Learning Influence Diagram from Data

Kazuo J. Ezawa
AT&T Laboratories
Room 7E-523
600 Mountain Avenue
Murray Hill, NJ 07974
kje@ulysses.att.com

Narendra K.Gupta
AT&T Laboratories
Room 7F-512
600 Mountain Avenue
Murray Hill, NJ 07974
gupta@ulysses.att.com

Abstract: *There are many cases where decisions are made (and actions are taken) repeatedly under uncertainty, and consequences (results) of those decisions are available. For example, in telecommunications industry repeatedly decisions are made every day for fraud detection and account treatment. Indicators (variables) that have large uncertainties are used to make these decisions. Furthermore, the consequences of such decisions are recorded for later analysis. Similarly, in the financial industry, stocks or currencies are traded based on some indicators (variables). The consequences of these trade can be found. Similarly in the medicine, the patient treatment decisions are made on the basis of the patient information, and the consequences of these decisions to the patients can be found. These data sets contain uncertain variables, decision variables, and value lottery (final outcomes). Furthermore these decisions may not be made not by a single decision maker, but by many decision makers. In contrast to a typical decision analysis, in these environments decisions are made repeatedly. This paper addresses the discovery of knowledge bearing on these decisions in the form of influence diagrams (normative decision models) using a novel supervised machine learning method that constructs Bayesian network models with decisions. Algorithms presented in this paper exploit the goal oriented characteristics of influence diagrams and generate a specific form of influence diagrams that are efficient, both to learn and evaluate. For this reason they are called "efficient" influence diagrams.*

1 Introduction

Many techniques have been developed in learning classification models from data consisting of many uncertain variables. Logistic regression, discriminant analysis, classification trees, neural net, Bayesian network, are a few examples of such techniques. Models generated by these techniques are often used to predict the future events. As such they can not be used as a decision support system. There is, however not much evidence of the work done towards learning decision models from data. In this paper we present an algorithm to learn influence diagrams that can be directly used to compute the best decisions. Like other learning techniques large amounts of training data is needed for learning influence diagrams. Besides the values of other variables, the data must contain the information about the decisions made and their affects on the value of interest. It is not necessary, however, that the data must contain only good decisions, or the decisions made by a single decision maker.

Our algorithm will be most useful for repeated decision problems, i.e., the problems where the same decision has to be taken repeatedly for many instances. In such environments it is hard for human decision makers to stay objective and consistent. Because of high social/economic repercussions of the decisions made in many environments a large repository of historical data essential for training is also available. Some examples of such environments are as follows. In the telecommunications industry repeated decisions are made every day for account treatment. The consequences of such decisions are recorded for later analysis. In the financial industry, stocks and currencies are traded every day, and consequences of these trades are known. In the medical field, the patient treatment decisions are made, and their consequences are later known.

The algorithm presented in this paper uses a novel supervised machine learning algorithm that constructs the Bayesian network models with and without decisions. It exploit the goal oriented characteristics of classification tasks and influence diagrams. It also generates a specific form of influence diagrams that are efficient to evaluate. For this reason they are called "efficient" influence diagrams. We use a goal-oriented

Bayesian network learning algorithm to learn efficient influence diagram. Particularly, we use the Advanced Pattern Recognition and Identification (APRI) system [6,7,8] developed at AT&T Laboratories. Such a learning algorithm, in contrast to general Bayesian network learning algorithms e.g. K2 [1], does not search for the best Bayesian network that fits the data. It searches for the Bayesian network that best predicts the variable values of interest.

In the rest of this section, we provide a brief introduction to influence diagrams, and the APRI training algorithm. We discuss the algorithm for learning efficient influence diagrams in section 2. In section 3, we present an algorithm to evaluate them. In section 4 we summarize the contribution of this work.

1.1 Learning goal oriented Bayesian network

Theoretically, the Bayesian Classifier [9] provides optimal classification performance. As a practical matter, however, its implementation is infeasible. Recent advances in evidence propagation algorithms [13, 11, 12, 10, 3] and computer hardware allow us to approximate the ideal Bayesian classifier by using Bayesian network models [1, 6, 7]. In our experience, a general Bayesian network learning algorithm has a poor classification performance for the real-world problems [8]. The reason for this is that general Bayesian network learning algorithms search for a Bayesian network that best fits the data. In such learning algorithms equal weight is given to the predictive accuracy of all the variables in the domain. In the classification problems we are only interested in predictive accuracy of the class variable. A Bayesian network learning algorithm that learns the network with this goal has been found[8] to have higher predictive accuracy. We call such a learning algorithm the goal oriented Bayesian network learning algorithm.

The classification problem can be solved by using the joint probability $p(V, X)$ of classes or populations V and the variables X that describe the data[1]. In particular, an observation X can be classified as an instance of class V, if V is the most probable according to the conditional probability distribution $p(V|X)$. Assessing $p(V|X)$ directly is often infeasible due to data and storage limitations. Instead the conditional probability of the attributes given the classes, $p(X|V)$, and prior distribution of the classes, $p(V)$, are assessed by analyzing a pre-classified training data set. With these probabilities in hand, Bayes rule then yields the desired conditional probability $p(V|X)$.

