From Exact to Anytime Solutions for Marginal MAP

Junkyu Lee, Radu Marinescu, Rina Decther, Alex Ihler
Overview

• Marginal MAP
  – Brief Summary of Past Activities
  – Other Algorithms

• Algorithms
  – Primal Graph
  – AND/OR Search Space
  – Advanced Heuristics
  – List of Algorithms

• Benchmark Result
  – Experiment Design
  – Coverage
  – One to One Score Comparison
  – BRAOB vs. WRBFAOO
  – WRAOB vs. WAOBF
Marginal MAP

• Marginal Maximum A Posteriori Task

• Given discrete multi-variate probability distribution,

\[ P(x_1, x_2, \ldots x_n) \]

• Compute

\[ \max_{x_i \in m} \sum_{x_j \in s} P(x_1, x_2, \ldots x_n) \]

  – m : MAP variable
  – s  : SUM variable
Brief Summary

- **Radu Marinescu, Rina Dechter, and Alexander Ihler**, “AND/OR Search for Marginal MAP Search.” *In UAI 2014*
  - *what heuristic is better for AOBB-MMAP?*
    - WMB-JG with 5~10 iteration on i ~ 10
    - WMB-MM on high i-bounds
    - WMB-JG/WMB-MM is better than relaxed order

- **Radu Marinescu, Rina Dechter, and Alexander Ihler**, "Pushing Forward Marginal MAP with Best-First Search." *In IJCAI 2015*
  - AND/OR search space (static heuristic) vs. OR search space (with static/dynamic heuristic)
    - The gain from AND/OR search space is huge
  - Best First AND/OR Search vs. Depth First AND/OR Search
    - Best First is better in MMAP (in MPE, Depth First was better)
    - Evaluating less number of exact summation subproblem
  - Recursive Best First AND/OR search with overestimation (Linear memory Best First)
    - Avoids memory issue, works like Best-first [expand 10~20% more nodes than best-first]

- **Junkyu Lee, Radu Marinescu, Rina Dechter, and Alexander Ihler**, “From Exact to Anytime Solutions for Marginal MAP.” Submitted to AAAI 2016
  - Best first search scheme is good but it returns solution in the end
  - So, investigate Anytime Best-First AND/OR search (as well as Anytime Depth-First AND/OR)
  - Current slides shows the result;
Other Algorithms

• J. Park and A. Darwiche, “Solving MAP exactly using systematic search” UAI 2003
  – AND/OR Branch & Bound + relaxed ordering heuristic
• C. Yuan and E. Hansen, “Efficient computation of jointree bounds for systematic MAP search.” IJCAI 2009
  – improve [J.Park and A. Darwhiche, 2003]
  – Incremental evaluation of unconstrained junction tree (static heuristic)
• Denis Maua and Cassio De Campos, “Anytime marginal map inference.” ICML 2012
  – Formulate Augmented MPE problem by instantiating MAP assignments
  – Build Junction a Tree, pass K best local messages to neighbors
• Q. Liu and A. Ihler, “Variational algorithms for marginal MAP.” JMLR, 2013
Graphical Model

- **Primal Graph**
  - represent conditional independence as a graph

- **Pseudo Tree**
  - (reverse) elimination order
  - all edges in G are back-arcs
AND/OR Search Space

• capture problem decomposition due to conditioning (shallower search tree)
• align the search with respect to pseudo tree (pseudo tree arrangement)
Conditioning vs. Elimination

Conditioning (search)

Elimination (inference)

Induced Width: Maximum Number of Parents in Induced Graph

Here, eliminating A results in a clique of size 4

A = 1 → A = k

k “sparser” problems

1 “denser” problem
Bucket Elimination

Algorithm elim-opt [Dechter, 1996]
Non-serial Dynamic Programming [Bertele & Briochi, 1973]

\[
\text{OPT} = \min_{a,e,d,c,b} f(a) + f(a,b) + f(a,c) + f(a,d) + f(b,c) + f(b,d) + f(b,e) + f(c,e)
\]

\[
\min_b \sum \text{Elimination/Combination operators}
\]

bucket B: \( f(a,b) \ f(b,c) \ f(b,d) \ f(b,e) \)

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

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

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

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

\[
\text{OPT}
\]

IJCAI 2015
Mini-Bucket Elimination Semantics

$\min_{B'} \sum f(\cdot)$

bucket B:

- $f(a, b')$
- $f(b', c)$

bucket C:

- $\lambda_{B\rightarrow C}(a, c)$
- $f(a, c)$
- $f(c, e)$

bucket D:

