Inference and Search for Discrete Graphical Models;
A tutorial and recent work

Rina Dechter
Bren school of ICS, University of California, Irvine

APPRL SITE-VISIT, April 25
Sample Applications for Graphical Models

Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal models.
Sample Applications for Graphical Models

Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal models.
Outline

- What are graphical models? Queries
- Inference
- Search
- Time vs space, search vs inference
- Bounding inference (Variational: BP, GBP, weighted mini-bucket, cost-shifting)
- Bounding Search (Sampling)
- Anytime algorithms
- Optimization: Tailoring solvers to instance
- Recent algorithmic development
- Summary
Graphical Models

A graphical model \((X,D,F)\):
- \(X = \{X_1, \ldots, X_n\}\) variables
- \(D = \{D_1, \ldots, D_n\}\) domains
- \(F = \{f_1, \ldots, f_r\}\) functions
  (constraints, CPTS, CNFs ...)

Operators:
- combination: Sum, product, join
- Elimination: projection, sum, max/min

Tasks:
- Belief updating: \(\Sigma_{x,y} \prod_j P_j\)
- MPE: \(\max_x \prod_j P_j\)
- CSP: \(\prod_x \times_j C_j\)
- Max-CSP: \(\min_x \Sigma_j F_j\)

All these tasks are NP-hard
- exploit problem structure
- identify special cases
- approximate
Queries

- **Optimization Queries**: MAP/MPE queries:
  \[ x_{AB}^* = \arg \min_{x_A, x_B} \sum_{x_\alpha} \varphi_{\alpha} \quad x_{AB}^* = \arg \max_{x_A, x_B} \prod_{x_\alpha} \varphi_{\alpha} \]

- **Likelihood queries**: (counting, partition function, marginal, probability of evidence)
  \[ Z = \sum_{x_A, x_B} \prod_{x_\alpha} \varphi_{\alpha} \]

- **Marginal MAP**:
  - Marginalize (sum) away variables A, then find optimal configuration of variables B
  \[ \mathbf{x}_B^* = \arg \max_{\mathbf{x}_B} \sum_{\mathbf{x}_A} \prod_{\alpha} \psi(\mathbf{x}_\alpha) \]

*Also satisfiability and expected utility*
Tree-solving is Easy

**Belief updating**
*(sum-prod)*

**MPE (max-prod)**

*Dynamic Programming, Inference*

**CSP – consistency**
*(projection-join)*

*Message-passing*

Trees are processed in linear time and memory

*Message-passing*
Inference vs conditioning-search

**Inference**

\[ \text{exp}(w^*) \text{ time/space} \]

**Search**

\[ \text{Exp}(n) \text{ time} \]
\[ O(n) \text{ space} \]

**Search+inference:**

\[ \text{Space: } \text{exp}(w) \]
\[ \text{Time: } \text{exp}(w+c(w)) \]
\[ w: \text{ user controlled} \]
**Inference vs conditioning-search**

**Inference**

\[ \text{Exp}(w^*) \text{ time/space} \]

**Search**

\[ \text{Exp}(w^*) \text{ time} \]
\[ O(w^*) \text{ space} \]

**Search+inference:**

- **Space:** \( \text{Exp}(q) \)
- **Time:** \( \text{Exp}(q+c(q)) \)

\( q \): user controlled
Inference
Query 1: Belief updating: $P(X|\text{evidence})=?$

$P(a|e=0) \propto P(a,e=0) =$

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

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

Variable Elimination

$h^B(a,d,c,e)$
Query 2: Finding MPE by Bucket Elimination

Algorithm BE-mpe (Dechter 1996, Bertele and Briochi, 1977)

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

\[
\max_X \prod
\]

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

bucket C: \quad P(c \mid a) \quad h_B\rightarrow_C (a, d, c, e)

bucket D: \quad h_C\rightarrow_D (a, d, e)

bucket E: \quad e=0 \quad h_D\rightarrow_E (a, e)

bucket A: \quad P(a) \quad h_E\rightarrow_A (a)

\[W^* = 4\] "induced width" (max clique size)

