Influence Diagrams, their Evaluation and their Applications

Boyd Witte
Roberto Lucchese
Lieuwe Vinkhuijzen

June 20, 2015

Abstract

1 Introduction

Modeling and optimizing a decision process is an active field of study within computer science. It proves a challenge to model the decision process of a business in a representation that is sufficiently specific to allow automated analysis which at the same time can be formulated, understood and approved of by the employees. Influence diagrams [2] are a tool to this end. They augment Bayesian Networks by introducing decision nodes and value nodes. Representing a decision process by such augmented Bayesian Networks allows the elicitation of processes of series of decisions which influence the environment in arbitrarily complex ways, while preserving the power of Bayesian networks to compute a sound prognosis.

If the user provides a utility function, the set of choices which maximizes his utility can be computed by novel algorithms due to [5] and [3]. The representation of a decision process in an influence diagram explicitly distinguishes the information known to a decision maker at the time of the decision from that which isn't. This not only allows us to properly model diagnostic and prognostic information, but as long as we do not require that all previous decisions are diagnostic to the next, influence diagrams may be equally well employed to model any matter of game, as Koller and Milch explore in [6], where they augment influence diagrams to Multi-Agent Influence Diagrams.

This paper is structured as follows. In Section 2 we give the definition and some theory of influence diagrams as described by Howard and Matheson in [2] and briefly describe unconstrained influence diagrams as defined by Jensen and Vomlelova in [3] and by Luque, Nielsen and Jensen in [4]. In Section 3 we discuss different methods of evaluating influence diagrams putting our attention on Shachter's method described in [5]. In Section 4 we will discuss some applications and finally, in Section 5, we will look at their advantages and shortcomings.

2 Influence Diagrams

An influence diagram (ID) models a process in which a user's decisions and the environment influence one another. When the environment lends itself to representation by a Bayesian network, the decision process may lend itself to representation by an influence diagram.

An influence diagram is a directed acyclic graph (DAG) of three types of nodes: *aleatory* nodes, which represent stochastic variables just as in a Bayesian network, *decision* nodes, which represent decisions a user must make, and *value* nodes, which provide a utility function. Figure 1 shows an example of an influence diagram. It will be useful to think of arcs into aleatory nodes as conditional arcs, and arcs into decision nodes as informational arcs. Conditional arcs, from an aleatory or a decision node, imply probabilistic dependence in the usual way: the child is conditioned upon the parents. Informational arcs represent the information known to the user when he makes the decision. Stochastic variables whose state is known to the user are distinguished from others by the labels *observables* and *unobservables*.

While influence diagrams with aleatory and decision nodes are useful for investigating a decision process, an influence diagram with a value node can be used to optimize the decision process. A value node provides a utility function which scores each configuration of the parents of the node. The goal of the influence diagram, then, is to find the series of choices which maximizes expected utility. In [2], Howard and Nielsen propose that an influence diagram may have any number of value nodes, so that the net utility is the sum of all utility functions.

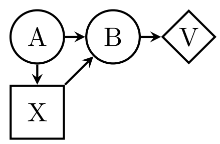


Figure 1: A simple influence diagram with decisions and stochastic nodes.

In general, an influence diagram may model a decision process in which any amount of decision makers cooperate to achieve a common goal. An influence diagram, therefore, imposes a partial temporal order on its decisions; a decision by one user may be made independently in time from some other, as they are not necessarily able to coordinate this order. We assume, however, that all users are aware of the order in they are to take their own decisions. In the special case when there is only one user, therefore, we require that the influence diagram impose a total rather than a partial temporal order on the decisions. Moreover, the *no forgetting* principle poses that users have perfect memory. Consequently, all informational predecessors of a decision are implicitly informational predecessors of all subsequent decisions.

For the remainder of this work, we will use the word *decision* to refer to a single decision variable, which represents a set of alternatives available to the user, and the word *choice* to refer to the situation when the user has chosen one of these alternatives. We will therefore hope to identify which choices optimize the utility function, and analyze whether certain decisions are influential.

