Anytime Probabilistic Reasoning
(Solving the Marginal MAP)

Rina Dechter
Information and Computer Sciences
UC-Irvine
Outline

- Graphical models, the Marginal Map task, anytime reasoning
- Inference and variational bounds
- AND/OR search spaces
- Combining methods: Heuristic Search
- Combining methods: Sampling
- Conclusion
Overview: Graphical Models

Bayesian Networks

Markov Logic

Deep Boltzmann Machines

Influence Diagrams
Bayesian Networks (Pearl 1988)

\[
P(S, C, B, X, D) = P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B)
\]

Queries: prediction, diagnosis, classification, decision making

**CPD:**

| C | B | P(D|C,B) |
|---|---|---------|
| 0 | 0 | 0.1 0.9 |
| 0 | 1 | 0.7 0.3 |
| 1 | 0 | 0.8 0.2 |
| 1 | 1 | 0.9 0.1 |

\[
P(D=0) = \sum_{S,L,B,X} P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B)
\]

MAP/MPE = \[\text{max}_{S,L,B,X} P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B)\]

P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?

UAI 2019
A graphical model consists of:

\[ X = \{X_1, \ldots, X_n\} \quad \text{-- variables} \]
\[ D = \{D_1, \ldots, D_n\} \quad \text{-- domains} \]
\[ F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\} \quad \text{-- (non-negative) functions or “factors”} \]

\[ P(S, C, B, X, D) = P(S) \ P(C|S) \ P(B|S) \ P(X|C,S) \ P(D|C,B) \]
## Types of Queries

### Tasks:

- **NP-hard**: exponentially many terms
- We will focus on **approximation** algorithms
  - **Anytime**: very fast & very approximate! Slower & more accurate

<table>
<thead>
<tr>
<th>Inference Type</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max-Inference</strong></td>
<td>$f(x^*) = \max_x \prod_{\alpha} f_\alpha(x_\alpha)$</td>
</tr>
<tr>
<td>(most likely config.)</td>
<td></td>
</tr>
<tr>
<td><strong>Sum-Inference</strong></td>
<td>$Z = \sum_x \prod_{\alpha} f_\alpha(x_\alpha)$</td>
</tr>
<tr>
<td>(data likelihood)</td>
<td></td>
</tr>
<tr>
<td><strong>Mixed-Inference</strong></td>
<td>$f(x^*<em>M) = \max</em>{x_M} \sum_{x_S} \prod_{\alpha} f_\alpha(x_\alpha)$</td>
</tr>
<tr>
<td>(optimal prediction)</td>
<td></td>
</tr>
</tbody>
</table>
Why Marginal MAP?

- Often, Marginal MAP is the “right” task:
  - We have a model describing a large system
  - We care about predicting the state of some part

- Example: decision making
  - Sum over random variables (random effects, etc.)
  - Max over decision variables (specify action policies)

Sensor network

Influence diagram:

- Complexity: NP^{PP} complete
- Not necessarily easy on trees
Marginal Map

- **Graphical Model:** $\mathcal{M} = \langle \textbf{X}, \textbf{D}, \textbf{F} \rangle$

  - **variables** $\textbf{X} = \{X_1, \ldots, X_n\}$
  - **domains** $\textbf{D} = \{D_1, \ldots, D_n\}$
  - **functions** $\textbf{F} = \{f_1, \ldots, f_r\}$

  $$P(\textbf{X}) = \frac{1}{Z} \prod_j f_j$$

- **Marginal MAP task:**

  $$\textbf{X} = \textbf{X}_M \cup \textbf{X}_S$$

  $$x_M^* = \arg\max_{x_M} \sum_{x_S} \prod_j f_j$$

  $\textbf{X}_M = \{A, B, C, D\}$
  $\textbf{X}_S = \{E, F, G, H\}$

Why is it harder? intuitively

UAI 2019
Goal: Anytime Bounds

- Desiderata
  - Meaningful confidence interval
  - Responsive
  - Complete
Building Blocks of Approximate Inference

