SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 6 No. 2
ISSN 2054-7390

Intelligent Decision Support Machines for Business Decisions

Hans W. Gottinger, STRATEC Munich
stratec_c@yahoo.com, gottingerhans@gmail.com

"How can we move from the classical view of rational agent who maximizes expected utility over an enumerable state-action space to a theory of the decisions faced by resource bounded AI systems deployed in the real word…" --- Gershman, Horvitz and Tenenbaum (2015).

ABSTRACT

An intelligent decision support machine (IDSM) is a computer-based interactive tool of decision making for well-structured decision and planning situations that uses jointly decision-theoretic methods and machine learning techniques with access to structured data bases. An IDSM emerges from a model based system underlying a decision model (under uncertainty) imposing a normative (prescriptive) structure of decision making.

In the past years there has been substantial attention devoted to the use of artificial intelligence (AI) techniques, first most commonly rule-based expert systems, more recently methods of machine learning as tools for decision support. Based on those rules we design a machine learning algorithm guiding through principles of an IDSM.

1 Introduction

An intelligent decision support machine (IDSM) is a computer-based interactive tool of decision making for well-structured decision and planning situations that uses jointly decision-theoretic methods and machine learning techniques with access to structured data bases. An IDSM emerges from a model based system underlying a decision model (under uncertainty) imposing a normative (prescriptive) structure of decision making.

In the past years there has been substantial attention devoted to the use of artificial intelligence (AI) techniques, most commonly rule-based expert systems, as tools for decision support. These systems typically use production rules to develop a diagnosis of a disease or a system malfunctioning, for example.

Given the diagnosis, the system generates a recommended solution to the problem. The solution may be a drug therapy in a medical domain or a set of parts to replace in a trouble shooting application.

Rule based techniques have proven to be very attractive for a variety of problems, particularly those which have fairly well structured (though possibly large) problem spaces, which can be solved through the use of heuristic methods or rules of thumb, and are currently solved by human experts. In these domains the reasoning and explanation capabilities offered by rule based expert systems are very effective. Given various types of goal or payoff functions the system generates a range of decision outcomes to choose

DOI: 10.14738/tmlai.62.4372
Publication Date: 29th March, 2018
URL: http://dx.doi.org/10.14738/tmlai.62.4372

from. Sequential decision rules need to be updated when applied to time evolving difficult problems relating to the following situations:

(i) there is substantial uncertainty on various levels of decision making;

(ii) the preferred solution is sensitive to the specific preferences and desires of one or several decision makers;

(iii) problems of rationality and behavioral coherence are intrinsic concerns of decision systems.

In related established fields such as operations research and management science we have been developing methods for allocating resources under various conditions of time, uncertainty and rationality constraints. Central to these methods is the existence of an objective or utility function, as an indicator of the desirability of various outcomes. We will draw on this body of knowledge, especially elements related to the normative use of individual and group decision making to approach difficult decision problems.

Decision making is best viewed as a process of making a series of related, incremental observations, judgements, and decisions. In some cases simple, deterministic relations such as those used in many rule-based expert systems can be used for informed decision making, while in others explicit treatment of uncertainty or vagueness in the domain and the objectives of the decision maker are needed. Therefore, for a single complex decision situation it may be desirable to combine deterministic reasoning with uncertainty and decision calculi in the course of exploring the decision situation, as our understanding and insight into a problem evolves. It is a move toward a “machine economicus” (Parkes and Wellman, 2015).

2 Structuring Decisions for Computation

An early fusion of Artificial Intelligence and Decision-Making started with expert systems consisting of a Knowledge Base (KB) and Inference Engine (IE) (Gottinger, and Weimann, 1990).

Expert Systems were basically static constructs.. The KB had to be constructed and created manually on present and past knowledge structure --- there was no endogeneous learning from data.. The inference engine consisted of logical inference rules, i.e. if-then rules, that were operating repeatedly on given knowledge bases. Yet the architecture of interaction between KB and IE can still be used in machine learning (ML) mode by opening up the KB to dynamic time related changes and future (uncertain) events, in the same way as we perceive Markov chains and processes. Let logic rules work probabilistically on changing events in a decision-theoretic framework. Thus ML systems are extensions of ES in two-fold ways: they work as mechanism designs on variety, diversity of decision relevant events, and they operate on probabilistic events in future courses of action (Pearl, 1993).

A procedure of ML on decision tree structures could be roughly as follows. First observe time-dependent changes in probability estimates with Bayesian updating on initial probability and likelihoods for given sets of events affecting outcomes (expected utility). Then preference changes on alternatives induce changing utilities. Further, risk preferences change on probabilistic outcomes (risk proneness, aversion).

The basic result of the axioms of decision theory is the existence of a value function for scoring alternative sets of outcomes under certainty and a utility function for scoring uncertain outcome bundles. If the decision maker accepts the axioms (say, Savage’s axioms, Savage 1954) in the sense that he would like his decision making to be consistent with these axioms, then the decision maker should choose that course of action which maximizes expected utility – with its early lead to stochastic learning models (Bush and Mosteller, 1955). The importance of these axioms is that encoding decision procedures based on these axioms provide a basis for recommendations by an intelligent decision aid under uncertainty. They provide an explicit set of norms by which the system will behave. Other authors have argued why an individual should accept the decision axioms for decision making. The acceptance of these axioms is implicit in the philosophy and design of decision methods described here.