2.1 Valuation of information

Influence diagrams allow the analysis of the question, "*What if I had (not) known A at the time of decision X?*", with A a stochastic variable. In the influence diagram, one need but add (or remove) an informational arc from A to X and compute the expected utility of both scenarios. The resulting difference is the value of knowledge of A, and it establishes an upper limit on the value of any test to elicitate A. Clearly, this value is strictly non-negative, as more knowledge at the time of a decision cannot decrease expected utility.

Figure 2 illustrates that it may be useful to know the degree of carcinogenicity of a chemical before deciding whether to ban, restrict or permit its usage. In practise, such information is not available, because it would be elicitated by an imperfect chemical test. If a chemist can provide information on the rate of false positives and false negatives, then the influence diagram may be modified to Figure 3. If the expected utility of this new influence diagram is greater than the cost of the test and the chemist, the test should be carried out. In particular, the chemist provides the conditional probability table of the test, node T in the diagram, conditioned on the actual degree of carcinogenicity.

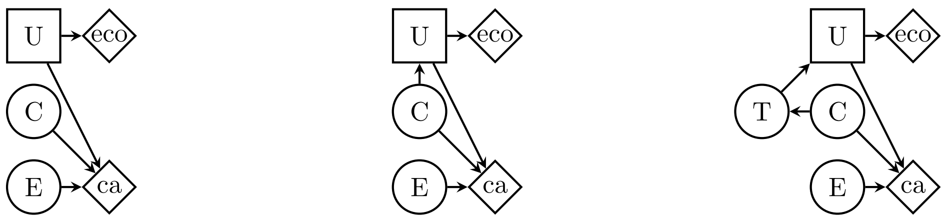


Figure 2: Comparison of the value of information on carcinogenicity. Node abbreviations are U for usage test, C for carcinogenicity, E for human exposure, eco for the economic value of the chemical, ca for the cost of the cancer it causes and T for the carcinogenicity test.

2.2 Decision trees

An influence diagram can be analysed by constructing its decision tree. A decision tree, as described by Howard and Nielsen in [2] and exemplified in Figure 3, represents a particular expansion of the probability distribution and the decisions of an influence diagram. Some influence diagrams can be translated into decision trees, but not all. In particular, when an unobservable variable influences both a predecessor and a successor of a decision node, the influence diagram can not be translated into a decision tree. This can be mediated by reversing the arcs in the influence diagram using Bayes' theorem. We call single-person influence diagrams with the no-forgetting principle *decision networks*. A decision network which can be translated to a decision tree is called a *decision tree network*.

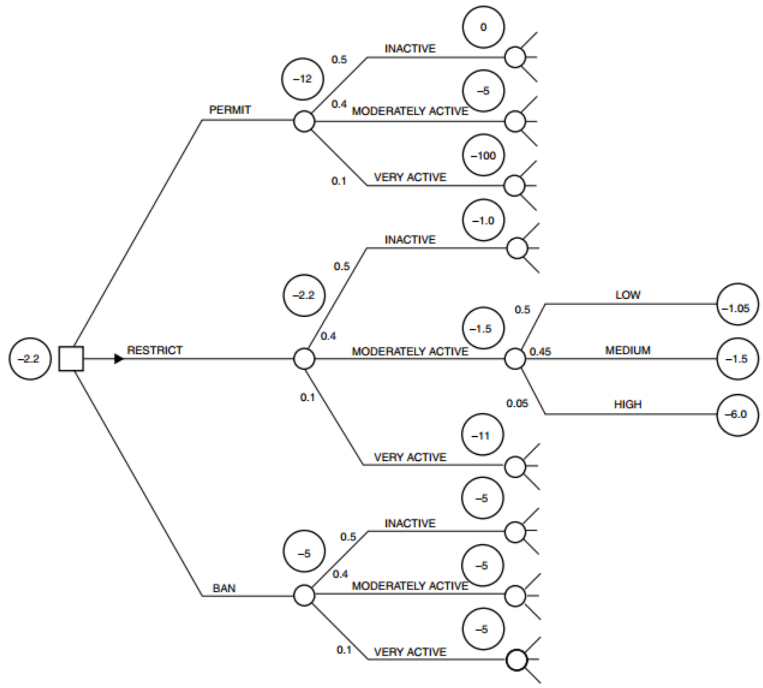


Figure 3: An example of a decision tree, constructed from the first influence diagram in Figure 2. Values are shown in millions of dollars. Source: [2]

