for Bounding Posterior Beliefs" Journal of Artificial Intelligence Research (JAIR), (2010) 1-38

- V. Gogate and R. Dechter “Importance Sampling based Estimation over AND/OR Search Spaces for Graphical Models” accepted to Artificial Intelligence, Forthcoming.
Thank you