\text{OPT}
Generating the MPE-tuple

1. $a' = \arg\max_a P(a) \cdot h^E(a)$

2. $e' = 0$

3. $d' = \arg\max_d h^C(a', d, e')$

4. $c' = \arg\max_c P(c | a') \times h^B(a', d', c, e')$

5. $b' = \arg\max_b P(b | a') \times P(d' | b, a') \times P(e' | b, c')$

Return $(a', b', c', d', e')$
Generating the MPE-tuple

1. $a' = \arg \max_a P(a) \cdot h^E(a)$

2. $e' = 0$

3. $c' = \arg \max_{c'} P(c | a')$

4. $d' = \arg \max_{d'} P(d | b, a')$

5. $b' = \arg \max_b P(b | a') \times \prod_b P(d' | b, a') \times P(e' | b, c')$

Time and space exponential in the induced-width / treewidth

$O(n^{k \uparrow w* + 1})$

Return $(a', b', c', d', e')$
Exact Inference solvers at UCI

- BE (Bucket Elimination)
- BEEM BE with External Memory, (UAI 2010)
- IGVO (Iterative Greedy Variable Ordering, AAAI 2011)
Search
OR Search Tree
AND/OR vs. OR Spaces

Time $O(nk^h)$
Space $O(n)$
height is bounded by $(\log n) w^*$
AND/OR Tree DFS Algorithm (Belief Updating)

Evidence: $E=0$

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

OR node: Marginalization by summation

AND node: product

Value of node = updated belief for sub-problem below
AND/OR Tree Search for Optimization

Goal: \( \min_X \sum_{i=1}^{9} f_i(X) \)

AND node = Combination operator (summation)

OR node = Marginalization operator (minimization)
AND/OR Search Graph

Constraint Satisfaction – Counting Solutions

context minimal graph

pseudo tree
All Four Search Spaces

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

Computes any query:
- MAP/MPE
- Likelihood ($p(\text{evidence})$)
- Marginal MAP

Any query is best computed
Over the c-minimal AO space
The impact of the pseudo-tree

- **Optimization**
  - Choose pseudo-tree with a minimal search graph
  - But determinism is unpredictable
  - And pruning by BnB is even more unpredictable

Min-Fill
(Kjaerulff90)
The Effect of Constraint Propagation

Domains are \{1,2,3,4\}

CONSTRAINTS ONLY

FORWARD CHECKING

MAINTAINING ARC CONSISTENCY
Search + Inference
W-Cutset conditioning + inference.

- Inference may require too much memory
- Condition on some of the variables

Time exp in cycle-cutset
Memory-linear

Graph Coloring problem

A=yellow
B=red

A=green
B=red
B=blue

B=blue
B=yellow
Search+Inference:
Trading Space for Time

- **AO(j):** searches depth-first, cache i-context
  - $j = \text{the max scope-size of a cache table.}$

  - **Space:** $O(n)$
  - **Time:** $O(nk^{w*} \log n)$

  - **Space:** $O(nk^{j})$
  - **Time:** $O(nk^{j} + mj)$

  - **Space:** $O(nk^{w*})$
  - **Time:** $O(nk^{w*} + 1)$
Search solvers at UCI

- **MAP solvers (AND/OR Branch and Bound):**
  - AOBB, AOBF(i-bound using MBE), (Marinescu, 2009)
  - BRAOBB(i-bound, MPLP,JGLP), (Otten 2013)
  - Distributed/parallel AOBB (Otten 2013)

- **Likelihood solvers:**
  - VEC(i): Variable elimination and conditioning (Gogate, 2009).
  - Aolib (AND/OR search for likelihood, (Mateescu 2007)

- **Marginal-MAP (new )**
  - AOBB-JG, AOBB-MM: AND/OR search+ weighted mini-bucket and cost-shifting, (Submitted to UAI 2014, Marinescu, Dechter and Ihler)
Approximate Inference
Loopy Belief Propagation

- Belief propagation is exact for poly-trees
- Loopy BP - applying BP iteratively to cyclic networks

No guarantees for convergence
Works well for many coding networks
Iterative Join-graphs Propagation (IJGP: kask, (Dechter and Mateescue, 2003), (GBP: yedidya et. Al., 2002....)