- Three building blocks paradigms
  - Effective at different types of problems

Variational methods
Reason over small subsets of variables at a time

(Monte Carlo) Sampling
Use randomization to estimate averages over the state space

(Heuristic) Search
Structured enumeration over all possible states
Combining Approaches

**Inference, Variational methods**

Bucket-elimination weighted mini-bucket (WMB)
[Dechter 1999, Dechter and Rish, 2003
Liu and Ihler, ICML 2011]

provide heuristic

provide proposal

WMB-IS
[Liu et al., NIPS 2015]

**Search**

AND/OR search
[Marinescu et al 2009, Lou et al., AAAI 2017]
Marinescu et al., IJCAI 2018]

refine proposal

help search

**Sampling**

dynamic importance sampling (DIS)
[Lou et al., NIPS 2017]
Outline

• Graphical models, The Marginal Map task
• Inference and variational bounds.
• AND/OR search spaces
• Combining methods: Heuristic Search for Marginal Map
• Combining methods: Sampling
• Conclusion
Bucket Elimination

Algorithm \textit{BE-bel} [Dechter 1996]

\[
p(A \mid E = 0) = \alpha \sum_{e,d,c,b} p(A) p(b \mid A) p(c \mid A) p(d \mid A, b) p(e \mid b, c) \mathbb{1}[e = 0]
\]

\[
\sum_b \prod \text{Elimination & combination operators}
\]

- **Bucket B:**
  \[p(b \mid A) \ p(d \mid b, A) \ p(e \mid b, c)\]

- **Bucket C:**
  \[p(c \mid A) \ \lambda_{B \to C}(A, d, c, e)\]

- **Bucket D:**
  \[\lambda_{C \to D}(A, d, e)\]

- **Bucket E:**
  \[\mathbb{1}[E = 0] \ \lambda_{D \to E}(A, e)\]

- **Bucket A:**
  \[p(A) \ \lambda_{E \to A}(A)\]

\[p(A \mid E = 0) = \frac{p(A, E = 0)}{p(E = 0)}\]

\(W^* = 4\)

“induced width” (max clique size)

\[\text{UAI 2019}\]
Bucket Elimination

Algorithm *BE-bel* [Dechter 1996]

\[
p(A | E = 0) = \alpha \sum_{e,d,c,b} p(A) p(b | A) p(c | A) p(d | A, b) p(e | b, c) \mathbb{1}[e = 0]
\]

\[\sum_{b} \prod \text{Elimination & combination operators}\]

Time and space exponential in the induced-width / treewidth

\[p(A | E = 0) = p(A, E = 0) / p(E = 0)\]

UAI 2019
Bucket Elimination

Algorithm BE-map [Dechter 1996]

\[ MPE = \max_{a,e,d,c,b} P(a)P(c \mid a)P(b \mid a)P(d \mid a,b)P(e \mid b,c) \]

\[ \max_{X} \prod \]

Time and space exponential in the induced-width / treewidth

bucket A: \( p(A) \)

\( \lambda_{E \rightarrow A}(A) \)

Opt

\( \text{OPT} \)

Induced width (max clique size)
Why is MMAP Harder for Inference (BE)?

Let’s apply Bucket-elimination: Complexity is exponential in the *constrained* induced-width

\[ \mathbf{X}_M = \{A, D, E\} \]
\[ \mathbf{X}_S = \{B, C\} \]

\[ \max_{\mathbf{X}_M} \sum_{\mathbf{X}_S} P(\mathbf{X}) \]

\[ P(\mathbf{X}) = \prod_j f_j \]

MAP* is the marginal MAP value
Why is MMAP Harder for Inference (BE)?

\[ X_M = \{A, D, E\} \]
\[ X_S = \{B, C\} \]

In practice, constrained induced is much larger!

\[ w^* = 4 \]
\[ \lambda(A, C, D, E) \]

\[ w^* = 2 \]
\[ \lambda(B, C) \]

(Park & Darwiche, 2003)
(Yuan & Hansen, 2009)
Decomposition-bounds:
Mini-bucket and weighted Mini-bucket
Tightening by Cost-shifting
Mini-Bucket Approximation

