Anytime Probabilistic Reasoning

Rina Dechter
Bren School of Information and Computer Sciences

Main Collaborators:
Alexander Ihler
Kalev Kask
Radu Marinescu
Qi Lou
Junkyu Lee
AI Renaissance

- Deep learning
  - Fast predictions
  - “Instinctive”

Tools:
Tensorflow, PyTorch, ...

- Probabilistic models
  - Slow reasoning
  - “Logical / deliberative”

Tools:
Probabilistic programming, Markov Logic, ...
Overview: Graphical Models

Bayesian Networks

Markov Logic

Deep Boltzmann Machines
Outline

• Overview of problems and methodology
• Main paradigms of approximate reasoning: Variational, Search, Sampling
• Combining approaches
• Future challenges
Graphical Models

- Describe structure in large problems
  - Large complex system $f(X)$
  - Made of “smaller”, “local” interactions $f_{\alpha}(X_{\alpha})$
  - Complexity emerges through interdependence

- More formally:
  A *graphical model* consists of:
  
  \[ X = \{X_1, \ldots, X_n\} \] -- variables (we’ll assume discrete)
  \[ D = \{D_1, \ldots, D_n\} \] -- domains
  \[ F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\} \] -- (non-negative) functions or “factors”

- Example:
  \[ F(A,B,C) = f(A,B) \cdot f(B,C) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>f(A,B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.24</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.56</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1.1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>f(B,C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.12</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.36</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Primal graph
Graphical Models

• Describe structure in large problems
  – Large complex system $f(X)$
  – Made of “smaller”, “local” interactions $f_\alpha(X_\alpha)$
  – Complexity emerges through interdependence

• Examples & Tasks
  – Maximization (MAP): compute the most probable configuration

\[ x^* = \arg \max_x \prod_\alpha f_\alpha(x_\alpha) \quad f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha) \]

[ Yanover & Weiss 2002 ]

[Phenylalanine]
Graphical Models

• Describe structure in large problems
  – Large complex system $f(X)$
  – Made of “smaller”, “local” interactions $f_{\alpha}(X_{\alpha})$
  – Complexity emerges through interdependence

• Examples & Tasks
  – Summation & marginalization

\[ p(x_i) = \frac{1}{Z} \sum_{x \setminus x_i} \prod_{\alpha} f_{\alpha}(x_{\alpha}) \]

and

\[ Z = \sum_{x} \prod_{\alpha} f_{\alpha}(x_{\alpha}) \]

“partition function”

Observation $y$

Marginals $p(x_i \mid y)$

Observation $y$

Marginals $p(x_i \mid y)$

e.g., [Plath et al. 2009]
Graphical Models

- Describe structure in large problems
  - Large complex system $f(X)$
  - Made of “smaller”, “local” interactions $f_\alpha(X_\alpha)$
  - Complexity emerges through interdependence

- Examples & Tasks
  - Mixed inference (marginal MAP, MEU, ...)

\[
  f(x^*_M) = \max_{x_M} \sum_{x_S} \prod_{\alpha} f_\alpha(x_\alpha)
\]

- Influence diagrams & optimal decision-making

  (the “oil wildcatter” problem)

  e.g., [Raiffa 1968; Shachter 1986]
A **graphical model** consists of:

- **Variables** $X = \{X_1, \ldots, X_n\}$
- **Domains** $D = \{D_1, \ldots, D_n\}$
- **Functions** $F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$

**Operators:**
- **Combination operator** (sum, product, join, ...)
- **Elimination operator** (projection, sum, max, min, ...)

**Types of queries:**

<table>
<thead>
<tr>
<th>Inference Type</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max-Inference (MAP)</strong></td>
<td>$f(x^*) = \max_x \prod_{\alpha} f_{\alpha}(x_\alpha)$</td>
</tr>
<tr>
<td><strong>Sum-Inference (P(\epsilon))</strong></td>
<td>$Z = \sum_x \prod_{\alpha} f_{\alpha}(x_\alpha)$</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>
</tbody>
</table>

All these tasks are NP-hard:
- exploit problem structure
- identify special cases
- approximate
Example Domains

• Natural Language processing
  – Information extraction, semantic parsing, translation, topic models, ...

• Computer vision
  – Object recognition, scene analysis, segmentation, tracking, ...