In addition, an approach to decision making based on decision theory has a mechanism, at least in principle, for handling completely new decision situations. The construct ensures the existence of a value and utility function. If the current expression of the preferences in the system does not incorporate the attributes of a new decision situation, the system can resort to the construction of a higher level or more general preference structure. By following these principles we are able to use the richness of modern decision theory and their axiomatic foundation (Fishburn 1988).

Domains in which there is a well developed empirical and theoretical basis for development of utility functions (e.g. financial and engineering decision making and some areas in medicine) are most promising.

Thus the decision axioms, along with the fundamentals of first order logic, provide a normative basis for reasoning about decisions. It is in this light that both logical and probabilistic inference will be utilized in an intelligent decision system.

3 Decision model based Representation

For decision making, a model consists of the following elements

(1) alternative prospects

(2) state descriptions,

(3) relationships, and

(4) preferences.

There can be no decision without alternatives, the set of distinct resource allocations from which the decision maker can choose. Each alternative must be clearly defined. State descriptions are essentially collections of concepts with which the decision is framed. It includes the decision alternatives and the outcomes which are related to the choices. The state description forms the means of characterizing the choice and outcome involved in a decision. The state description is also intertwined with expression of relationships. Relationships are simply the mappings of belief in some elements of the state description to others. The relations could be represented as logic relations, if-then rules, mathematical equations, or conditional probability distributions. The final component of a decision model is preferences. These are the decision maker's rankings in terms of desirability for various possible outcomes. They include not only his rankings in terms of the various outcomes which may occur in a decision situation, but also his attitude toward risky outcomes and preferences for outcomes which may occur at various times. They also embody information identifying those factors in a decision situation that are of concern, whether a factor indicates a desirable or undesirable outcome, and how to make tradeoffs among alternative collections of outcomes.

4 Influence Diagrams

As a computationally convenient way for a decision model based representation we deal with influence diagrams. Influence diagrams are network depictions of decision situations (Howard & Matheson 1981). They lend themselves to a dynamic representation of decision situations. They also give rise to forms of neural network presentations facilitating machine learning with deep learning (Davis,2016). Until recently, their primary use has been in the professional practice of decision analysis as a means of eliciting and communicating the structure of decision problems. Each node in the diagram represents a variable or decision alternative; links between nodes connote some type of influence'. Decision makers and experts in a given domain can view a graphical display of the diagram, and readily apprehend the overall structure and nature of dependencies depicted in the graph. There has been additional attention devoted to influence diagrams based on their uses in providing a complete mathematical description of a decision problem and as representations for computation. In addition to representing the general structure of a decision model, information characterizing the nature and content of particular links is attached to the diagram (Holtzman, 1989). The diagram then presents a precise and complete specification of a decision maker's preferences, probability assessments, decision alternatives, and states of information. In addition the diagrammatic representations can be directly manipulated to generate decision-theoretic recommendations and to perform probabilistic inference. The formalism of Bayes networks (Pearl,1993) are identical graphical constructs which express probabilistic dependencies (no preferences or decisions). Following the notation of Shachter (1986) we define the syntax and semantics of influence diagrams.

Definition .An influence diagram is an acyclic directed graph G = (N,A) consisting of a set, N, of nodes and a set, A, of arcs. The set of nodes, N, is partitioned into subsets V, C, and D. There is one value node in V, representing the objective of the decision maker. Nodes in C, the chance nodes, represent uncertain outcomes. Nodes in D, the decision nodes, represent the choices or alternatives facing the decision maker.

A simple diagram appears in Fig.1. By convention, the value node is drawn as a diamond, chance nodes are drawn as circles, and decision nodes are drawn as rectangles.

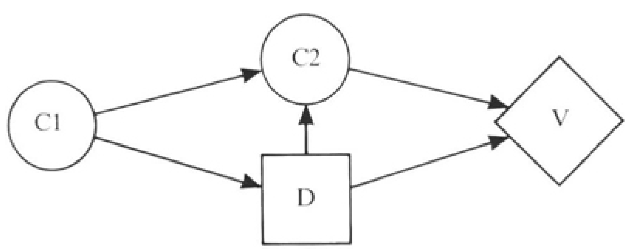


Figure.1 — A Simple Influence Diagram.

V is the value node, the proposition which embodies the objective to be maximized in solving the decision problem. C1 and C2 represent uncertainties, and D represents the decision. The semantics of arcs in the graph depend on the type of the destination node. Arcs into value or chance nodes denote probabilistic dependence. These arcs will be referred to as probabilistic links. Arcs terminating in decisions indicate the state of information at the time a decision is made. Thus, C1 is an uncertainty which is probabilistically influenced (conditioned) by C2 and the decision. The ultimate outcome V, depends on the decision D and C2.