Split a bucket into mini-buckets → bound complexity

\[ \text{bucket} (X) = \left\{ f_1, f_2, \ldots, f_r, f_{r+1}, \ldots, f_n \right\} \]

\[ \lambda_X (\cdot) = \max_x \prod_{i=1}^{n} f_i(x, \ldots) \]

\[ \lambda_{X,1}(\cdot) = \max_x \prod_{i=1}^{r} f_i(x, \ldots) \]

\[ \lambda_{X,2}(\cdot) = \max_x \prod_{i=r+1}^{n} f_i(x, \ldots) \]

\[ \lambda_X (\cdot) \leq \lambda_{X,1}(\cdot) \lambda_{X,2}(\cdot) \]

Exponential complexity decrease: \( O(e^n) \longrightarrow O(e^r) + O(e^{n-r}) \)
Mini-Bucket Elimination

[Dechter & Rish 2003]

For optimization

\[ \lambda_{B \rightarrow C}(a, c) = \max_b f(a, b) \ f(b, c) \]
\[ \lambda_{B \rightarrow D}(d, e) = \max_b f(b, d) \ f(b, e) \]
\[ \lambda_{C \rightarrow E}(a, e) = \max_c \ldots \]

\[ U = \text{upper bound} \]
Mini-Bucket Decoding

\[ b^* = \arg \max_b f(a^*, b) \cdot f(b, c^*) \cdot f(b, d^*) \cdot f(b, e^*) \]

\[ c^* = \arg \max_c f(c, a^*) \cdot f(c, e^*) \cdot \lambda_{B \rightarrow C}(a^*, c) \]

\[ d^* = \arg \max_d f(a^*, d) \cdot \lambda_{B \rightarrow D}(d, e^*) \]

\[ e^* = \arg \max_e \lambda_{C \rightarrow E}(a^*, e) \cdot \lambda_{D \rightarrow E}(a^*, e) \]

\[ a^* = \arg \max_a f(a) \cdot \lambda_{E \rightarrow A}(a) \]

**Greedy configuration = lower bound**

For optimization

\[ B: \begin{array}{cc} f(a, b) & f(b, c) \\ f(b, d) & f(b, e) \end{array} \]

\[ C: \begin{array}{cc} f(c, a) & f(c, e) \end{array} \lambda_{B \rightarrow C}(a, c) \]

\[ D: \begin{array}{cc} f(a, d) \end{array} \lambda_{B \rightarrow D}(d, e) \]

\[ E: \begin{array}{cc} \lambda_{C \rightarrow E}(a, e) & \lambda_{E \rightarrow E}(a, e) \end{array} \]

\[ A: \begin{array}{cc} f(a) \end{array} \lambda_{E \rightarrow A}(a) \]

\[ U = \text{upper bound} \]
Bucket and Mini-Bucket Elimination

- **Input**: I – max number of variables allowed in a mini-bucket
- **Output**: [Lower bound (P of suboptimal solution), upper bound]

**Example: MBE-mpe(3)** versus **BE-mpe**

<table>
<thead>
<tr>
<th>mini-bucket</th>
<th>B:</th>
<th>C:</th>
<th>D:</th>
<th>E:</th>
<th>A:</th>
</tr>
</thead>
<tbody>
<tr>
<td>B:</td>
<td>(f(a, b)) (f(b, c)) (f(b, d)) (f(b, e))</td>
<td>(\lambda_{B\rightarrow C}(a, c)) (f(c, a)) (f(c, e))</td>
<td>(f(a, d)) (\lambda_{B\rightarrow D}(d, e))</td>
<td>(\lambda_{C\rightarrow E}(a, e)) (\lambda_{D\rightarrow E}(a, e))</td>
<td>(f(a)) (\lambda_{E\rightarrow A}(a))</td>
</tr>
</tbody>
</table>

\[U = \text{upper bound}\]

\[w^* = 2\]  