• Computational biology
  – Pedigree analysis, protein folding and binding, sequence matching, ...

• Networks
  – Webpage link analysis, social networks, communications, citations, ....

• Robotics
  – Planning & decision making
Anytime Bounds

• Desiderata
  – Meaningful confidence interval
  – Responsive
  – Complete

\[ f(x^*) = \max_x \prod_{\alpha} f_\alpha(x_\alpha) \]
\[ Z = \sum_x \prod_{\alpha} f_\alpha(x_\alpha) \]
\[ f(x_M^*) = \max_{x_M} \sum_{x_S} \prod_{\alpha} f_\alpha(x_\alpha) \]
Approximate Inference

• Three major 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

• Bounds
• Responsive
• Complete

• Bounds
• Responsive
• Complete

• Bounds
• Responsive
• Complete

UMD 5/3/2019
Combining Approaches

**Inference, Variational methods**

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

**Search**

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

**Sampling**

- Dynamic importance sampling (DIS)
  - [Lou et al., NIPS 2017]

provide heuristic

provide proposal

refine proposal

help search

WMB-IS

[Liu et al., NIPS 2015]
Outline

- Overview of problems and methodology
- Main paradigms of approximate reasoning:
  Variational, Search, Sampling
- Combining approaches
- Future challenges
Query 1: Belief updating: $P(X|\text{evidence})=?$

**Belief updating using Variable Elimination**

**Primal Graph:**

1. **Objective:** Evaluate $P(a|e=0)$ given the primal graph.
2. **Approach:** Use Variable Elimination to compute $P(a,e=0)$.

**Variables and Graph Structure:**
- $A$, $B$, $C$, $D$, $E$
- Edges between nodes represent conditional probabilities.

**Mathematical Derivation:**

$$P(a|e=0) \propto P(a,e=0) = \sum_{e=0,d,c} P(a)P(b|a)P(c|a)P(d|b,a)P(e|b,c)$$

**Variable Elimination Steps:**

1. **Eliminate $c$:**
   $$P(a)\sum_d \sum_c P(c|a) \sum_b P(b|a)P(d|b,a)P(e|b,c)$$

2. **Results:**
   - $h^B(a,d,c,e)$
   - Final expression for $P(a|e=0)$
Marginals by Bucket Elimination
(Dechter 1999)

\[ P(A \mid E = 0) = \alpha \sum_{E=0,D,C,B} P(A) \cdot P(B \mid A) \cdot P(C \mid A) \cdot P(D \mid A,B) \cdot P(E \mid B,C) \]

Elimination operator

bucket B: \( P(b \mid a) \quad P(d \mid b,a) \quad P(e \mid b,c) \)

bucket C: \( P(c \mid a) \quad \lambda_{B\rightarrow C}(a, d, c, e) \)

bucket D: \( \lambda_{C\rightarrow D}(a, d, e) \)

bucket E: \( e=0 \quad \lambda_{D\rightarrow A}(a, e) \quad W^* = 4 \)

bucket A: \( P(a) \quad \lambda_{E\rightarrow A}(a) \quad \text{"induced width" (max clique size)} \)

\[ P(a \mid e=0) \]

Complexity time and space \( O(nk^{W^*+1}) \)
Bucket and Mini-Bucket Elimination

\[ X = \{A, B, C, D, E, F, G\} \]

\[ F(X) = f(A)f(A, B)f(A, D)f(A, G)f(B, C)f(B, D) \]
\[ f(B, E)f(B, F)f(C, D)f(C, E)f(E, G) \]

**Bucket-Elimination (Dechter, 1999)**
Exponential in tree-width \( O(nk^w) \)

**Mini-Bucket Elimination (Dechter & Rish, 2003)**
Exponential in i-bound \( O(nk^i) \)

\[ \lambda^D(A) = \max_D f(A, D) \]
\[ \lambda^D(B, C) = \max_D [f(B, D) f(C, D)] \]

\[ \lambda^D(A, B, C) = \sum_D f(B, D) f(C, D) f(A, D) \]
Bucket and Mini-Bucket Elimination

\[ \sum_X F(X) \]

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

\( \lambda^X(\cdot) \leq \lambda^X,1(\cdot) \lambda^X,2(\cdot) \)

\[ \lambda^D(A) = \sum_D f(A, D) \]

\[ \lambda^D(B,C) = \sum_D f(B, D) f(C, D) \]

\[ \lambda^D(A,B,C) = \sum_D f(B, D) f(C, D) f(A, D) \]

