État de l'art - LIMIDs

Alexander Kunitsyn
alexander.kunitsyn@etu.sorbonne-universite.fr
Jad Chedid Al Rahbany
jad.chedid_rahbany@etu.sorbonne-universite.fr

Mars 2025

1 Introduction

Understanding the use of limited memory influence diagrams (LIMIDs) for modelling decision-making under uncertainty has several prerequisites, which will be explored in the following sections.

1.1 Bayesian networks

A Bayesian network is a graphical model that represents the conditional dependencies between multiple random variables as a directed acyclic graph (DAG).

We will define several important concepts, starting with conditional independence: suppose that event A is a hypothesis to be evaluated and B and C are observations. A and B are conditionally independent given C when $P(A|B,C) = P(A|C)$. In other words, as the probability of A occurring is the same regardless of whether C happened alone or in conjunction with B, B is irrelevant to the certainty of A. One defines the set of parent notes $pa(x)$ of a node x as the set of nodes that have an arc directed towards the node x - the set of variables that influence the distribution of x.

A set of random variables represented by nodes in a directed acyclic graph is a Bayesian network if for each of its nodes the local Markov property (LMP) is satisfied. This means that every variable is conditionally independent of its non-descendants given its parent variables. Rephasing in terms of the definition provided in the previous paragraph, in a Bayesian network, knowing the values of the non-descendants of a variable x is irrelevant to the value of x if the values of $pa(x)$ are known. It follows that in a Bayesian network, the value of a variable depends only on the value of its parents.

This is in line with the definition of a Markov process, where the future state of a system (x) only depends on its current state ($pa(x)$). Indeed, if x is conditionally dependent on a variable outside of $pa(x)$, this should be indicated with an arc, which, in turn, makes this new predecessor a parent of x.

Another important definition is that of the Markov blanket. The Markov blanket of a variable is a set of nodes such that all the information about this variable can be inferred from this set. If this set is minimal, it is called the Markov boundary. In a Bayesian network, the Markov blanket (boundary) of a variable is the set that contains the variable's parents, children, and the parents of its children (co-parents).

Knowing the values of the variables in the Markov blanket of x - their observations - is enough information to render x independent of the rest of the network. This dramatically simplifies the calculations needed to predict x.

Let A, B, C be sets of nodes of Bayesian network G. B d-separates A and C if A and C are conditionally independent given B. In terms of graphs, this means that every path that connects nodes from A to nodes from C has a node from B or its descendant on the way. More specifically, we can consider the following cases:

1. Chain structures ($a \to b \to c$): if b is in B, the path is blocked.
2. Fork structures ($a \leftarrow b \to c$): if b is in B, the path is blocked.
3. Collider structures/v-structures ($a \to b \leftarrow c$): if and only if b is *not* in B and none of its descendants are in B.

If all paths between A and C are blocked, A and C are conditionally independent given B. In this case, we also talk of the global Markov property being satisfied, meaning that for disjoint sets of nodes A and C, we have $P(A, C|B) = P(A|B)P(C|B)$.

1.2 Influence diagrams

Influence diagrams (IDs) are an extension of Bayesian networks that add decision nodes and utility nodes in order to mathematically represent a decision situation. IDs should satisfy the assumptions that decisions are temporally-ordered and are made in the context of all previous decisions. Influence diagrams have several types of nodes and arcs:

- Nodes:
    1. Decision nodes: shown as a rectangle, corresponds to a decision to be made
    2. Chance/uncertainty nodes: shown as an oval, corresponds to a random variable
    3. Utility nodes: shown as a diamond, corresponds to a utility function whose value one tries to maximise
- Arcs:
    1. Informational arcs: end in a decision node
    2. Conditional arcs: end in an uncertainty node

3. Functional arcs: end in a utility node

Influence diagrams are a compact way to represent decision problems under uncertainty, which makes them a useful tool for visualising and mathematically formalising a number of real-world scenarios.

However, finding exact solutions to a general ID is known to be NP-hard, which means that brute-force calculations are infeasible as the number of variables grows. A number of methods exist for simplifying the resolution, such as converting IDs into equivalent decision trees to then be solved with dynamic programming, removing some decision and chance nodes among others. If the ID happens to have the structure of a polytree (a DAG where no two nodes are connected by more than one path), belief propagation can be used, which is of polynomial complexity.

One approach to mitigating the challenge posed by loopy structures in a graph, which enables non-circular belief propagation, is to transform the graph into a junction tree. This restructuring facilitates more straightforward message passing. However, the computational complexity of generating these messages remains exponential.

It should be clear, therefore, that to make calculations easier and allow for methods and algorithms that are efficient irrespective of the structure of the graph, some of the axioms mentioned earlier might be relaxed.

1.3 LIMIDs - Limited memory influence diagrams

Limited memory influence diagrams (LIMIDs) were introduced by Nilsson and Lauritzen in 2001. The axioms of no-forgetting and temporal integrity of the order of decisions is simplified, allowing for easier calculation and creating a structure appropriate for modelling real-life scenarios that often include multiple independent forgetful decision-makers with a higher degree of accuracy. LIMIDs keep the definitions of arcs and nodes as specified in relation to traditional influence diagrams.

While LIMIDs relax a few constraints on traditional influence diagrams, they introduce their own computational challenges. For instance, as the requirement for decisions to be in correct chronological order has been relaxed, figuring out the optimal order now becomes a problem to be solved. Simplifying an influence diagram to create a LIMID is also not easy in the general case - a possible method consists of finding ”irrelevant” information through d-separation and rendering these new submodels independent of each other.

Exact inference in LIMIDs involves finding the optimal strategy by exhaustively evaluating MEU over all possible decision paths. While this guarantees an exact optimal solution, it becomes computationally infeasible for larger models. Approximate inference trades exactness for efficiency by finding tight bounds for the MEU instead of computing it exactly. This approach is useful when no clear decomposition presents itself - i.e. in the general case of a LIMID.

It is this latter approach which we will try to implement in code. The baseline model that we will be comparing against is brute-force inference. The challenge is creating a heuristic that works on models where the decomposition is not immediately evident. For example, in the scenario of pig farming from the cited Nilsson and Lauritzen paper, separating individual months makes sense logically. However, in many real-life applications, like IDs/LIMIDs representing the decision-making process in a large company, often no immediately-obvious decomposition exists.

2 Article

The paper explains an algorithm that aims to lighten the load on computation, by dividing the MEU computation task on smaller submodels, structured as a submodel decomposition tree to facilitate communication through message sending/receiving.

To do this, the paper defines a rigid mathematical model, allowing us to manipulate graphical elements inside formulas through a valuation algebra.
MEU $:= \max_{\Delta} \mathbb{E}[\sum_{U_i \in U} U_i]$ with Δ the set of decisions nodes and U the set of utility functions.

The final goal is to decompose an ID / LIMID into a submodel tree to evaluate the LMEU (local MEU) over all stable submodels (D', U') with
$D' \subseteq D, U' \subseteq U$, $\text{LMEU}_{\mathcal{M}(D',U')} := \max_{\Delta_{D'}} \mathbb{E}[\sum_{U_i \in U} U_i | pa(D')]$