Definition — Each node's label is a restricted proposition, a proposition of the form p (t_1 t_2 ... t_n) where each t_i, is either an object constant or alternative set.

We now define a set $\Omega(i)$ and a mapping π_i, for each node.

Definition — The set $\Omega(i)$ is the outcome set for the proposition represented by node i. It is a set of mutually exclusive and collectively exhaustive outcomes for the proposition.

Definition — The predecessors of a node i are the set of nodes j with arcs from j to i.

Definition — The successors of a node i are the set of nodes j such that there is an arc from i and j.

The mapping π_i, depends on node type. The domain of each mapping is the cross product of the outcome sets of the predecessors of node i. Let the cross product of predecessors of i be CP(i) where

$$CP(i) = \{ \Omega(i1) \times \Omega(i2) \ldots \times \Omega(in) | \text{nodes } i1, \ldots, in$$

ϵ predecessors of node i}

The range of each mapping π_i depends on the type of node i. Each is discussed in turn.

The value node

The value node expresses the decision maker's relative valuation of different possible combinations of outcomes for its predecessors. Since we require a cardinal measure for expected value calculations, the outcome set of value nodes has some restrictions. In terms of the previous definition of the outcomes set, the set $\Omega(i)$ for the value node is:

$$\Omega(i) = \{i\Theta | \Theta \in \{\{x_1/X_{11}\}\{x_1/X_{12}\}, \ldots, \{x_1/X_{1K}\}\}\}$$

where $X_{11} \ldots X_{1K} \in R$

Thus, there is only one restricted variable, x_1, and its values, the $X_{1K}{}^S$, are real valued. We can therefore associate with each member of $\Omega(i)$ exactly one real number. The value function π_i, for a value node is defined as follows:

$$\pi_i: CP(i) \rightarrow \text{alternative set of } X_1 \{X_{11}, X_{12}, \ldots, X_{1K}\}$$

This function maps each combination of outcomes for the predecessors into a single real number. This will be used to express the expected value or the expected utility as a function of the outcomes of the predecessor nodes.

Chance nodes

Chance nodes represent uncertain propositions that are not directly controlled by the decision maker. The members of $\Omega(i)$ are the possible outcomes for the proposition. There are two types of chance nodes. A stochastic chance node admits uncertainty regarding the outcome of the proposition given the values of its predecessors. In this case the mapping π_i is a conditional probability density function.

$$\pi_i: CP(i) \times \Omega(i) \rightarrow [0,1]$$

in probabilistic terms $\pi_i(\omega_i | \ \omega_{i1}, \ \omega_{i2}, \ \ldots \ \omega_{in})$. If node i has no predecessors, then $\pi_i(\omega_i)$ is a prior (unconditional) probability distribution.

The other type of chance node is a deterministic chance node. The outcome of a deterministic chance node is a deterministic function of the outcomes of its predecessors. In this case, the mapping π_i, is defined as follows.

$$\pi_i: CP(i) \rightarrow \Omega(i)$$

Thus, given the values of its predecessors, there is no uncertainty regarding the outcome of the proposition. If node z has no predecessors, the $\pi_i(\ )$ is constant.

Decision nodes

Decision nodes represent propositions which are under the direct control of the decision maker. The members of $\Omega(i)$ are the alternative outcomes from which the decision maker can choose. At the time the decision is made, the decision maker knows the outcomes of the predecessors of i. The mapping π_i therefore expresses the optimal decision choice as a function of what is known at the time the decision is made.

$$\pi_i: CP(i) \rightarrow \Omega(i)$$

The mapping is calculated in the course of manipulating an influence diagram and the associated maximization of expected value. Though the construct is similar to that of the mapping for a deterministic chance node, it differs in that it is the result of an optimization. For this reason we define a special construct.

Definition — A decision function $\pi_{d,i}$, is a function

$$\pi_{d,\ i}: CP(i) \rightarrow \Omega(i)$$

determined as the result of an optimization (see Transformations).

Transformations

An influence diagram is said to be a decision network if (1) it has at least one node, and (2) if there is a directed path which contains all the decision nodes (Howard & Matheson 1981). The second condition implies that there is a time ordering to the decisions, consistent with the use of an influence diagram to represent the decision problem for an individual. Furthermore, arcs may be added to the diagram so that the choices made for any decision are known at the time any subsequent decision is made. These are 'no-forgetting' arcs, in that they imply the decision maker (1) remembers all of his previous selections for decisions, and (2) has not forgotten anything that was known at the time of a previous decision.

The language of influence diagrams is a clear and computable representation for a wide range of complex and uncertain decision situations. The structure of dependencies (and lack thereof) is explicit in the linkages of the graph, as are the states of information available at each state in a sequence of decisions. The power of the representation lies, in large part, in the ability to manipulate the diagram to either (1) express an alternative expansion of a joint probability distribution underlying a particular model, or (2) to generate decision recommendations. The basic transformations of the diagram required to perform these operations are node removal and arc reversal. These operations will be illustrated and defined with respect to a generic set of node labels: i and j are chance nodes, v is the value node. The labels p1, p2, and p3 will in general represent groups of predecessors of i, j, or v as indicated by the figures. In the interest of simplifying the descriptions of the operations, they will be treated as individual nodes. More detailed descriptions of these operations appear in Shachter (1986) and Holtzman(1989).