**BE-mpe**

<table>
<thead>
<tr>
<th>mini-bucket</th>
<th>B:</th>
<th>C:</th>
<th>D:</th>
<th>E:</th>
<th>A:</th>
</tr>
</thead>
<tbody>
<tr>
<td>B:</td>
<td>(f(a, b)) (f(b, c)) (f(b, d)) (f(b, e))</td>
<td>(\lambda_{B\rightarrow C}(a, c, d, e)) (f(c, a)) (f(c, e))</td>
<td>(f(a, d)) (\lambda_{C\rightarrow D}(a, d, e))</td>
<td>(\lambda_{D\rightarrow E}(a, e))</td>
<td>(\lambda_{E\rightarrow A}(a))</td>
</tr>
</tbody>
</table>

\[\text{OPT}\]

\[w^* = 4\]  

[Dechter and Rish, 1997]
Properties of Mini-Bucket Elimination

- Bounding from above and below
- Complexity: $O(r \exp(i))$ time and $O(\exp(i))$ space.
- Accuracy: determined by Upper/Lower bound.
- As $i$ increases, both accuracy and complexity increase.
- Possible use of mini-bucket approximations:
  - As anytime algorithms
  - As heuristics in search

Relaxation upper bound by mini-bucket

Consistent solutions (greedy search)
Tightening the Bound

\[ \log f(x^*) = \max_x \sum_{\alpha} \theta_\alpha(x_\alpha) \leq \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{x_\alpha} \left[ \theta_\alpha(x_\alpha) + \sum_{i \in \alpha} \lambda_{i\to\alpha}(x_i) \right] \]

- **Bound solution using decomposed optimization**
  - Solve independently: optimistic bound

- **Tighten the bound by re-parameterization**
  - Enforces lost equality constraints using Lagrange multipliers

Add factors that “adjust” each local term, but cancel out in total

Reparameterization:
\[ \forall j : \sum_{\alpha \ni j} \lambda_{j\to\alpha}(x_j) = 0 \]
Add factors that “adjust” each local term, but cancel out in total.

Reparameterization:
\[ \forall j : \sum_{\alpha \ni j} \lambda_{j \to \alpha}(x_j) = 0 \]

\[
\log f(x^*) = \max_x \sum_{\alpha} \theta_\alpha(x_\alpha) \leq \min_{\{\lambda_{i \to \alpha}\}} \sum_{\alpha} \max_{x_\alpha} \left[ \theta_\alpha(x_\alpha) + \sum_{i \in \alpha} \lambda_{i \to \alpha}(x_i) \right]
\]

- Many names for the same class of bounds
  - Dual decomposition [Komodakis et al. 2007]
  - TRW, MPLP [Wainwright et al. 2005; Globerson & Jaakkola 2007]
  - Soft arc consistency [Cooper & Schiex 2004]
  - Max-sum diffusion [Warner 2007]
Anytime Approximation

- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly
Anytime Approximation

- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly
Anytime Approximation

- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly
WMB for Marginal MAP

\[ \sum_x f_1(x) \cdot f_2(x) \leq \left[ \sum_x f_1(x) \frac{1}{w_1} \right]^{w_1} \cdot \left[ \sum_x f_2(x) \frac{1}{w_2} \right]^{w_2} \]

\[ w_1 + w_2 = 1 \]

\[ \lambda_{B \rightarrow C}(a, c) = \sum_b f(a, b)f(b, c) \]

\[ \lambda_{B \rightarrow D}(d, e) = \sum_b f(b, d)f(b, e) \]

\[ \vdots \]

\[ \lambda_{E \rightarrow A}(a) = \max_e \lambda_{C \rightarrow E}(a, e)\lambda_{D \rightarrow E}(a, e) \]

\[ U = \max_a f(a)\lambda_{E \rightarrow A}(a) \]