A summation query; e.g., partition function

[Dechter 1999; Dechter & Rish, 2003]
Bucket and Mini-Bucket Elimination

\[ \max_X F(X) \]
\[ F(X) = f(A)f(A, B)f(A, D)f(A, G)f(B, C)f(B, D) \]
\[ f(B, E)f(B, F)f(C, D)f(C, E)f(E, G) \]

**Bucket-Elimination**
Exponential in tree-width \( O(nk^w) \)

**Mini-Bucket Elimination**
Exponential in i-bound \( O(nk^i) \)

A maximization query; e.g., MAP

\[ \lambda^D(A, B, C) = \max_D f(B, D)f(C, D)f(A, D) \]

**mini-buckets**

\[ \lambda^D(A) = \max_D f(A, D) \]

\[ \lambda^D(B, C) = \max_D f(B, D)f(C, D) \]
Bucket and Mini-Bucket Elimination

[Dechter 1999; Dechter & Rish, 2003, Liu & Ihler 2011]

A maximization query; e.g., MAP

\[
\max_X F(X) = f(A) f(A, B) f(A, D) f(A, G) f(B, C) f(B, D) f(B, E) f(B, F) f(C, D) f(C, E) f(E, G)
\]

Bucket-Elimination
Exponential \(O(nk^w)\)

Assigning MAP value, greedily

\[
\begin{align*}
\lambda^D(A, B, C) &= \max_D f(B, D) f(C, D) f(A, D) \\
\lambda^B(A) &= \arg \max f(a) \lambda^B(a) \\
\lambda^C(A, B) &= \lambda^C(A, B) \\
\lambda^E(A, B) &= \lambda^E(A, B) \\
\lambda^F(A, B) &= \lambda^F(A, B) \\
\lambda^G(A, F) &= \lambda^G(A, F) \\
\lambda^D(A, B, C) &= \lambda^D(A, B, C) \\
\lambda^C(A, B) &= \lambda^C(A, B) \\
\lambda^E(A, B) &= \lambda^E(A, B) \\
\lambda^F(A, B) &= \lambda^F(A, B) \\
\lambda^G(A, F) &= \lambda^G(A, F) \\
\end{align*}
\]

\[
\begin{align*}
a^* &= \arg \max_a f(a) \lambda^B(a) \\
b^* &= \arg \max_b f(a, b) \lambda^C(a^*, b) \lambda^F(a^*, b) \\
c^* &= \arg \max_c f(b, c) \lambda^D(a^*, b^*, c) \lambda^E(a^*, c) \\
f^* &= \arg \max_f f(b, f) \lambda^G(a, f) \\
d^* &= \arg \max_c f(a^*, d)f(b^*, d)f(c^*, d) \\
e^* &= \arg \max_e f(b^*, e)f(c^*, e) \\
\end{align*}
\]

\[
\text{return } (a^*, b^*, c^*, d^*, e^*, f^*, g^*), \text{ exact}
\]
Bucket and Mini-Bucket Elimination

\[ \max_X F(X) \]

\[ F(X) = f(A)f(A, B)f(A, D)f(A, G)f(B, C)f(B, D)f(B, E)f(B, F)f(C, D)f(C, E)f(E, G) \]

\[ a^* = \arg\max_a f(a) \lambda^B(a) \lambda^D(a) \]

\[ b^* = \arg\max_b f(a, b) \lambda^C(a^*, B) \lambda^C(a^*, B) \]

\[ c^* = \arg\max_c f(b, c) \lambda^D(b^*, C) \lambda^E(b^*, C) \]

\[ d^* = \arg\max_c f(a^*, d)f(b^*, d)f(c^*, d) \]

\[ e^* = \arg\max_e f(b^*, e)f(c^*, e) \]

returns \( F(a^*, b^*, c^*, d^*, e^*, f^*, g^*) \) a lower bound
Properties of Bucket Elimination and WMB

- Bounding from above and below
- Complexity: $O(r \exp(i))$ time and $O(\exp(i))$ space.
- Accuracy: determined by Upper/Lower bound.
- As $i$-bound increases, both accuracy and complexity increase.
- Message passing tightens bounds (next slides).