Removal of a stochastic chance node, i, which is a predecessor of a value node, v, is performed by taking conditional expectation.

The new expected value function for v is calculated as follows:

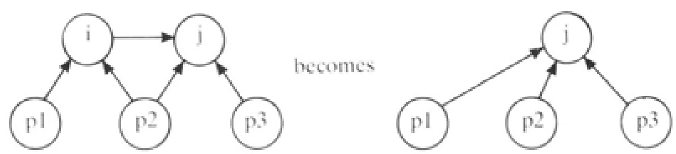
$$\pi_{\text{new},v}\ (\omega_{p1},\omega_{p2},\omega_{p3}) = \sum_{\omega_i \in \Omega(i)} \pi_{\text{old},v}(\omega_{p1},\omega_{p2},\omega_{p3})\ \pi_i(\omega_i|\omega_{p1},\omega_{p2})$$

The value nodes new predecessors are p 1, p2, and p3.

Removal of a deterministic chance node, i, which is a predecessor to the value node, v, is performed by substitution. The picture of this process is the same as the previous case. The new expected value function for v is:

$$\pi_{\text{new, v}}(\omega_{p1},\ \omega_{p2},\ \omega_{p3}) = \pi_{\text{old, v}}(\ \pi_i,\omega_{p2}),\ \omega_{p2},\ \omega_{p3})$$

Removal of a stochastic chance node, i, which is a predecessor to another chance node, j, is also performed by taking conditional expectation.

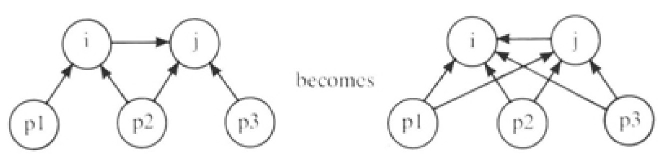


The new distribution for successor node j is calculated as:

$$\pi_{\text{new},j}(\omega_{p1},\omega_{p2},\omega_{p3}) = \sum_{\omega_i \in \Omega(i)} \pi_{\text{old},j}(\omega_j|\omega_i,\omega_{p2},\omega_{p3})\ \pi_i(\omega_i|\omega_{p1},\omega_{p2})$$

The new predecessors of j are the predecessors of j other than i, that is, p1, p2 and p3.

Removal of a decision node, i, predecessor to the value node v is performed by maximizing expected utility. The decision node can only be removed when all of its predecessors are also predecessors of the value node; that is, the choice is based the expectations for the value, given what is known.



After removal the new expected value function for v is:

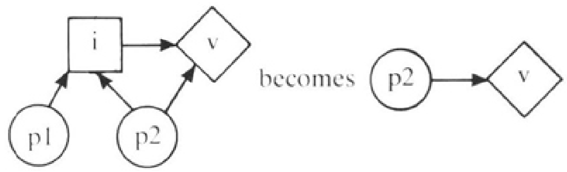
$$\pi_{\text{new},v}(\omega_{p2}) = \underset{\omega_i \in \Omega(i)}{\text{Max}}\ \pi_{\text{old},v}(\omega_i,\omega_{p2})$$

The new predecessors of v are the predecessors of i which are also predecessors of v, p2 as illustrated here. Note that there may be some informational predecessors of i, for example p1, which are not predecessors v before the removal. The values of these variables are irrelevant to the decision, since the expectation for the value is independent of their values. The optimal policy for the decision i is:

$$\pi_i = \underset{\omega_i \in \Omega(i)}{\text{argmax}}\ \pi_{\text{old},v}(\omega_i,\omega_{p2})$$

this is the calculated π_i for decision nodes. We will refer to this calculated mapping as the decision function for i, $\pi_{d,i}(\omega_{p2})$.

Reversal of a probabilistic link between chance nodes is an application of Bayes' rule. Reversing a link from node i to node j can be performed so long as there is no other path from i to j (this is necessary to prevent the reversal from creating a cycle).



In reversing, the new conditional probability description for i and j are calculated as:

$$\pi_{\text{new},j}(\omega_j|\omega_{p1},\omega_{p2},\omega_{p3}) = \sum_{\omega_i \in \Omega(i)} \pi_{\text{old},j}(\omega_j|\omega_i,\omega_{p2},\omega_{p3})\ \pi_{\text{old},i}(\omega_i|\omega_{p1},\omega_{p2})$$

$$\pi_{\text{new},j}(\omega_i|\omega_j,\omega_{p1},\omega_{p2},\omega_{p3}) = \frac{\pi_{\text{old},j}(\omega_j|\omega_i,\omega_{p2},\omega_{p3})\ \pi_{\text{old},i}(\omega_i|\omega_{p1},\omega_{p2})}{\pi_{\text{new},j}(\omega_j|\omega_{p1},\omega_{p2},\omega_{p3})}$$

The operations of reversal and removal allow a well formed influence diagram to be transformed into another 'equivalent' diagram. The original and the transformed diagrams are equivalent in two senses. First, the underlying joint probability distribution and state of information associated with each is identical, since the diagram expresses alternative ways of expanding a joint distribution into a set of conditional and prior distributions (Howard & Matheson 1981). Secondly, the expectation for the value in the diagram and the sequence of recommended actions from decision node removal are invariant over these transformations (Shachter 1986, Holtzman 1989). In the next section, we focus on applying a sequence of these manipulations to obtain these recommendations.