Marginal MAP

\[ \Sigma_B \]

bucket B: \[ f(a, b) \quad f(b, c) \]

\[ \Sigma_C \]

bucket C: \[ \lambda_{B \rightarrow C}(a, c) \quad f(a, c) \quad f(c, e) \]

\[ \Sigma_D \]

bucket D: \[ f(a, d) \quad \lambda_{B \rightarrow D}(d, e) \]

\[ \Sigma_E \]

bucket E: \[ \lambda_{C \rightarrow E}(a, e) \quad \lambda_{D \rightarrow E}(a, e) \]

\[ \max_A \]

bucket A: \[ f(a) \quad \lambda_{E \rightarrow A}(a) \]

Can optimize over cost-shifting and weights (single pass “MM” or iterative message passing)

\[ U = upper \ bound \]

[Dechter and Rish, 2003]

[Dechter and Rish, 2003] UAI 2019
**Bucket Elimination (BE) and WMB**

Bounding from above and below

Relaxation upper bound by mini-bucket

Pros:
- Computationally bounded
- Gives upper or lower bound
- Cost-shifting Message passing
- Improves bound

Cons:
- Not anytime!
- Not asymptotically tight without more memory

Consistent solutions (greedy search)
Outline

• Graphical models: definition, examples, methodology
• Inference and variational bounds
• AND/OR search spaces
• Variational bounds as search heuristics
• Combining methods: Heuristic Search for Marginal Map
• Conclusion
**Potential Search Spaces**

- **Full OR search tree**: 126 nodes
- **Context minimal OR search graph**: 28 nodes
- **Full AND/OR search tree**: 54 AND nodes
- **Context minimal AND/OR search graph**: 18 AND nodes

Computes any query:
- Constraint satisfaction
- Optimization (MAP)
- Marginal (P(e))
- **Marginal map**

*Dechter & Mateescu, 2007*

Any query is best computed Over the c-minimal AO search space
AND/OR Tree: An Alternative Model

Cost of the solution tree: the product of weights on its arcs

Cost of \( (A=0, B=1, C=1, D=1, E=0) \) = \( 0.6 \cdot 0.6 \cdot 0.5 \cdot 0.8 \cdot 0.5 = 0.0720 \)
Value of a Node (e.g., Probability of Evidence)

- **AND node**: product
- **OR node**: Marginalization by summation

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

**AND** node: product

**OR** node: Marginalization by summation

---

### Evidence: D=0

<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

<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>.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

<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=1

<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>.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>.7</td>
<td>.3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.6</td>
<td>.4</td>
</tr>
</tbody>
</table>

---

### Calculation of P(D=1, E=0)

\[
P(D=1, E=0) = \prod_{n' \in \text{children}(n)} v(n')
\]

### Summation over children

\[
\sum_{n \in \text{children}(n)} w(n, n') v(n')
\]
AND/OR Search and Variable Elimination

Related to sum-product Networks or Arithmetic circuit
AND/OR Search for Marginal MAP

MAP variables
SUM variables

Node types
- OR (MAP): max
- OR (SUM): sum
- AND: multiplication

primal
\[ X_M = \{A, B, C, D\} \]
\[ X_S = \{E, F, G, H\} \]
AND/OR Search for Marginal MAP

For MMAP search space is:

\[ k^{hc} \] on a AND/OR tree

\[ k^{wc} \] on AND/OR graph

C-induced-width = 6
Induced-width = 2
Outline

- Graphical models, The Marginal Map task
- Inference and variational bounds
- AND/OR search spaces
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion
Search Aided by Variational Heuristics

Given a partial assignment, \( [\hat{a} = 1, \hat{e} = 0] \)
(weighted) mini-bucket gives an admissible heuristic:

\[
\begin{align*}
\tilde{h}(\hat{a}, \hat{e}, D) &= \lambda_{C \rightarrow E}(\hat{a}, \hat{e}) \\
&\quad + f(\hat{a}, D) + \lambda_{B \rightarrow D}(D, \hat{e}) \\
\end{align*}
\]

“cost so far”:
\[
g(\hat{a}, \hat{e}, D) = f(A = \hat{a})
\]

For MAP, marginal map and partition function
AND/OR Search for Marginal MAP

MAP variables

SUM variables

constrained pseudo tree

primal graph

W-mini-buckets

\[ f(a, b) f(b, c) f(b, d) \]

\[ \lambda_{B \rightarrow C}(a, c) f(a, c) f(c, e) \]

\[ f(a, d) \lambda_{B \rightarrow D}(d, e) \]

\[ \lambda_{C \rightarrow E}(a, e) \lambda_{D \rightarrow E}(a, e) \]

\[ f(a) \lambda_{E \rightarrow A}(a) \]
Exact MMAP Solvers: Best or Depth-First Search?

