

Submodel Decomposition for Solving Limited Memory Influence Diagrams (Student Abstract)

Junkyu Lee

Donald Bren School of Information and Computer Sciences
 University of California, Irvine
 6210 Donald Bren Hall, Irvine, CA, 92697
 junkyul@uci.edu, +1-949-241-3783

Abstract

This paper presents a systematic way of decomposing a limited memory influence diagram (LIMID) to a tree of single-stage decision problems, or submodels and solving it by message passing. The relevance in LIMIDs is formalized by the notion of the partial evaluation of the maximum expected utility, and the graph separation criteria for identifying submodels follow. The submodel decomposition provides a graphical model approach for updating the beliefs and propagating the conditional expected utilities for solving LIMIDs with the worst-case complexity bounded by the maximum treewidth of the individual submodels.

Introduction

The sequential decision making under uncertainty is an important class of problems that asks for the optimal strategy of a rational agent that interacts with a system in sequence by intervening in the system dynamics and observing the partial state to achieve the maximum total expected utility. The original influence diagram (Howard and Matheson 2005) was a compact representation for the decision tree that assumed no-forgetting agent, and a linear order on decisions. (Lauritzen and Nilsson 2001) introduced a limited memory influence diagram (LIMID) that generalizes IDs by introducing the notion of relevance and dropping the perfect recall assumption. From the game-theoretical perspective, a LIMID represents a single-player game under imperfect information and imperfect recall by using a Bayesian network for updating the beliefs while searching for the multi-selves equilibrium. Recent approaches for solving IDs are relying on valuation algebra (Jensen, Jensen, and Dittmer 1994; Mauá, de Campos, and Zaffalon 2012). However, (Dechter 2000; Pralet, Schiex, and Verfaillie 2006) showed that the constrained induced width could be tightened by exploiting the algebraic structure. For solving LIMIDs, (Lauritzen and Nilsson 2001) proposed a single policy update algorithm that solves a subclass of LIMIDs called soluble LIMIDs. (Mauá, de Campos, and Zaffalon 2012) presented the optimal algorithm for solving LIMIDs by using the valuation algebra.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Limited Memory Influence Diagrams

A LIMID \mathcal{M} consists of the chance variables \mathbf{X}_C associated with the conditional probability functions $\{P_{C_i}|C_i \in \mathbf{X}_C\}$, the decision variables \mathbf{X}_D associated with the policy functions $\{\Delta_{D_i}|D_i \in \mathbf{X}_D\}$, and a set of utility functions \mathbf{U} defined over the chance or decision variables. A directed acyclic graph \mathcal{G} of \mathcal{M} represents the functions by directed arcs, where the parents of a node X denoted by $\text{pa}(X)$ encode conditioning variables for the probability functions or scope of the utility functions. The $\text{pa}(D)$ for a decision variable D defines the observed random variables for making decision D , hence it encodes a precedence relation denoted by $\mathbf{O}_D \prec \{D\}$. Graphically, chance variables are drawn as circles, decision variables are squares, and the utility functions are diamonds as shown in Figure 1. The task of solving LIMIDs is computing the maximum expected utility, $\text{MEU} := \max_{\Delta} \mathbb{E}[\sum_{U_i \in \mathbf{U}} U_i]$, and finding the optimal strategy Δ^* achieving the MEU under the imperfect recall.

Relevance and Graph Separation in LIMIDs

Let's define the partial evaluation of \mathcal{M} with respect to a pair (D, U) of a decision variable and a utility function as a locally maximum expected value of U obtained by perturbing $\Delta(D|\mathbf{O}_D)$, i.e., $\max_{\Delta'_D} \mathbb{E}[U|\Delta \setminus \{\Delta_D\} \cup \{\Delta'_D\}]$. Then the relevant utility functions for a decision D denoted by $\text{REL}_U(D)$, the relevant observed variables for a pair of a decision D and a relevant utility U denoted by $\text{REL}_O(D, U)$, and the relevant hidden variables denoted by $\text{REL}_H(D, U)$ can be defined as a subset of the variables and functions of \mathcal{M} that induces any change in the locally optimal policy Δ^*_D or

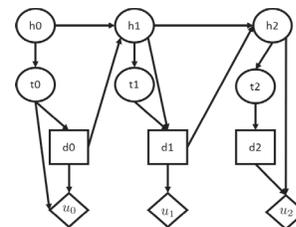


Figure 1: A 3 stage LIMID.