Figure 3 shows the decision tree of the first influence diagram of Figure 2. To reiterate, it shows the expansion of the probability tree, in this case, the usage decision must come first, because it has no observable variables, and beyond that, the arbitrary expansion order of carcinogenic activity and then human exposure is used. The circles beside the branching points represent the expected value at that point. The expected value of the leaf, therefore, represent the value of the utility function if the situation which it represents comes to pass. The leaf with the highest utility function must be on the path of the best choice, so it is clear that there is a brute force approach to expectation maximization. In section 3, however, we present a faster algorithm to this end.

2.3 Unconstrained influence diagrams

It is not always possible or desirable to impose a total temporal order on the decisions in a decision process. The benefit of unconstrained influence-diagrams (UIDs) over IDs is that they are able to model this meta-decision: which decision will I take first? Instead of using an ID to determine the optimal choices for a certain ordered set of decisions as we have seen in in Section 2, the user now also needs to know the optimal order to take each decision in. When an order of decisions and the choices have been determined, we say that the UID has been solved.

In order to solve a UID, every possible order of decisions needs to be evaluated. This is done by converting the UID to a strategy directed acyclic graph (S-DAG). The S-DAG shows all possible paths from the starting condition to the final decision. For each path the user can determine the value for certain decisions.

Consider the following example, which we borrow from [4]. A doctor has to decide whether or not to administer treatment to a patient who may suffer from diabetes. He has an expensive but accurate blood test and a cheap but inaccurate urine test at his disposal, along with symptoms that are readily observable, to assess the likelihood that the patient has diabetes.

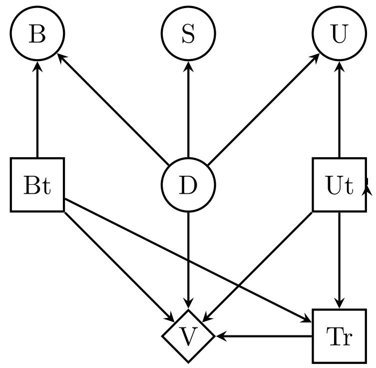


Figure 4: UID representing a doctor's process of deciding whether to administer treatment to a patient potentially suffering from diabetes. The abbreviations used are D for diabetes, S for symptoms, B and U for the results of the blood and urine tests, respectively, Bt and Ut for blood and urine tests, Tr for treatment and V for the resulting value.

Figure 4 illustrates how a UID models the diabetes example. The fact that having diabetes influences the outcome of the blood and urine tests is indicated in the usual way. The arrows from D, Bt, Ut and Tr to V indicate that the value node is influenced by these decisions, just as in a regular ID. Bt and Ut influence the results of their respective tests in the usual way, too, but there is no implicit temporal order in which to execute them. A wealthy, concerned patient might opt to do an expensive blood test, while an uninsured and uncaring patient might opt for the cheap urine test first.

The arcs from Bt and Ut to Tr indicate the order in which the decisions are to be taken. In particular, a decision may only be taken when all its antecedents have. Analogously, an observable variable only becomes observable when all of its antedecent decisions have been made. Therefore a partial order is imposed on the order in which decisions are taken and in which order evidence becomes available.

When solving the diabetes problem, we first transform the UID into an S-DAG as seen in Figure 5. The user identifies the two paths, and has to evaluate for each path the choices that result in the most favorable outcome. Luque, Nielsen and Jensen propose an algorithm for this task in [4].

It is interesting to see how Bt influences B. The decision Bt is either *yes* or *no*. If a blood test is conducted, then the result of the blood test, B, depends on whether the patient has diabetes, D. It will then yield either a positive or a negative result. Otherwise, if no blood test is carried out, B takes the value NaN, and nothing is learned about D. This is an example of value-dependent independence. Whether D is independent of B depends on the value of Bt: $D \perp\!\!\!\perp B \quad | \quad Bt = \text{no}$, whereas $D \not\perp\!\!\!\perp B \quad | \quad Bt = \text{yes}$.