Depth-First search

Best-First search

Lower bound

The MAP search space

Best-first search is superior

Expanding fewer full MAP Solutions, thus less conditional sums

[Marinescu, Dechter, Ihler, AAAI 2014]
Anytime Solvers for Marginal MAP


- **Weighted Best-First search:**
  - Weighted Restarting AOBF (WAOBF)
  - Weighted Restarting RBFAOO (WRBFAOO)
  - Weighted Repairing AOBF (WRAOBF)

- **Interleaving Best-first and depth-first search:**
  - Look-ahead (LAOBF),
  - alternating (AAOBF)

---

**Weighted A* search** [Pohl 1970]
- non-admissible heuristic
- Evaluation function:
  \[ f(n) = g(n) + w \cdot h(n) \]
- Guaranteed w-optimal solution, cost \( C \leq w \cdot C^* \)

---

- Better guidance for depth-first dives using improved heuristics
- Memory robust best-first search using improved lower bounds
LAOBF (Best-first AND/OR Search with Depth-First lookaheads)

Best-first selection

Depth-first lookahead

Best-first expansion & update

$T_b$

Select($T_b$)

- depth-first dive at the tip of $T_b$
- compute global lower bound
- cache summation subproblems

Select a tip node $n$

Expand and Update $n$

Update($n$)

Expand($n$)

Cutoff parameter: perform depth-first dive at every $\theta$ number of node expansions.

Best partial solution tree: $T_b$
AAOBF (Alternating Best-First and Depth-First)

Depth-first greedy expansion
Best-first re-direct $T_b$

Depth-first re-direct $T_l$
Best-first expansion & update

Select($T_l$)
Select($T_b$)
Expand($n$)
Update($n$)
Select($T_l$)
Select($T_b$)
### Benchmarks and Evaluation Methods

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>#. inst</th>
<th>$n$</th>
<th>$k$</th>
<th>$w_c$</th>
<th>$h_c$</th>
<th>$w_u$</th>
<th>$h_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>grid</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>easy</td>
<td>15</td>
<td>144 – 1156</td>
<td>2 – 2</td>
<td>16 – 52</td>
<td>50 – 164</td>
<td>15 – 49</td>
<td>48 – 198</td>
</tr>
<tr>
<td>hard</td>
<td>60</td>
<td>144 – 1156</td>
<td>2 – 2</td>
<td>25 – 375</td>
<td>42 – 421</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>pedigree</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hard</td>
<td>40</td>
<td>334 – 1289</td>
<td>4 – 7</td>
<td>35 – 237</td>
<td>63 – 259</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>promedas</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hard</td>
<td>40</td>
<td>453 – 1849</td>
<td>2 – 2</td>
<td>11 – 490</td>
<td>36 – 507</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 1: Benchmark instances. #. inst is the number of instances in each domain. We also distinguish easy and hard instances. The minimum and the maximum values from the set of problems are shown in the following parameters: $n$ is the number of variables, $k$ is the maximum domain size, $w_c$ is the constrained induced width, $h_c$ is the height of the pseudo tree corresponding to the constrained elimination ordering. The unconstrained induced width, $w_u$ and pseudo tree height, $h_u$ are also shown to highlight the difficulty of hard Marginal MAP problem instances.
Anytime Bounds of Marginal MAP

- Search: LAOBF, AAOBF, BRAOBB, WAOBF, WAOBF-rep
- Heuristic: WMB-MM (20)
- Memory: 24 GB

- Anytime lower and upper bounds from hard problem instances with i-bound 12 (left) and 18 (right).