*i-Bounded join-graph*

\[ JG(3) \]

**Join-tree = Tree-Decomposition**

more accuracy

less complexity
Mini-Bucket Elimination

\[ \max_B \Pi \]

Bucket B

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

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

Bucket C

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

Bucket D

\[ h^B (A,D) \]

Bucket E

\[ E = 0 h^C (A,E) \]

Bucket A

\[ P(A) h^E (A) h^D (A) \]

\[ W=2 \]

Node duplication, renaming

\[ P(A) \]

\[ P(B|A) \]

\[ P(C|A) \]

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

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

\[ MPE^* \text{ is an upper bound on } MPE --U \]

\[ \text{Generating a solution yields a lower bound} --L \]
Mini-bucket and mini-clustering

- **Complexity**: $O(r \exp(i))$ time and $O(\exp(i))$ space.
- As $i$ increases, both accuracy and complexity increase.
- Applicable to all queries.
- Weighted mini-bucket for optimization

**Pairwise Model**

- $B_2 : \{\psi_{23}\}, \{\psi_{12}\}$
- $B_3 : \{\psi_{34}, m_{2\to3}(y_3)\}$
- $B_4 : \{\psi_{45}, m_{3\to4}(y_4)\}, \{\psi_{14}\}$
- $B_5 : \{\psi_{15}, m_{4\to5}(y_5)\}$
- $B_1 : \{m_{2\to1}(y_1), m_{4\to1}(y_1), m_{5\to1}(y_1)\}$
Tightening Bounds via cost-shifting

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree
Decomposition view

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree
- Enforce lost equality constraints via Lagrange multipliers

\[
\max_x \sum_{ij} E_{ij}(x_i, x_j) \leq \min_{\lambda} \sum_{ij} \max_x E_{ij}(x_i, x_j) + \lambda_j(x_i) + \lambda_i(x_j)
\]

\[\forall i \sum_j \lambda_{ij}(x_i) = 0\]
Update the original factors (FGLP)

- Tighten all factors over $x_i$ simultaneously
- Compute **max-marginals**
  \[
  \forall \alpha, \quad \gamma_\alpha(x_i) = \max_{x_\alpha \setminus x_i} f_\alpha
  \]
- & update:
  \[
  \forall \alpha, \quad f_\alpha(x_\alpha) \leftarrow f_\alpha(x_\alpha) - \gamma_\alpha(x_i) + \frac{1}{|F_i|} \sum_\beta \gamma_\beta(x_i)
  \]
Join-graph based cost-shifting

(Ihler, Flerova, Dechter, Otten, UAI 2012)

Join Graph

$\max_B \Pi$

$\max_B \Pi$

Bucket B

$P(E|B,C)$

$P(C|A) h^B (C,E)$

$h^B (A,D)$

$h^B (A,E)$

$h^E (A) h^D (A)$

$E = 0$

$h^A (A)$

$P(A)$

$P(D|A,B)$ $P(B|A)$
Bounded Inference solvers at UCI


- Bounding schemes:
  - MB(i-bound),
  - weighted-MB(i) (Ihler, 2012)
  - FGLP (Ihler, 2012)
  - JGLP(i-bound) (Ihler 2012)

- Bounding schemes provide heuristic for AND/OR search
Approximate Search: Sampling stochastic local search
Sampling and Local Search at UCI