Solution procedures

On the basis of these manipulations, there exist algorithms to evaluate any well formed influence diagram (Shachter 1986). For purposes of probabilistic inference, we need two separate algorithms. In one version, which applies to well formed diagrams, evaluation consists of reducing the diagram to a single value node with no predecessors, the value of which is the expected value of the decision problem assuming the optimal policy is followed. In the course of removing decisions, the optimal policy, that is, the set of decision functions $\pi_{d,i}$ associated with each decision is generated. In the other algorithm, the objective is to determine the probability distribution for a variable, as opposed to its expected value. Both versions of the algorithm are described below.

Procedure EXPECTED VALUE (diagram)

1. Verify that the diagram has no cycles.
2. Add 'no-forgetting' arcs between decision nodes as necessary.
3. WHILE the value node has predecessors

3.1 IF there exists a deterministic chance node predecessor whose only successor is the value node, THEN
Remove the deterministic chance node into the value node ELSE

3.2 IF there exists a stochastic chance node predecessor whose only successor is the value node, THEN
Remove the stochastic chance node into the value node ELSE

3.3 IF there exists a decision node predecessor and all the predecessors of the value node are predecessors of the decision node, THEN
Remove the decision node into the value node ELSE

3.4 BEGIN

3.4.1 Find a stochastic predecessor X to the value node that has no decision successors.

3.4.2 For each successor S_x of X such that there is no directed path from X to S_x
Reverse Arc from X to S_x

3.4.3 Remove stochastic predecessor X

4. END

At the conclusion of the EXPECTED VALUE procedure, the value node has no predecessors, and its single value is the expected value of the value node. Optimal decision functions are generated in the course of removing the decision nodes.

The algorithm to solve for a probability description (or lottery) for a node is as follows.

Procedure PROBABILITY-DISTRIBUTION (diagram)

1. Verify that the diagram has no cycles.
2. IF the value node is deterministic, THEN convert to a probabilistic chance node with unit probability on deterministic values.
3. WHILE the value node has predecessors

3.1 IF there exists a deterministic chance node predecessor whose only successor is the value node, THEN
Remove the deterministic chance node into the value node ELSE

3.2 IF there exists a stochastic chance node predecessor whose only successor is the value node, THEN
Remove the stochastic chance node into the value node ELSE

3.3 IF there exists a decision node predecessor and all the predecessors of the value node are predecessors of the decision node, THEN
Remove the decision node from the list of predecessor ELSE

3.4 BEGIN

3.4.1 Find a stochastic predecessor X to the value node that has no decision successors.

3.4.2 For each successor S_x of X such that there is no directed path from X to S_x
Reverse Arc from X to S_x

3.4.3 Remove stochastic predecessor X

4. END

The termination of this procedure is a probabilistic chance node with probabilities over the alternative possible outcomes of the original value node. Note that if decision predecessors are encountered in the algorithm, the distribution will be conditioned on the possible choice of the decision variables. The procedure does not remove decision nodes or generate decision functions.

5 Decision Language

Recall the elements that are necessary to represent a decision domain — alternatives, state descriptions, relationships, and preferences. We will summarize by indicating how each element of a decision description can be expressed with respect to the constructs generated above.

First, recall that propositions form the basic unit of representation for a decision domain. There are three levels of knowledge regarding a proposition expressible in the language. First, it is possible to express a fact for a proposition, that is, a set of values for the variables (as in a fact substitution) in the proposition that are asserted to be true with certainty. Second, the values of the variables in a proposition may be restricted to some set. Thus, the outcomes for that proposition are restricted to a collectively exhaustive, mutually exclusive set, termed the alternative outcomes. Finally, a probability distribution can be used to associate each possible outcome with a probability. We have also shown how probability distributions and outcome sets are expressed for conjunctions of propositions.

Alternatives, the decision maker's options, are expressed in the set of outcomes for a proposition which is the consequent of an informational influence. The fact that a proposition has alternative outcomes and is the consequent of an informational influence defines it as a decision proposition. State descriptions consist of the set of facts and probabilities expressed within or deducible from a domain description. Relationships between states are expressed by the various types of influences available in the language; the logic, probabilistic, and informational influences expressed for the domain. Preferences are handled by identification of a particular proposition whose outcome incorporate the decision maker's objectives. A real valued variable in the proposition is identified as the objective — i.e. the value to be maximized or minimized. A logical influence is defined which is capable of computing this value as a function of other propositions in the domain.