- The horizontal axis is the CPU time in log scale and the vertical axis is the value of marginal MAP in log scale.

- Benchmarks: Pedigrees, promedas, grids, planning. A fraction of variables selected as MAP (10% hard instances).
Anytime Bounds of Marginal MAP

i-bound = 12

Hard Grid 90-30-5 (N=900, F=900, K=2, S=3, Wc=246), i-bound=12

i-bound = 18

Hard Grid 90-30-5 (N=900, F=900, K=2, S=3, Wc=246), i-bound=18

UAI 2019
Average Gap Quality

But, schemes are limited to tractable conditioned-summation.
Outline

• Graphical models, The Marginal Map task
• Inference and variational bounds
• AND/OR search spaces
• Combining methods: Heuristic Search for Marginal Map
• Combining methods: Sampling
• Conclusion
Importance Sampling

- Basic empirical estimate of probability:

\[
\mathbb{E}[u(x)] = \int p(x)u(x) \approx \hat{u} = \frac{1}{m} \sum_{i} u(\tilde{x}^{(i)}) \quad \tilde{x}^{(i)} \sim p(x)
\]

- Importance sampling:

\[
\int p(x)u(x) = \int q(x) \frac{p(x)}{q(x)} u(x) \approx \frac{1}{m} \sum_{i} \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})} u(\tilde{x}^{(i)}) \quad \tilde{x}^{(i)} \sim q(x)
\]

![Diagram showing importance weights](image)
Choosing a Proposal - WMB-IS

Pr \[|\hat{Z} - Z| > \epsilon \] \leq 1 - \delta

\[ \epsilon = \sqrt{\frac{2V \log(4/\delta)}{m}} + \frac{7U \log(4/\delta)}{3(m-1)} \]

- Can use WMB upper bound to define a proposal \( q_{\text{wmb}}(x) \)

\[ \tilde{b} \sim w_1 q_1(b|\tilde{a}, \tilde{c}) + w_2 q_2(b|\tilde{d}, \tilde{e}) \]

**Weighted mixture:**
- Use minibucket 1 with probability \( w_1 \)
- Or, minibucket 2 with probability \( w_2 = 1 - w_1 \)

where

\[ q_1(b|a, c) = \left[ \frac{f(a, b) \cdot f(b, c)}{\lambda_{B \rightarrow C}(a, c)} \right] \frac{1}{w_1} \]

\[ \vdots \]

\[ \tilde{a} \sim q(A) = f(a) \cdot \lambda_{E \rightarrow A}(a)/U \]

**Key insight:** provides bounded importance weights!

\[ 0 \leq f(x)/q_{\text{wmb}}(x) \leq U \quad \forall x \]
Probabilistic Lower Bounds for MMAP

ANYSBFS: Anytime Stochastic Best-First Search
ANYLDFS: Anytime Learning Depth-First Search
[Marinsecu, Dechter, Ihler IJCAI 2018]

WMB-IS [Liu et. Al. 2015]

Compute a (probabilistic) lower bound on the conditioned sum subproblem

$$Pr(\hat{Z} - \Delta(n, \delta) \leq Z) \geq (1 - \delta)$$

WMB based importance sampling scheme:

- $n$ - number of samples
- $\delta$ - confidence value
- $Z_{wmb}$ - result of WMB
- $\hat{Z}$ - Importance Sampling estimate

$$\Delta(n, \delta) = \sqrt{\frac{2\text{var}(w(x)) \log(2/\delta)}{n}} + \frac{7Z_{wmb} \log(2/\delta)}{3(n-1)}$$

Solving the conditioned SUM subproblem is hard!