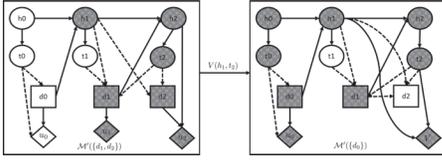


Figure 2: Submodel decomposition. Shaded nodes are relevant sets and dashed arrows are the partial history in memory.

the partial evaluation. We also define a *submodel* $\mathcal{M}'(D, U)$ for the partial evaluation with respect to a pair (D, U) as a projection of \mathcal{M} on $\text{REL}_O(D, U) \cup \text{REL}_H(D, U)$. It is straightforward to extend the above definitions over a set of decisions or utility functions. If $\text{REL}_H(D, U)$ includes another decision variable D' , then the partial evaluation of $\mathcal{M}'(D, U)$ depends on the local policy function $\Delta_{D'}$, which requires expanding the submodel to $\mathcal{M}'(\{D, D'\}, U)$ and jointly optimizes multiple policy functions, or searches for the equilibrium over the multiple decisions. We say a submodel is *stable* if its $\text{REL}_H(D, U)$ is free of other decision variables, and a locally optimal policy function computed by a stable submodel is globally optimal.

Identifying a stable submodel by the perturbation analysis is sound and complete, but it does not lead to efficient algorithms. Therefore, we use basic graph separation criteria (Pearl 2009) such as the d-separation, the back-door set for a pair of variables (X, Y) denoted by $\text{BD}(X, Y)$, and the front-door set denoted by $\text{FD}(X, Y)$ to identify the relevant sets. The $\text{REL}_U(D)$ is the set of utility nodes that are descendants of X_D and not d-separated from D given \mathbf{O}_D (Nielsen 2002), the $\text{REL}_O(D, U)$ is a subset of \mathbf{O}_D that remains as a $\text{BD}(D, U)$ after removing non-requisite observations, and the $\text{REL}_H(D, U)$ is a subset of hidden variables that is an element of any $\text{FD}(\{D\} \cup \text{REL}_O(D), U)$.

Submodel Decomposition

Unlike IDs with the perfect recall, LIMIDs may not have a total order on decision variables. A decision node is extremal (Lauritzen and Nilsson 2001) if its relevant information set blocks all the influences from other decisions to its relevant utility functions. A LIMID is soluble when there exists a total extremal order on the decision variables. For solving insoluble LIMIDs, we offer a partial solution ordering scheme. We first define two operations on two submodels $\mathcal{M}_1(\mathbf{D}_1, \mathbf{U}_1)$ and $\mathcal{M}_2(\mathbf{D}_2, \mathbf{U}_2)$ of a $\mathcal{M}(\mathbf{X}_D, \mathbf{U})$.

Definition 1 (Combining and marginalizing submodels).

1. *Combination:* Given two submodels $\mathcal{M}_1(\mathbf{D}_1, \mathbf{U}_1)$ and $\mathcal{M}_2(\mathbf{D}_2, \mathbf{U}_2)$ we define $\mathcal{M}'(\mathbf{D}', \mathbf{U}') := \mathcal{M}_1(\mathbf{D}_1, \mathbf{U}_1) \otimes \mathcal{M}_2(\mathbf{D}_2, \mathbf{U}_2)$ as a submodel generated by merging two submodels, i.e., $\mathbf{D}' = \mathbf{D}_1 \cup \mathbf{D}_2$, and $\mathbf{U}' = \text{REL}_U(\mathbf{D}')$. While merging two submodels, we also add informational arcs from all the nodes from $\text{REL}_I(\mathbf{D}', \mathbf{U}') = \text{REL}_I(\mathbf{D}_1, \mathbf{U}_1) \cup \text{REL}_I(\mathbf{D}_2, \mathbf{U}_2)$ to $\forall D_i \in \mathbf{D}'$.
2. *Marginalization:* $\Downarrow_{\mathcal{M}_2(\mathbf{D}_2, \mathbf{U}_2)} \mathcal{M}_1(\mathbf{D}_1, \mathbf{U}_1)$ is a submodel $\mathcal{M}'(\mathbf{D}', \mathbf{U}')$ with $\mathbf{D}' = \mathbf{D}_1 \setminus \mathbf{D}_2$ and $\mathbf{U}' = \text{REL}_U(\mathbf{D}_1 \setminus \mathbf{D}_2) \cup \{V\}$, where V is a new

value node inserted to the model for the conditional maximum expected utility of \mathcal{M}_2 defined by $\max_{\Delta}(\mathbf{D}_2 | pa(\mathbf{D}_2)) \mathbb{E}[\sum_{U_i \in \text{REL}_U(\mathbf{D}_2)} U_i | pa(\mathbf{D}_2)]$.