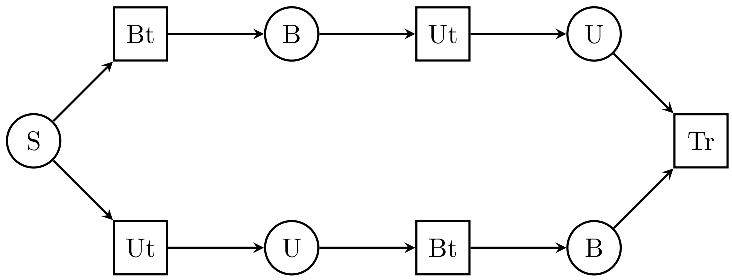


Figure 5: An S-DAG based upon the UID in Figure 4.
Source [4]

3 Evaluating Influence Diagrams

Bayesian inference is already a computationally difficult problem, and adding decisions to the mix makes life no easier. Worse still, when a utility function evaluates the outcomes of certain stochastic variables, maximizing expected utility requires Bayesian inference for every conceivable strategy. To reiterate, the goal of our problem is of one of two kinds. In case I, we are interested in the probability of a particular variable taking a particular value given a set of choices. Case II pertains to maximizing expected utility. Evaluating an unrestricted influence diagram has the additional difficulty that the order of the decisions in part of the decision problem.

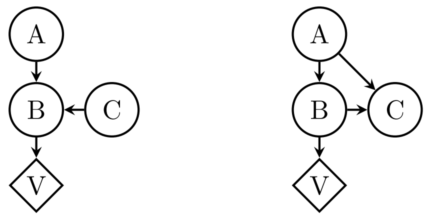
Nevertheless, both versions can be solved by algorithms in better ways than a brute force approach. Normal IDs are solved for case I by a traditional Bayesian approach, for example using Pearl's algorithm. Case II is solved by an algorithm by Shachter [5]. Unconstrained IDs are solved for case II by an algorithm due to Luque et al. [4]. It is an anytime algorithm; it can be stopped at any time to return an admissible strategy. In this section, we will discuss both.

3.1 Evaluating constrained, single-person influence diagrams

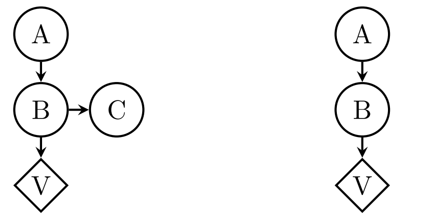
Shachter [5] provides the first algorithm to this end. It uses marginalisation, Bayes' rule, and expectation maximization as tools behind graph transformations. These graph transformations chip away at the influence diagram while preserving the expected utility and remembering the best decisions, until the graph has been dissolved, at which point the answer has been computed. Shachter's algorithm uses three graph transformations to compute the optimal strategy: arc reversal, removal of barren nodes, removal of aleatory nodes and expectation maximization, illustrated in Figure 6.

When a node has incoming but no outgoing arcs, it clearly has no effect on the utility. These *barren nodes* may be safely removed from the diagram. A barren node may be created by reversing all outgoing arcs from an aleatory node. Together, these two graph transformations have the effect of simplifying the diagram, at the computational cost of recomputing several probability distributions.

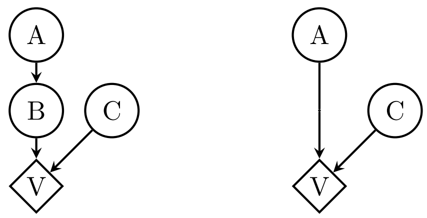
When node B is directly incident upon node V, as illustrated in Figure 6c, node B may be removed from the graph, provided the CPT of V is updated to be conditioned on the parent of B. Node V may be either another aleatory node or the value node. A similar graph transformation removes a decision node incident on the value node. Instead of marginalisation, however, we update the CPT to reflect the scenario in which the best choice was made. This process is called expectation maximization.