6 Example: A Decision Process

This section presents a simple example, using the decision language to describe a specific subproblem in a decision domain. Consider a security trader dealing in a single instrument, perhaps a particular Treasury security issue or foreign currency. The dealers' task is to trade continually in the instrument in order to make a profit. The traders decisions are what quantity of the security to buy or sell at each instant of the trading day. The fundamental strategy is to 'buy low, sell high,' which is considerably easier to write down than to execute. The dealer's primary uncertainty is that the price of the security will be in the future. Changes in the price are dynamic and dependent on the price in previous periods as well as some other economic conditions or market factors. The trader wishes to maximize his expected profit at some terminal time (Cohen et al. 1985)

The following basic decision alternatives represent the trader's decision to buy, sell, or do nothing (hold) in each trading period. The set of propositions for this situation is shown below along with an interpretation for each. Alternative values for restricted variables are shown in brackets {}. These propositions constitute the means of expressing state descriptions for this domain:

(PROFIT profit time)	Trader's net profit. This is the cumulative total of all the trader's gains and losses in terms of profits since trading was initiated.
(POSITION value time)	Trader's net holding of the security. This is the cumulative total of all the trader's sales and purchases in terms of units of the security.
(TRADE {BUY SELL HOLD} time)	Trader's decision alternatives.

(PRICE {90 91 92} time)	Range of security prices. This is a restriction on the assumed range of prices that the instrument can adopt.
(FUTURES-EXPIRE time)	Futures contract expiration. Futures are contracts for the delivery of a given security at a future date. Standard security future contracts expire on a predetermined date (e.g. the 3rd Friday in March, June, September, etc.). This proposition is true if 'time' occurs on a date when futures contracts mature.
(FUTURES-VOLUME {HEAVY MODERATE} time)	Indicator of activity level for futures markets. The level of activity in futures affects the levels of activity and prices in the 'cash' market (i.e. for current delivery) that is considered in this example.
(GURU{BULLISH BEARISH} time)	Forecast by a market prognosticator or analyst. This represents the information of some outside expert. The 'guru' is 'bullish' if he believes prices are likely to rise, and 'bearish' if prices are thought to fall.

We now describe the set of relationships which characterizes this domain.

The trader's profit and position are simply accounting relations, expressed as deterministic influences. We assume an initial position of zero units of the security, an initial profit of zero dollars, and a single trade quantity of 100 units. The facts (PROFIT 0.0 0)← and (POSITION 0.0 0)← indicate the trader starts with no holdings and no profit.

The net position of the trader is the difference between total sales and total purchases by the trader and depends on the trade made in the current period and net holdings in the previous period.

(POSITION new-position time) ←(– time 1 last-time) Λ
(POSITION old-position last-time) Λ
(TRADE BUY time) Λ
(+old-position 100 new-position)

(POSITION new-position time) ← (– time 1 last-time) Λ
(POSITION old-position last-time) Λ
(TRADE SELL time) Λ
(– old-position 100 new-position)

(POSITION old-position time) ← (– time 1 last-time) Λ
(POSITION old-position last-time) Λ
(TRADE HOLD time)

The profit level at any time is composed of the profit the trader has accumulated so far, plus an adjustment for the amount of the security the trader is holding (net position). If we identify maximizing profit as the objective of the trader, then his preferences among various outcomes (for PRICE, PROFIT, and POSITION) in terms of other propositions are expressed by the logic influence:

(PROFIT new-profit time) ← (— time 1 last-time) Λ
(PROFIT old-profit last-time) Λ
(PRICE old-price last-time) Λ
(PRICE price time) Λ
(POSITION old-position last-time) Λ
(— price old-price change-in-price) Λ

(*old-position change-in-price change-in-value) Λ
(+ old-profit change-in-value new-profit)

The PRICE proposition is the uncertain proposition in this example. The conditional probability distribution for PRICE is expressed by a series of probabilistic influences. A simple influence is:

(PRICE new-price time)$|_p$ (— time 1 last-time) Λ
(PRICE old-price last-time)

$= \pi_p(\omega_{(\text{PRICE new-price time})} | \omega_{(\text{PRICE old-price last-time})})$

This influence expresses a simple stochastic update of price given the previous periods price. If futures contracts for the security expire in a particular period, then the price is independent of the old price and is expressed by a different distribution:

(PRICE new-price time)$|_p$ (FUTURES-EXPIRE time) Λ
(FUTURES-ACTIVITY level time)

$= \pi_p(\omega_{(\text{PRICE new-price time})} | \omega_{(\text{FUTURES-ACTIVITY level time})})$

Note that since (FUTURES-EXPIRE time) is not restricted, the conditional probability distribution π_p will not have separate entries for alternative outcomes of (FUTURES-EXPIRE time), therefore the distribution can be written solely in terms of the alternative outcomes for (FUTURES-ACTIVITY level time). The term (FUTURES-EXPIRE time) is a condition that must be true for the conditional probability distribution $\pi_p()$ to be applicable. Note that the representation allows the expression of several conditional distributions simultaneously, and allows for the expression of conditions regarding which of the alternatives is appropriate by interweaving of deterministic information (e.g. FUTURES-EXPIRE) and uncertain outcomes (e.g. PRICE).