Relaxation upper bound by mini-bucket

Consistent solutions (greedy search)
Tightening the Bound: Weighted Mini-Bucket (WMB)

[Dechter 2003, Liu & Ihler 2011]

Bounds can be tightened by optimizing weights.

\[ F(X) = f(A)f(A, B)f(A, D)f(A, G)f(B, C)f(B, D)f(B, E)f(B, F)f(C, D)f(C, E)f(E, G) \]

- Holder inequality \[ w_1 + w_2 = 1 \]

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

\[
\lambda^D(A) = \left[ \sum_D f(A, D)^{\frac{1}{w}} \right]^{w_1}
\]

\[
\lambda^D(B,C) = \left[ \sum_D f(B, D)f(C, D)^{\frac{1}{w}} \right]^{w_2}
\]

\[ (\lambda^D(A) = \sum_D f(A, D) ) \]

\[ (\lambda^D(B,C) = \sum_D f(B, D) f(C,D)) \]
Tightening the Bound: Reparameterize Functions

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

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

- Bound solution using decomposed optimization
- Solve independently: optimistic bound
- Tighten the bound by re-parameterization
  - Enforces lost equality constraints using Lagrange multipliers

Many names for the same class of bounds:
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
Properties of Bucket Elimination and WMB

- Bounding from above and below
- Complexity: $O(r \exp(i))$ time and $O(\exp(i))$ space.
- Accuracy: determined by Upper/Lower bound.
- As $i$-bound increases, both accuracy and complexity increase.
- Message passing tightens bounds.
- But: Not anytime! not asymp. tight w/o more memory

Relaxation upper bound by mini-bucket

Consistent solutions (greedy search)
Outline

• Overview of problems and methodology
• Main paradigms of approximate reasoning: Variational, Search, Sampling
• Combining approaches
• Future challenges
## Potential search spaces

<table>
<thead>
<tr>
<th>Context minimal OR search graph</th>
<th>28 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context minimal AND/OR search graph</td>
<td>18 AND nodes</td>
</tr>
</tbody>
</table>

### OR tree
- **time**: $O(k^n)$
- **memory**: $O(n)$

### AND/OR tree
- **time**: $O(nk^h)$
- **memory**: $O(n)$

### OR graph
- **time**: $O(nk^{pw*})$
- **memory**: $O(nk^{pw*})$

### AND/OR graph
- **time**: $O(nk^{w*})$
- **memory**: $O(nk^{w*})$

### Expressions
- $f(x^*) = \max_x \prod_{\alpha} f_{\alpha}(x_{\alpha})$
- $Z = \sum_x \prod_{\alpha} f_{\alpha}(x_{\alpha})$
- $f(x_M) = \max_{x_M} \sum_{\alpha} \prod_{\alpha} f_{\alpha}(x_{\alpha})$

### Search Tree
- **Full OR search tree**: 126 nodes
- **Full AND/OR search tree**: 54 AND nodes
- **OR tree**: 0 nodes
- **AND/OR tree**: 1 nodes
- **OR graph**: 0 nodes
- **AND/OR graph**: 1 nodes

Any query can be computed over any of the search spaces.
Cost of a Solution Tree

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)

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

**AND node**: product

**OR node**: Marginalization by summation

\[
\prod_{n' \in \text{children}(n)} v(n')
\]

\[
\sum_{n \in \text{children}(n)} w(n, n') v(n')
\]
Answering Queries: Sum-Product (Belief Updating)

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

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

Evidence: \( E=0 \)

\[ P(B|A) \]

\[
\begin{array}{c|cc}
A & B=0 \ B=1 \\
0 & .4 \ .6 \\
1 & .1 \ .9 \\
\end{array}
\]

\[ P(C|A) \]

\[
\begin{array}{c|cc}
A & C=0 \ C=1 \\
0 & .2 \ .8 \\
1 & .7 \ .3 \\
\end{array}
\]

\[ P(A) \]

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

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

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

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

Cache table for D

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?