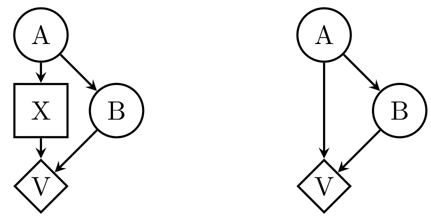
(a) Arc reversal applied to the arc between nodes B and C.



(b) Node C is removed, because it does not influence the utility V.



(c) Node B is removed.



(d) Decision node X is removed through expectation maximization.

Figure 6: Illustrations of the four graph transformations used in Shachter's algorithm.

In [5], Shachter proves that there is always a way to apply these four operations in a series to dissolve the influence diagram until only the value node remains, and that each transformation preserves the maximal expected utility.

4 Applications

In this Section we present three representative cases of real world applications of influence diagrams. They are reproduced from a paper by M. Gomez [10] in which more that a hundred scenarios are presented.

4.1 Software Change Management

In [8] C. J. Burgess, I. Dattani, G. Hughes, J. H.R. May and K. Rees show how IDs can be applied to a Software Change Management problem. In particular, when a new version of a software has to be released a lot of different Requirements (Work Packages) can be included in this new version. For each possible Requirement the package to be included must meet certain characteristics. The complete ID used to determine whether or not to include any particular Requirement in the next software release is presented in Figure 7.

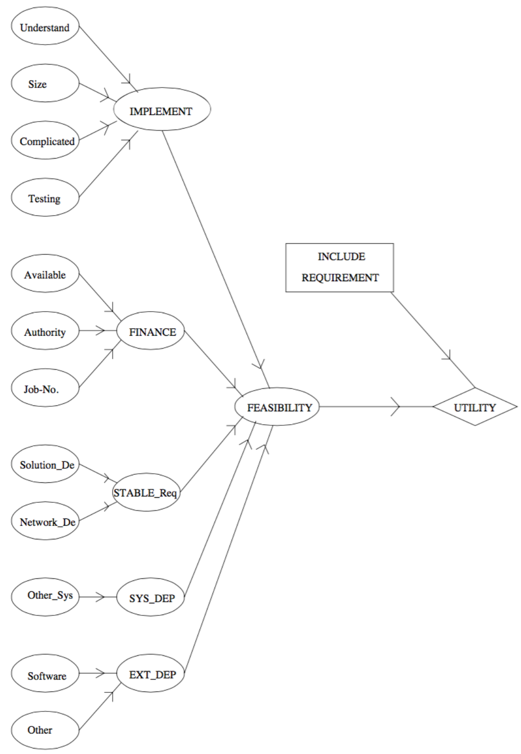


Figure 7: Requirements Capture Influence Diagram. Source: [8]

4.2 Baltic Cod management

In [9] S. Kuikka, M. Hilden, H. Gislason, S. Hansson, H. Sparholt and O. Varis present an approach, consisting of three main steps, to identify management measures that help to reduce the risk of Baltic cod overfishing. In particular, from a modeling point of view, their task is to model the effects of the mesh size and fishing mortality on the interest variables (catch, biomass, etc.). They also want to demonstrate how sensitive the advised actions are to the relevant scientific uncertainties.

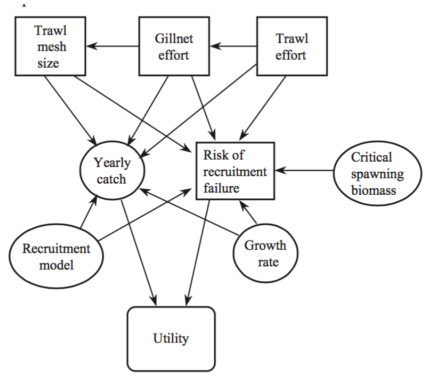


Figure 8: Influence diagram used to model Baltic cod management. Source: [9]

Using their approach, of which the ID is presented in Figure 8, based on realistic environmental conditions assumptions and current fishing mortality rates they could verify that the actual Baltic cod fishery is unsustainable.