- Likelihood queries:
  - W-cutset sampling (Gibbs and importance)
  - SampleSearch (Importance sampling)
  - AND/OR sampling (Importance sampling)
  - Hybrid of all (Importance sampling)

- MAP/MPE
  - GLS+ (Hutter et. al., 2005)
  - STLS (Stochastic tree local search, Milchgrub and Dechter, submitted)
Outline

- What are graphical models? Queries
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference (BP, GBP, mini-bucket variational)
- Bounding Search (Sampling)
- Anytime search algorithms for MAP
- Optimization: Tayloring solver to problem
- UCI Algorithm Library
- Conclusions
MAP by AND/OR Branch-and-Bound

Decomposition of independent subproblems

Prune based on current best solution and heuristic estimate (mini-bucket heuristic).

Cache table for F (independent of A)

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

[Marinescu & Dechter, AIJ'09]
AOBB + Central Enhancements

\[ \min_{\lambda} \sum_{(ij)} \max_{X} f_{ij}(X_i, X_j) + \lambda_{ij}(X_i), \lambda_{ji}(X_j) \]

**Cost-shifting (MPLP) Re-parametrization**
Tighter bounds by iteratively solving linear programming relaxations and message passing on join graph.

**Breadth-First Subproblem Rotation**
Improved anytime performance through interleaved processing of independent subproblems.

**Enhanced Variable Ordering Schemes**
Highly efficient, stochastic minfill / mindegree implementations for lower-width orderings.

+ SLS (Hutter et. Al 2005)

(Ihler, Flerova, Dechter, Otten, 2012, Otten and Dechter 2011, Kask, Gelfand, Otten, Dechter 2010)
This year’s advancements
A New Algorithm for Marginal MAP

- *(Submitted to UAI-2014) Improving Marginal Map for Graphical Models”*
- Radu Marinescu, Rina Dechter, Alex Ihler.

- **Problem:** \( \mathbf{x}^*_B = \arg \max_{\mathbf{x}_B} \sum \prod_{\mathbf{x}_A} \psi(\mathbf{x}_\alpha) \)

Marginalize away variables A, then find optimal configuration of variables B

Figure 2: AND/OR search spaces for marginal MAP

*Improving Marginal Map for Graphical Heuristics generated by weighted mini-bucket and moment-matching heuristics.*
- **Branch and Bound Search of AND/OR search**

Figure 5: Number of instances solved (top) and number of wins (bottom) by benchmark.
**Weighted AND/OR Search**

*Paper submitted to ECAI-2014: “Evaluating Weighted DFS Branch and Bound over Graphical Models” Natalia Flerova, Radu Marinescu, Rina Dechter*

*Empirically evaluation proposed algorithms wAOBB and wBRAOBB against Weighted Best-First search (wAOBB) and Breadth-First AND/OR Branch and Bound (BRAOBB)*
A New Tree-based Stochastic Local search for MAP

STLS: Cutest-driven Local Search for MPE” Alon Milchgrub and Rina Dechter, submitted UAI-2014

Problem: \[ x^* = \arg\min_{x} \sum_{i} \varphi_i(x_i) + \sum_{i<j \in E} \psi_{i,j}(x_i, x_j) \]

Algorithm STLS:
Input: an instance of the energy minimization problem
Output: an assignment to the problem approximating its minimum

1. Initialize the variables according to the current initialization scheme (see 3.2)
2. Find the set of variables \( T \) which are not a part of any cycle using the algorithm described in [15].
3. Repeat until stagnation is declared, once a set number of iterations in which no variable has changed its value have passed (or time bound has passed):
   (a) Find a cycle-cutset \( C \) for \( V \setminus T \).
   (b) Update the values of the variables:
      - Given the values of the cycle-cutset \( C \) inducing a forest \( F = V \setminus C \), use BP to find the optimal assignment on \( F \).
      - Given the values of the tree variables \( T \), use some local search algorithm (i.e. HOPFIELD MODEL, GLS+[6], etc.) in order to update the values \( C \) (optional implementation).