W=5, h=6

(C K H A B E J L N O D P M F G)
Combining Approaches: Search + Variational

Variational methods

Search

provide heuristics

WMB

Sampling

dynamic importance sampling (DIS)

For MAP, marginal map and partition function

WMB-IS

[Liu et al., NIPS 2015]

[Lou et al., NIPS 2017]
Given a partial assignment, \([\hat{a} = 1, \hat{e} = 0]\) (weighted) mini-bucket gives an admissible heuristic:

"cost to go":
\[
\tilde{h}(\hat{a}, \hat{e}, D) = \lambda_{C\rightarrow E}(\hat{a}, \hat{e}) + f(\hat{a}, D) + \lambda_{B\rightarrow D}(D, \hat{e})
\]

"cost so far":
\[
g(\hat{a}, \hat{e}, D) = f(A = \hat{a})
\]
MBE Heuristic Guides AO Search

\[ f(T') = w(A,0) + w(B,1) + w(C,0) + w(D,0) + h(D,0) + h(F) = 12 \leq f^*(T') \]

h(n) ≤ v(n)

\[ L = \text{lower bound} \]
Exploiting Heuristic Search Principles

- **Weighted Heuristic:** [Pohl 1970]

\[ f(n) = g(n) + w \cdot h(n) \]

- Guaranteed w-optimal solution, cost \( C \leq w \cdot C^* \)

- **Interleaving Best + Depth-First search**

Goal: anytime bounds
And anytime solution

MMAP
Anytime Bounds for Optimization

(2001-2017)
• Kask and Dechter. Artificial Intelligence, 2001,
• Otten and Dechter: AI Commun. (2012)
• Flerova, Marinescu and Dechter Artificial Intelligence, 2016.
• Flerova, Marinescu and Dechter, (2016) JAIR", 2016
• Lam, Kask, Larrosa, and Dechter. " JAIR 2017.
• Otten and Dechter (JAIR 2017)
• Marinescu, Lee, Dechter and Ihler. JAIR, 2018.

UAI Competition MPE/MAP

Anytime Bounds for Summation

(AAAI’17, AAAI’18: Lou, Dechter and Ihler)

- Heuristic search for summation
  - Heuristic function upper bounds value (sum below) at any node
  - Expand tree and compute updated bounds, using a priority gap

\[ Z \leq U = h_{00} + h_{010} + h_{011} + h_1 \]
Anytime Behavior of AOBFS

(a) PIC’11/queen5_5_4

(b) Protein/1g6x
Anytime Bounds for Marginal MAP

- Complexity: $\text{NP}^{\text{pp}}$ complete
- Not necessarily easy on trees

Marinescu, Dechter and Ihler, 2014

MAP variables

SUM variables

Constrained pseudo tree
Anytime Solvers for Marginal MAP

- **Weighted Heuristic:** [Lee et. al. AAAI-2016, JAIR 2019]
  - Weighted Restarting AOBF (WAOBF)
  - Weighted Restarting RBFAOO (WRBFAOO)
  - Weighted Repairing AOBF (WRAOBF)

- **Interleaving Best and depth-first search:** (Marinescu et. al AAAI-2017)
  - Look-ahead (LAOBF),
  - alternating (AAOBF)

Exploiting heuristic search ideas

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

Goal: anytime bounds
And anytime solution
Anytime Bounds of Marginal MAP

(UAI’14, IJCAI’15, AAAI’16, AAAI’17, JAIR 2019 (Marinescu, Lee, Ihler, Dechter))

- 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.
Combining Approaches: **Sampling+Search**

**Variational methods**

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

**And/or Search**

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

**Sampling**

For MAP, marginal map and partition function
[Marinescu et al 2009, Lou et al., AAAI 2017, JAIR 2019]
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:

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

“importance weights”

\[ w^{(i)} = \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})} \]
IS on a Bayesian or Markov Network?

- Draw samples from $P[A|E=e]$ directly?
  - Model defines un-normalized $p(A,...,E=e)$
  - Build (oriented) tree decomposition & sample

\[
\begin{align*}
\bar{b} & \sim f(\bar{a}, b) \cdot f(b, \bar{c}) \cdot f(b, \bar{d}) \cdot f(b, \bar{e}) / \lambda_{B \rightarrow C} \\
\bar{c} & \sim f(c, \bar{a}) \cdot f(c, \bar{e}) \cdot \lambda_{B \rightarrow C}(\bar{a}, c, \bar{d}, \bar{e}) / \lambda_{C \rightarrow D} \\
\bar{d} & \sim f(\bar{a}, d) \cdot \lambda_{B \rightarrow D}(d, \bar{e}) / \lambda_{D \rightarrow E}(\bar{a}, \bar{e}) \\
\bar{e} & \sim \lambda_{D \rightarrow E}(\bar{a}, e) / \lambda_{E \rightarrow A}(\bar{a}) \\
\bar{a} & \sim p(A) = f(a) \cdot \lambda_{E \rightarrow A}(a) / Z
\end{align*}
\]

Downward message normalizes bucket; ratio is a conditional distribution.