4.3 Two-person prisoner's dilemma

In [6], Koller and Milch show that complex non-cooperative non-zero-sum games can be easily modeled by influence diagrams. For example, a two-person prisoner's dilemma is modelled by the influence diagram in Figure 9.

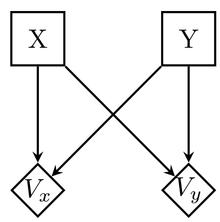


Figure 9: An influence diagram which models the prisoner's dilemma.

The two value nodes represent the asymmetric values attached to different outcomes by the two agents. Just as in normal influence diagrams, each decision is associated with a decision maker, in these decision diagrams, analogously, each value node is associated with one decision maker.

A strategy for a particular user consists of an assignment of conditional probabilities to each of his decisions, conditioned of course on the observables at each decision. Depending on the value functions, there may not be a set of strategies which is optimal for everybody. However, it is conventional in game theory to say that players act rationally if they are in a Nash equilibrium. Specifically, a set of strategies is in a Nash equilibrium when no player can increase his expected utility by changing his strategy, given that no other players deviate from their strategy.

5 Discussion

In this work, we have introduced influence diagrams as Bayesian networks, augmented to model decision processes. We have seen that they are a compact yet powerful tool which allows both for elicitation of complex processes by non-computer experts and for their sound analysis using the known paradigm of Bayesian inference. Although influence diagrams have been around a long while, there are still directions for future research.

In [2] and [1], Howard and Nielsen present IDs as a way for domain experts to model decision processes without the need for an analyst to write a custom software program to compute a decision. While this is certainly an improvement over custom software, we still feel that the involvement of domain experts is best limited to a minimum, using them to identify the variables and decisions but relying on automated systems to sample large databases to identify the appropriate dependencies and construct the corresponding conditional probability tables.

Influence diagrams are a good way to solve problems with tradeoffs between utility and risk. Each set of choices yields a number of possible outcomes. These choices present a very complex Pareto front, and Howard and Nielsen propose simply solving it through expectation maximalization. This can be an appropriate strategy, particularly because the utility can be made to be risk-averse. However, when an influence diagram has multiple value nodes, for example, one for profit and one for environmental impact, it is not obvious that a net value of benefits minus costs does justice to the complexity of the situation. Multicriteria optimization is a field of study in its own right, therefore we believe that the high-dimensional Pareto fronts constructed by influence diagrams deserve further research.

We belive that in general, most research into Bayesian networks can be modified to apply to influence diagrams, too. A lot of future work, therefore, is to augment the relevant parts of this research.

References

[1] Miller III, Allen C., et al. Development of automated aids for decision analysis. STANFORD RESEARCH INST MENLO PARK CA, 1976.

[2] Howard, Ronald A., and James E. Matheson. “Influence diagrams.”Decision Analysis 2.3 (2005): 127-143.

[3] Jensen, Finn V., and Marta Vomlelov. “Unconstrained influence diagrams.”Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 2002.

[4] Luque, Manuel, Thomas Dyhre Nielsen, and Finn V. Jensen. “An anytime algorithm for evaluating unconstrained influence diagrams.”The Fourth European Workshop on Probabilistic Graphical Models. 2008.

[5] Shachter, Ross D. “Evaluating influence diagrams.”Operations research 34.6 (1986): 871-882.

[6] Koller, Daphne, and Brian Milch. “Multi-agent influence diagrams for representing and solving games.”Games and Economic Behavior 45.1 (2003): 181-221.

[7] Manuel Gomez. “Real-World Applications of Influence Diagrams.”(2004)

[8] Colin J. Burgess, Ilesh Dattani, Gordon Hughes, John H.R. May, Kearton Rees. "Using Influence Diagrams in Software Change Management"

[9] S. Kuikka, M. Hilden, H. Gislason, S. Hansson, H. Sparholt and O. Varis. "Modeling environmentally driven uncertainties in Baltic cod (Gadus morhua) management by Bayesian influence diagrams"(1998)

[10] Erik Jrgensen. "Textbook notes of herd management: Influence Diagrams: Potential use and current limitations"(1996)