Proposition 1 (Submodel elimination by a partial solution ordering \mathcal{O}_D). Let $\langle \mathcal{M}_k \rangle_{k=1}^T$ be a sequence of submodels of \mathcal{M} and $\langle \mathcal{G}_k \rangle_{k=1}^T$ be a sequence of the DAGs of \mathcal{M}_k with $\mathcal{M}_T = \mathcal{M}$ and $\mathcal{G}_T = \mathcal{G}$. The partial solution ordering $\mathcal{O}_D := \langle \mathbf{D}_k \rangle_{k=1}^T$ is a sequence of disjoint subsets of \mathbf{X}_D such that: (1) $\cup_{k=1}^T \mathbf{D}_k = \mathbf{X}_D$, (2) $\mathcal{M}_{t-1}(\cup_{k=1}^{t-1} \mathbf{D}_k) \Downarrow_{\mathcal{M}'(\mathbf{D}_t)} \mathcal{M}_t(\cup_{k=1}^t \mathbf{D}_k)$, (3) $\text{REL}_H(\mathbf{D}_t, \text{REL}_U(\mathbf{D}_t)) \cap (\mathbf{X}_D \setminus \mathbf{D}_t) = \emptyset$.

Definition 2 (Submodel tree decomposition \mathcal{T} of \mathcal{M}). Given \mathcal{M} and its partial solution ordering \mathcal{O}_D generated along with $\langle \mathcal{M}_k \rangle_{k=1}^T$ and $\langle \mathcal{G}_k \rangle_{k=1}^T$, the submodel tree decomposition is a triple $\mathcal{T} := \langle T, \chi, \psi \rangle$, where $T = (C, S)$ is an oriented tree, and two labelling functions χ and ψ map a node $C \in C$ to a subset of decision variables in the partition $\chi(C) = \mathbf{D}_k \in \mathcal{O}_D$ and its submodel $\psi(C) = \mathcal{M}'(\mathbf{D}_k)$ of \mathcal{M}_k . A directed edge $(C_i, C_j) \in S$ connects two nodes if $\text{REL}_O(\chi(C_i)) \subseteq \mathbf{X}^{\psi(C_j)}$ where $\mathbf{X}^{\psi(C_j)}$ is the set of all variables of the submodel $\psi(C_j)$.

The submodel tree decomposition facilitates computing the MEU and the optimal strategy of LIMIDs. The message passing algorithm propagates a conditional expected utility between clusters, where each cluster partially evaluates a stable submodel by using probabilistic graphical model inference algorithms. The space and time complexity is exponential in the maximum treewidth of the individual submodels. The submodel tree decomposition provides a graphical model approach for decomposing LIMIDs, and lower complexity submodel graph decompositions can follow by relaxing the exact tree decomposition in various ways.

References

- Dechter, R. 2000. A new perspective on algorithms for optimizing policies under uncertainty. In *Proceedings of the 5th Conference on Artificial Intelligence and Planning Systems*, 72–81.
- Howard, R. A., and Matheson, J. E. 2005. Influence diagrams. *Decision Analysis* 2(3):127–143.
- Jensen, F.; Jensen, F. V.; and Dittmer, S. L. 1994. From influence diagrams to junction trees. In *Proceedings of the 10th international conference on Uncertainty in Artificial Intelligence*, 367–373.
- Lauritzen, S. L., and Nilsson, D. 2001. Representing and solving decision problems with limited information. *Management Science* 47(9):1235–1251.
- Mauá, D. D.; de Campos, C. P.; and Zaffalon, M. 2012. Solving limited memory influence diagrams. *Journal of Artificial Intelligence Research* 44:97–140.
- Nielsen, T. D. 2002. Decomposition of influence diagrams. *Journal of Applied Non-Classical Logics* 12(2):135–150.
- Pearl, J. 2009. *Causality*. Cambridge university press.
- Pralet, C.; Schiex, T.; and Verfaillie, G. 2006. From influence diagrams to multi-operator cluster dags. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, 393–400.