A Weighted Mini-Bucket Bound for Solving Influence Diagrams

Junky Lee1, Radu Marinescu2, Alexander Ihler1, Rina Dechter1
1University of California, Irvine  {junkyul, ihler, dechter}@ics.uci.edu  2IBM Research, Ireland  {radu.marinescu}@ie.ibm.com

Summary

A bounding scheme that interleaves variable elimination and
reparameterization on valuations for influence diagrams (IDs) over the weighted mini-bucket decompositions of influence diagram.

It improves the quality of the bound and computation time compared with state-of-the-art decomposition bounds of IDs, and generate admissible heuristic evaluation functions suitable for AND/OR graph search.

Backgrounds

Valuation for Influence Diagrams

- Chance variables C = \{S_1, S_2, S_3, S_4\}
- Decision variables D = \{D_1, D_2\}
- Probability functions P = \{P(S_1), \ldots, P(S_4)\}
- Utility functions U = \{U(D_1), \ldots, U(D_2)\}
- Partial ordering constraint C = \{S_1, S_2, S_3, S_4\} < \{D_1, D_2\}
- Policy functions \( \delta_{S_1} = S_1 \rightarrow D_1 \rightarrow S_2 \rightarrow D_2 \)

MEU = \( \sum_{S_1, S_2} \max_{S_3} \sum_{S_4} \max_{V} \Pi U(D_1) \Pi \Pi U(D_2) \Pi U(C) \)

Valuation for IDs:

\[ \psi(X, Y) = \psi(X) \downarrow \psi(Y) \] (probability, expected utility value)

Combination:

\[ \psi_{1+2} = \psi_{1}(X) \psi_{2}(Y) \]

Marginalization:

\[ \phi = \psi \downarrow \psi(X) \downarrow \psi(Y) \]

Valuations for IDs form a commutative semi-ring

- Axiomatization of valuation algebra ensures decomposition by re-arranging combinations and marginalizations (distributive law).
- Local computation is implemented by Bucket Elimination. [Dechter, 1992]

Re-writing MEU Query by Valuation Algebra for IDs:

\[ \psi = \{(P, 0) | P \in \{1, U\} \} \cup \{1, U\} \] (probability utility value)

MEU = \( \sum_{S_1, S_2} \max_{S_3} \sum_{S_4} \max_{V} \Pi U(D_1) \Pi \Pi U(D_2) \Pi U(C) \)

Graphical Model Decomposition

- The exact decomposition captures dynamic programming structure.
- The approximate decomposition introduces auxiliary variables and bounds the complexity.
- Messages can be passed over a decomposed join-graph to generate upper bounds.

A Weighted Mini-bucket Bound for IDs

- Previous Work: Decomposition Bounds for IDs [Lee, H. Dachter, 2016]
  - When marginalizing X from the combination of valuations, exchange combination and marginalization.
  \[ \psi_{1+2} = \psi_1(X) \psi_2(Y) \]
  \[ \phi_1 = \psi_1(X) \psi_2(Y) \]
  \[ \phi_2 = \psi_1(X) \psi_2(Y) \]
  \[ \psi_{1+2} = \phi_1 \downarrow \phi_2 \]
  - The powered-sum operator
  \[ \sum_{\psi_{X_1} \otimes \psi_{X_2}} \phi_{X_1} \otimes \phi_{X_2} \leq \sum_{\psi_{X_1} \otimes \psi_{X_2}} \phi_1 \otimes \phi_2 \]
  - The exact decomposition captures dynamic programming structure.
  - \( \psi_{MEU_{Bounds}} \) is a surrogate upper bound.

Optimization Setup for Computing Weights and Costs

Objective: value component of \( \psi_{MEU_{Bounds}} \)

Parameters:

\[ \frac{w_1}{w_2 + w_3} \]

Constraints:

\[ w_1 \leq 1 \]

\[ \frac{w_1}{w_2 + w_3} \leq 1 \]

\[ w_1 \geq \frac{w_2}{w_3 + w_1} \geq 0 \]

\[ \frac{w_1}{w_2 + w_3} \geq 0 \]

Algorithm

Initialization:

- Loop: Create a mini-bucket tree and join-graph, allocate functions to mini-buckets and initialize weights uniformly.
- Loop: MBE-WCC, optimize weights w.r.t. the initial function allocations (optional).

Interleaving VE and optimization:

- Loop: Process one layer of mini-buckets at a time.
- Loop: Optimize weights over the mini-bucket for relaxation limits.
- Loop: Generate messages by marginalization and pass messages downward.
- Loop: Combine messages and weights before eliminating the next layer.

Algorithm configurations:

- MBE: no optimization, fix weights.
- MBE-WCC: optimize weights, uniform weights.
- MBE-WC: optimize cost functions only.
- MBE-WC-I: optimize both step initial weight optimization step over the join graph.
- MBE-WC-II: optimize both and perform optional weight optimization step over the join graph.

Experiments

Comparing Decomposition Bounds for IDs

- Compared 4 different configurations of the WMBE-ID, state-of-the-art JGDID, and MBE, MBE with relaxed ordering.
- Round: 15, 10, 15, 10, 20
- Non-laxative algorithms (WMBE-ID, MBE, MBE-Re) produced very loose bounds.
- At lower bounds, JGDID performs the best.
- At higher bounds, WMBE-WCC produced tighter bounds (ratio less than 1.0) than JGDID, JGDID in most of the instances.

WMBE-ID vs. JGDID (I=1)

- Compared the ratio of the time and bound from WMBE-WCC against JGDID.
- At high bounds, WMBE-WCC produced tighter bounds (ratio less than 1.0) than JGDID, JGDID in most of the instances.

This work supported in part by NSF grants IIS-1526642 and IIS-1254871, the U.S. Air Force (Contract FA9550-10-C-0050), and DARPA (contract W31P4Q-10-C-0015).