Can use, WMB, Generalized belief propagation for proposal.
Choose a Proposal Combine w Search

- Cutset Sampling [Bidyuk and Dechter (JAIR, 2007)]
- Sampling [Bidyuk and Dechter (JAIR, 2007)]
- SampleSearch [Gogate and Dechter (Artif Intell, 2011)]
- AND/OR sampling [Gogate and R. Dechter (Artif Intell, 2012)]
- Sampling-based lower bounds [Gogate, Dechter (Intelligenza Artificiale, 2011)]
- Dynamic Importance Sampling (DIS) [Lou, Dechter, and Ihler (NIPS 2017)]
- Abstraction Sampling [Broka, Dechter, Ihler and Kask (UAI, 2018)].
- Finite-sample Bounds for MMAP [Lou, Dechter, and Ihler. (UAI 2018)]
- WMB Importance Sampling (WMB-IS) [Liu, Fisher, Ihler (ICML 2015)]

Building blocks in current algorithms for Markov Logic Networks

- Probabilistic Theorem Proving: Gogate and Domingos, CACM 2016,
Choosing a proposal- WMB-IS

[Liu, Fisher, Ihler 2015]

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

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

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

\[ 0 \leq f(x)/q_{\text{wmb}}(x) \leq U \quad \forall x \]
WMB-IS Bounds

- Finite sample bounds on the average
  \[ \Pr \left[ |\hat{Z} - Z| > \epsilon \right] \leq 1 - \delta \]

- Confidence interval depends on two parts
  - Empirical variance, decreasing as \( 1/m^{1/2} \)
  - Upper bound \( U \), decreasing as \( 1/m \)

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

“Empirical Bernstein” bounds

[Liu, Fisher, Ihler 2015]
Combining Approaches

Variational methods

WMB

provide heuristics

Search

WMB-IS

provide proposal

[Liu et al., NIPS 2015]

refine proposal

Sampling

dynamic importance sampling (DIS)

[Lou et al., NIPS 2017]

For MAP, marginal map
and partition function
Dynamic Importance Sampling

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

- Interleave
  - Building search tree (expand Nd nodes) (For partition function)
  - 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
Two-step Sampling

\[ \propto w_0^A u_0^A \]

\[ \propto w_{01}^{AB} u_{01}^{AB} \]

\[ \propto w_{010}^{ABC} u_{010}^{ABC} \]

\[ \sim q(\cdot | ABC = 010) \]

\[ \propto w_{010}^{ABF} u_{010}^{ABF} \]

\[ \sim q(\cdot | ABF = 010) \]
Finite-sample Bounds for DIS

**Theorem:** Define the deviation term

\[
\Delta = \text{HM}(U) \left( \sqrt{\frac{2 \widehat{\text{Var}}(\{\hat{Z}_i/U_i\}_{i=1}^N)}{N} \ln(2/\delta)} + \frac{7 \ln(2/\delta)}{3(N-1)} \right)
\]

then, \( \Pr[Z \leq \hat{Z} + \Delta] \geq 1 - \delta \) and \( \Pr[Z \geq \hat{Z} - \Delta] \geq 1 - \delta \).

\( \widehat{\text{Var}}(\{\hat{Z}_i/U_i\}_{i=1}^N) \): empirical variance of \( \{\hat{Z}_i/U_i\}_{i=1}^N \).
Individual Results

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

(a) pedigree/pedigree33  (b) protein/lco6  (c) BN/BN_30

(d) pedigree/pedigree37  (e) protein/lbge  (f) BN/BN_129
Outline and Challenges

- Overview of problems and methodology
- Main paradigms of approximate reasoning: Variational, Search, Sampling
- Combining approaches
- Future challenges
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
Thank You!

For publication see:
http://www.ics.uci.edu/~dechter/publications.html