- $f(a, d)$
- $\lambda_{B\rightarrow D}(d, e)$

bucket E:

- $\lambda_{C\rightarrow E}(a, e)$
- $\lambda_{D\rightarrow E}(a, e)$

bucket A:

- $f(a)$
- $\lambda_{E\rightarrow A}(a)$

$L = \text{lower bound}$

IJCAI 2015
Advanced Heuristics

• Weighted Mini-Bucket Elimination

• Join Graph Cost Shifting Scheme
List of Algorithms

• Exact Search
  – Depth First AND/OR Branch & Bound (AOBB)
  – Best First AND/OR (AOBF)
  – Recursive Best First AND/OR with Overestimation (RBFAOO)

• Anytime Search
  – Breath Rotate AOBB (BRAOBB)
  – restarting weightedAOBF (WAOBF)
  – reaparing weighted AOBF (WRAOBF)
  – restarting weighted RBFAOO (WRBFAOO)

• Heuristic
  – Weighted Mini-Bucket Elim(i) (WMB(i))
  – Weighted Mini-Bucket Elim(i) + Moment Matching (WMB-MM(i))
  – Weighted Mini-Bucket Elim(i) + JG(n) (WMB-JG(i, n))
weighted best first search

- Inflate heuristic function by some weight $w > 1$
  - do best first search with inadmissible heuristic
  - $w$-optimality
    - suboptimal solution is within the factor of $w$

- anytime best first search algorithm
  - start from high weight and decrease it
  - $64 \rightarrow 8 \rightarrow 2.82 \rightarrow 1.68 \rightarrow 1.29 \rightarrow 1.13 \rightarrow \ldots \rightarrow 1.0$

- restarting: discard explicited search space

- repairing: repair explicited search space with more accurate heuristic
Benchmark

• Domains
  – GRID (15 problems, 5 instances per 1 problem)
  – PEDIGREE (10 problems, 5 instances per 1 problem)
  – PROMEDAS (10 problems, 5 instances per 1 problem)

• Time Limit
  – 2 hour

• Memory Limit
  – 24 GB
  – Cache: Best First Searches 4 GB
Benchmark Instances

3 domains total 175 instances
Percentage of Instances Solved

• WMB-MM(i) (strength of heuristics)
  – 6, 12, 18, 20

• Exact Algorithms
  – AOBB, AOBF, RBFAOO

• Anytime Algorithms
  – WAOBF, WRAOBF, WRBFAOO, BRAOBB
$i=6$
i=18
i=20
Summary

• Percent of Instances Reported a Solution
  – Increases as i-bound increases
  
  – anytime coverage (covered if a solution reported)
    of anytime best-first search is close to BRAOBB
  
  – coverage decreased at i=20
    • static heuristic table size was too big
    • percent of optimally solved instances increased
Score of algorithm against others

• i bound (strength of heuristics)
  – 4, 6, 12, 18

• time bound
  – 1 sec, 30 sec, 20 min, 1hr

• Score = \( \frac{\text{win} + \frac{\text{tie}}{2}}{\text{win} + \text{tie} + \text{lose}} \)

• WRBFAO0, WAOBF, WRAOBF, BRAOBB, AOB0B
i=6, t= [1 sec, 30 sec]
\[ i=6, \ t = [20 \text{ min}, \ 1\text{hr}] \]
$i=12$, $t=[1 \text{ sec}, 30 \text{ sec}]$
i=12, t=[20 min, 1 hr]
$i=18, \ t=[1 \ sec, \ 30 \ sec]$
$i=18, \ t=[20 \text{ min}, 1 \text{ hr}]$
Summary

• $i=6$
  
  – BRAOBB > WAOBF > WRBFAOO > WRAOBF > AOBB

• $i=12, 18$
  
  – 1 sec
    • BRAOBB > WAOBF > WRAOBF > AOBB > WRBFAOO
  
  – 30 sec, 20 min, 1 hr
    • WRBFAOO > WAOBF > WRAOBF > BRAOBB > AOBB
BRAOBB vs WRBFAOO

• i bound (strength of heuristics)
  – 6, 12, 18

• time bound
  – 1 sec, 30 sec, 20 min, 1hr
i=6, t=[1 sec, 30 sec, 20 min, 1 hr]
i=12, t=[1 sec, 30 sec, 20 min, 1 hr]
Anytime Weighted BF

• i bound (strength of heuristics)
  – 4, 6, 12, 18

• w-optimality
  – weighted best-first search
Weight $i=4, 6$
Weight $i=12, 18$
AND MAP Nodes \(i=4, 6\)
AND MAP Nodes i=12, 18