If this last feature is utilized the two update stages are performed alternately until convergence or until the set number of iterations has passed.
Optimization: Tailoring solver to the problem instance
Optimization

- **The problem**: determine how much time will take a solver $A(par)$ to solve a problem instance $c$

- **Approaches**:
  - Worst-case analysis based on the graph-parameters (tree-width), time space tradeoff for exact schemes
  - Learning: over a benchmark class
  - Stratified Sampling: of problem instance for a solver to estimate search space.
Worst-Case Analysis

- **Optimization**
  - Choose pseudo-tree with minimal search graphs
  - But determinism is unpredictable
- And pruning by BnB is unpredictable

Min-Fill (Kjaerulff90)

Hypergraph Partitioning (h-Metis)
Learning a Regression Model for Complexity Estimation (Otten and Dechter, 2012)

- Number of nodes $N(n)$ as linear function of features $\varphi_j(n)$:

$$ \log N(n) = \sum_j \lambda_j \varphi_j(n) $$

related: [Leyton-Brown, Nudelman, Shoham 2009]
Subproblem Features $\varphi_j(n)$

- Use both static and dynamic characteristics:
  - Structural
  - Subproblem bounds
  - Limited AOBB probe.

- Prediction performance, learning per problem instance:

<table>
<thead>
<tr>
<th>Problem</th>
<th>MSE</th>
<th>PCC</th>
<th>TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedreg10, p=1440, fixed=6</td>
<td>0.828</td>
<td>0.825</td>
<td>0.468</td>
</tr>
<tr>
<td>pedreg41, p=1106, fixed=10</td>
<td>0.842</td>
<td>0.833</td>
<td>0.312</td>
</tr>
<tr>
<td>largebam3-11.59, p=396, fixed=9</td>
<td>0.797</td>
<td>0.789</td>
<td>0.199</td>
</tr>
<tr>
<td>pblock5, p=111, fixed=3</td>
<td>0.841</td>
<td>0.831</td>
<td>0.159</td>
</tr>
<tr>
<td>7.26-52, p=240, fixed=3</td>
<td>0.797</td>
<td>0.789</td>
<td>0.199</td>
</tr>
</tbody>
</table>

- Dynamic:
  - Problem solution cost, derived from solution.
  - Problem solution cost, provided by the “upper and lower bound, excluding the heuristic.
  - Probability of determinism (zero probability of pseudo tree leaf.

- Static:
  - Average depth of terminal search nodes within probe.
  - Average node depth within probe (denoted $d$).
  - Average branching degree, defined as $\sqrt{5000}$.

- 34: Max. subproblem variable context size minus mini bucket $i$-bound.
Predicting Depth-First Branch and Bound Search Trees (Levi, Lars and Dechter, IJCAI 2013, CP-2014 submission)

UCI Library: Summary

- **Exact/anytime:**
  - **Likelihood:** BE, BEEM, VEC(w), AOlPe(c-bound)
  - **MAP:** VE, BEEM (external memory/multi-core), AOBb(i), BRAOBb(i), DAOOPT(Distributed AOBb).
  - Marginal Map (currently developed)

- **Approximation/anytime, for all queries:**
  - BP, IJGP(i-bound)
  - IJGP-Importance Sampling(i-bound)
  - IJGP-SampleSearch(i-bound)
  - MBE (mini-bucket), Weighted-mini-bucket, reparameterized MB
  - STLS (currently developed, for MAP)

- **Supporting schemes:** Variable-ordering (IGVO)
Summary

- What are graphical models? Queries
- Inference
- Search
- Time vs space, search vs inference
- Bounding inference (Variational: BP, GBP, weighted mini-bucket, cost-shifting)
- Bounding Search (Sampling)
- Anytime algorithms
- Optimization: Tailoring solvers to instance
- Recent algorithmic development
- Summary

Exact algorithms

Approximations

Anytime

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