$\#P - \text{complete}$
Stochastic Anytime Search for MMAP (Grids)

ANYSBFS: Anytime Stochastic Best-First Search
ANYLDFS: Anytime Learning Depth-First Search

\[ \text{ACC}_{lb} = \frac{|l_t - l^*|}{l^*} \]

\[ \text{ACC}_{ub} = \frac{|u_t - u^*|}{u^*} \]

grid: relative accuracy (lower bounds)

grid: relative accuracy (upper bounds)

\( l_t \) — lower bound at time \( t \)
\( l^* \) — tightest lower bound found

Average over 150 instances

\( u_t \) — upper bound at time \( t \)
\( u^* \) — tightest upper bound found

Average over 150 instances

(Lower plots are better)
Stochastic Anytime Search for MMAP (Planning)
Summary

Inference, Variational methods

Bucket-elimination weighted mini-bucket (WMB)
[Dechter 1999, Dechter and Rish, 2003 Liu and Ihler, ICML 2011]

provide heuristic

provide proposal
WMB-IS [Liu et al., NIPS 2015]

Search
AND/OR search (AODFS)
[Marinescu et al 2009, Lou et al., AAAI 2017]
Marinescu et al., IJCAI 2018

refine proposal

help search

Sampling
dynamic importance sampling (DIS)
[Lou et al., NIPS 2017]
Dynamic Importance Sampling

(For partition function) [Lou, Dechter, Ihler, NIPS 2017, AAAI 2019]

• Interleave
  – Building search tree (expand Nd nodes)
  – Draw samples given search bound (Nl samples)

• Key insight: proposal changes (improves) with each step
  – Use weighted average: better samples get more weight

\[ \hat{Z} = \frac{\text{HM}(U)}{N} \sum_{i=1}^{N} \frac{\hat{Z}_i}{U_i}, \quad \text{HM}(U) = \left[ \frac{1}{N} \sum_{i=1}^{N} \frac{1}{U_i} \right]^{-1} \]

– Derive corresponding concentration bound on Z

UAI 2019
Aggregated Results

[Lou, Dechter, Ihler, NIPS 2017, AAAI 2019]

Table 1: Mean area between upper and lower bounds of logZ, normalized by WMB-IS, for each benchmark. Smaller numbers indicate better anytime bounds. The best for each benchmark is bolded.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>AOBFS</th>
<th>WMB-IS</th>
<th>DIS ((N_l=1, N_d=1))</th>
<th>DIS ((N_l=1, N_d=10))</th>
<th>two-stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedigree</td>
<td>16.638</td>
<td>1</td>
<td>0.711</td>
<td>0.585</td>
<td>1.321</td>
</tr>
<tr>
<td>protein</td>
<td>1.576</td>
<td>1</td>
<td>0.110</td>
<td>0.095</td>
<td>2.511</td>
</tr>
<tr>
<td>BN</td>
<td>0.233</td>
<td>1</td>
<td>0.340</td>
<td>0.162</td>
<td>0.865</td>
</tr>
</tbody>
</table>

UAI 2019
Anytime Bounds in UAI Evaluations

- 2006 (aolib)
- 2008 (aolib)
- 2011 (dao-opt)
- 2014 (dao-opt) (merlin)

MPE/MAP

MMAP

Software

- **daoopt**
  - [https://github.com/lotten/daoopt](https://github.com/lotten/daoopt)
  (distributed and standalone AOBB solver)

- **merlin**
  (standalone WMB, AOBB, AOBF, RBFAOO solvers)
  open source, BSD license

**pyGM**: Python Toolbox for Graphical Models by Alexander Iher.
Continuing Work

• Combining approaches:
  – Tune the hyper-parameters automatically
  – Extend to decision networks

• Languages and Tools:
  – Relational languages
  – Handle constraints specification and continuous functions
  – Temporal domains; Planning, e.g., Influence diagrams, MDPs, POMDPs
  – Cross interaction of deep learning and graphical models → Reinforcement

See our poster on Thursday
(Lee, Marinescu, Ihler, Dechter)
Thank You!

See our poster on Thursday

Alex Ihler
Kalev Kask
Irina Rish
Bozhena Bidyuk
Robert Mateescu
Radu Marinescu
Vibhav Gogate
Emma Rollon
Lars Otten
Natalia Flerova
Andrew Gelfand
William Lam
Junkyu Lee
Qi Lou