The trader also has information available from the market analyst (the 'GURU') regarding his view of the market is being bullish or bearish. The guru therefore provides the trader with an indicator of overall market trends. The traders opinion of this expert is expressed in the following influence:

(GURU assessment time)$|_p$ (+time 1 next-time)Λ
(PRICE price time) Λ
(PRICE next-price next-time)

$= \pi_p(\omega_{(\text{GURU assessment time})} | \omega_{(\text{PRICE price time})} \wedge_{(\text{PRICE next-price next-time})})$

This influence expresses the trader's probability distribution with respect to the guru's forcast for each possible set of prices for the current and subsequent period (i.e. the alternative outcomes of (PRICE price time)Λ(PRICE next-price next-time)).

The final component indicates the decision in this domain and the information available. The influence states that at time 2 the trader knows the price and guru assessment.

(TRADE action 2)$|_i$ (GURU assessment 2)Λ(PRICE price 2)

For all periods, the trader knows the price when making a trade.

(TRADE action time)$|_i$ (PRICE price time)

These other facts and priors define the state of the knowledge base at a given time. For example:

(PRICE price 0)$|_p \equiv \pi_p(\omega_{(\text{PRICE price 0})})$
(FUTURES-ACTIVITY level time)$|_p \equiv \pi_p(\omega_{(\text{FUTURES-ACTIVITY level time})})$

(FUTURES-EXPIRE 2)←

The first two statements express prior probability distribution for prices at time zero, and futures activity for any period respectively. The last is a fact expressing that futures expire in period 2.

7 Algorithms for ML on Decision Trees

Turning now toward algorithms on machine learning for decision trees in view of Markov or Bayesian networks we could further explore the structure as laid down in the previous sections. ML resembles neural networks for supervised or unsupervised (deep) learning. The decision tree would find the most similar training instances by a sequence of tests on different input attributes. The tree is composed of decision nodes and leaves , starting from the root, each decision node applies a splitting test to the input and depending on the outcome we take one alternating on the branches. Tree learning is nonparametric – the tree grows as needed and its size depends on the complexity of the problem underlying the data for a simple task, the tree is small, whereas a difficult task may grow a large tree. There are different decision tree models and learning algorithms depending on the splitting test used in the decision nodes and the interpolation done at the leaves: one very popular approach nowadays is the random forest.

Proceeding in several key steps reveals the structure of ML algorithms guided by training data in each of these steps.

(a) Starting with time-dependent changes in probability estimates ,i.e.,

$\pi(\omega_i | \omega_j) = \pi(\omega_i ,\omega_j)/ \pi(\omega_j)$ with time dependent conditionalization $\pi(t)(\omega_i | \omega_j) = \pi(t)(\omega_i ,\omega_j)/ \pi(t) (\omega_j)$ with $t = t_o < t < t^*$ and t^* maximum time limit. With chaining of events, as causal chains on multiple events form a network of things with a directed graph of several variables. Consider a distribution π on a Bayesian network of a conditionalized chain such that

$\pi(\omega_1 , \ldots,\omega_n) = \pi(\omega_1), \pi(\omega_2 | \omega_l), \ldots , \pi (\omega_n | \omega_l ,\ldots, \omega_{n-1})$

The joint probability is expressible in terms of a product of n functions.

(b) Expected utility $U = [\pi(t), u(\omega)]$.

In view of moving from probability estimates to calculating expected utility , learning about a random variable Ω can be decided before actually observing its value, i.e. let ω be the value of Ω then the utility of action a will will be

$U(a |\omega) = \sum_\omega u(\omega) \pi(\omega|a , \theta, \Omega=\omega)$

with $\omega|a$ as being consequence of action and θ as evidence or state. Choosing the best among the pending alternatives we get the value $U(\Omega=\omega) = \max_a u(a |\omega)$.

Since we are not sure about the actual outcome of Ω we must average $U(\Omega=\omega)$ over all possible values of Ω weighted by their probabilities. Thus, the utility over all Ω values is

$$U(\Omega) = \sum_\Omega \pi(\Omega=\omega|\theta) (U(\Omega=\omega) .$$

(c) Change of Preference. $u(x_t) > u(y_t) \Leftrightarrow x_t \, P \, y_t$ with x,y given prospects and P a strict preference relation. Inducing some minimal rationality on preferences of given prospects x,y , a weak utility preserving order subject to (i) shift parameter changes of tastes, (ii) tradeoff parameters, lexicographic vs. compensatory preferences, (iii) satisficing or computational constraints (bounded rationality).

(d) Risk Preference/Aversion Profile. Prior specification of utility function as concave (risk averse), convex (risk seeking) or linear (risk neutral) as impacting final decision outcomes.

Thus the outcomes change accordingly if either $U[\sum \pi(t)\, \omega] \geq\leq [\pi(t)\, u(\omega)]$.

(e) Basic Propagation. Dynamic learning would involve an interactive enccoding scheme based on some standardized assumptions: (i) all interactions between variables are linear, (ii)sources of uncertainty are normally (gaussian) distributed and serially uncorrelated, (iii) the influence diagram scheme as a causal network is singly connected. Each new piece of evidence can be viewed as a perturbation that propagates through the network via message- passing between neighboring processors.

Each variable Ω has a set of backward variables , say, X_1, X_2, … , X_n and a set of forward variables, say, Y_1, Y_2, … , Y_m. The relationship could be captured by the linear equation

$$\Omega = b_1 X_1 + b_2 X_2 + \dots ,+ X_n + \varepsilon_\Omega$$

ε_Ω being a stochastic residual component (noise) with zero mean and uncorrelated with any other noise variable.

Each processor Ω is sourced by the following set of parameters

(i) link coefficients installed for interaction between (a) – (d),

(ii) distribution of the random variables originating from Ω,

(iii) backward messages from roots of the tree network from each reverse link to Ω , $\pi_\Omega(x_i)$, I = 1, …, n ,

(iv) forward messages from descending variables of Ω , $\lambda_\Omega(y_j)$, j = 1,…,m .

8 Summary and Conclusions

We assert that the decision making process consists of a series of related activities and assessments with different types of reasoning. The representations and inference methods are designed to provide an integral set of methods for assisting decision makers in these activities and improving performance.

The techniques are grounded on the premise that for effective intelligent decision support, the representation of a decision situation in a computer must reflect the alternatives, beliefs, and preferences of the user of the system. Therefore, the approach developed here focuses on (1) the development of representations and techniques which construct a probabilistic or decision-theoretic model for a particular user, query, and state of information, and (2) support the exploration of alternative representations and models for various phenomena by the user.

Central to the development of powerful computer based decision aids is a means of expressing information and relationships important to describing a particular decision domain. We developed a language based on first order logic for the description of states, alternatives, beliefs, and preferences associated with a decision domain and decision maker. The language includes constructs for explicitly enumerating the alternative possible outcomes for uncertain propositions, probability distributions over these outcomes, the choices facing the decision maker, and his preferences regarding alternative outcomes of differing likelihood. The concept of a logic rule has been generalized to provide for the expression of conditional probabilities (i.e. the probability of a proposition over its alternative outcomes given some conjunctions of propositions is true) and of information availability at the time of decision.

Various deductive inference techniques for reasoning using the first order domain representation are described. The methods are grounded on the dynamic construction of a probabilistic or decision-theoretic model in response to a query and decision domain description in the representation language.

These techniques have the following characteristics which distinguish them from previous approaches:

- Probabilistic and decision theoretic reasoning are integrated with logical, deterministic inference.
- The techniques do not impose assumptions of conditional independence on the probabilistic representation.
- The control structure in the inference methods serves to minimize the size of a probabilistic representation.
- The approach is capable of constructing multiple models for the same phenomena. This enables reasoning about the performance and results of different models within the same environment.

Based on those rules we designed a machine learning algorithm guiding through principles of an IDSM. We could translate the sequential parts into proper codes through Python or Tensaflow.

Additional research areas relate to improving the performance and efficiency of the techniques as implemented. The current system relies on influence diagrams as a formalism for representing decision problems, and the algorithm developed by Shachter(1986) to solve for the optimal sequence of decisions. In the context of the Internet we could highlight intelligent decision procedures through IT platform management and data analytics (Gottinger, 2017).

REFERENCES

[1] Bush,R.R. and Mosteller, F.(1955),Stochastic Models for Learning, John Wiley: New York

[2] Cohen, P. R. (1985) Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach, Pitman, London.

[3] Fishburn, P. C. (1988) Nonlinear Preference and Utility Theory, The Johns Hopkins Univ. Press: Baltimore, MD.

[4] Gershman,S.J., Horvitz,E.J. and Tenenbaum,J.B. (2015), “Converging Rationality: A converging paradigm for intelligence in brains, minds and machines”, Science Magazine 349(6245), July 17, 273-278

[5] Gottinger,H.W and Weimann,P. (1990), Artificial Intelligence, A Tool for Industry and Management , Ellis Horwood: Chichester, Sussex, England

[6] Gottinger,H.W.(2017), Internet Economics: Models, Mechanisms and Management, Bentham Science: London

[7] Holtzman, S. (1989) Intelligent Decision Systems, Addison-Wesley: Reading, Mass.

[8] Howard, R. A. and Matheson, J. E. (1981) “Influence Diagrams”, 1981, In: Howard, R. A. and J. E. Matheson (eds), The Principles and Applications of Decision Analysis, SDG Publications, Strategic Decisions Group: Menlo Park, California, 1984.

[9] Parkes,D.C. and Wellman,M.P.(2015), "Economic Reasoning and Artificial Intelligence", Science Magazine 349(6245), July 17, 273-278

[10] Pearl, J. (1988) Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann: Los Altos, Ca.

[11] Davis, R. (2016) Neural Networks and Deep Learning Explained, AWS: London

[12] Savage, L. J. (1954) The Foundations of Statistics, Wiley Publications: New York.

[13] Shachter, R. D. (1986) 'Evaluating Influence Diagrams', Operations Research 34.