Merely representing $p(V, X)$ can be difficult when there are a large number of variables, particularly, if the distribution does not have a convenient structure. Bayesian Networks can be used to encode a wide variety probability distributions by factoring them into possibly smaller sizes. The variables in the Bayesian networks generated by the goal oriented learning algorithm have a common parent, namely the class node V. This is not true of all Bayesian networks. It is however, a feature of networks learned by the goal oriented Bayesian network learning algorithm that helps them address the classification problem at hand.

The Advanced Pattern Recognition & Identification (APRI) system developed at AT&T Laboratories is a goal oriented Bayesian network learning system. APRI constructs Bayesian network like the one described above. It uses the mutual entropy to perform dependency selection [2]. It uses thresholds on mutual entropy, first, to select a set of variables and then to select a set of dependencies among the chosen variables.

1.2 APRI training algorithm

APRI constructs graphical probability models in a four-step process. Three inputs are required: a database of training cases and two parameters, $T_{\pi x}$ and T_{xx}, each ranging between zero and one. $T_{\pi x}$ governs variable selection (or equivalently, selection of links between the class node and the variable nodes). T_{xx} governs selection of variable-to-variable links.

In the first step, APRI parses the input database and characterizes its variables. If the class variable is continuous, APRI first defines the class outcomes either by discretization or kernel density estimation. APRI then scans the database to identify the outcome sets for each variable. For continuous variables it either estimates the kernel density or uses information-based discretization.

In the second step, APRI chooses the variables for the final model. It computes the mutual information between the class node and the individual variables, then ranks the variables accordingly. We represent mutual information between the class variable π and

1. Bold-faced capital letters will be used to denote vectors of features or attributes, such as an entire observations.

an independent variable X by $I(\pi;X)$.

Without loss of generality, let the indices from 1 to K provide the mutual-information ranking of the initial variables, so that $I(\pi;X_1) \geq I(\pi;X_2) \ldots \geq I(\pi;X_k)$. APRI selects the top J variables out of this ranking, such that:

$$\sum_{j=1}^{J} I(\pi;X_j) \geq T_{\pi x} \sum_{k=1}^{K} I(\pi;X_k) \qquad (1)$$

Thus, the parameter $T_{\pi x}$ establishes a mutual information threshold for choosing relevant variables. A value of 1 indicates that all the variables should be incorporated in the model. Values less than 1 indicate that the least informative variables should be excluded. In the graphical models generated by APRI, the class node becomes the parent of each of the selected variables.

The third step is akin to the second one, save that it identifies relationships between variables. In particular, it computes the conditional mutual information $I(X_i;X_j|\pi)$ between pairs of the J previously identified variables, where $i \neq j$. These candidate links are rank ordered. The highest ranked are then selected until the cumulative value is just T_{xx} times the total conditional mutual information. Directionality of these links is based on the mutual information variable ranking determined in the second step, with higher ranked variables pointing towards lower ranked ones.

In the fourth and final step, APRI estimates $p(\pi)$ and $p(X_i|C(X_i))$ using frequency counts, where $C(X_i)$ represents the parents or predecessors of X_i, including the class node π.

1.3 Influence Diagrams

An influence diagram is a graphical representation of a decision problem under uncertainty that explicitly captures the probabilistic dependence and the flow of information. It is a mathematically precise description of the problem, and can be directly evaluated [14, 3, 5] to compute the best set of decisions.

An influence diagram is an acyclic directed graph with four types of nodes (representing different types of variables) and two types of arcs (representing different types of relationships between nodes).

A circle symbolizes a chance node (*E*) and describes uncertain "event" using mutually exclusive potential outcomes with their associated probabilities. A heavy circle symbolizes a deterministic node (*F*) which represents functional relationships between nodes (variables). It contains a deterministic function that describes the functional relationships. A square symbolizes a decision node (*D*) which represents a decision variable for the decision maker and contains decision alternatives. A diamond symbolizes a value node (*V*) which represents the goal of the decision problem and contains the value function which measures the quality of the final outcome. In this paper ***X*** denotes all variables. ***D*** denotes all decision variables, ***E*** chance variables, and ***F*** deterministic variables in the influence diagram.

An arc into a chance node is called a conditional arc, and indicates that there is a probabilistic dependency between the node and its predecessor(s). An arc into a decision node is an informational arc, It indicates that before a decision maker makes a decision, (s)he has the information related to its predecessor(s). In this paper we will also call these predecessor(s) as observable variables.

A successor of node *I* is a node on a directed path emanating from node *I*. A successor node that is adjacent to node *I* is called a direct successor of the node *I* and is denoted as *S(I)*.

A predecessor of node *I* is a node on a directed path terminating at node *I*. A predecessor that is adjacent to node *I* is called a direct predecessor of node *I* and is denoted as *C(I)*. Sets of set of predecessor nodes for nodes in ***D*** will be represented by ***IP***.

Each node I has an associated variable X_I, outcome space Ω_I, and value x_i which represents a particular outcome from Ω_I. A subset of Ω_I is denoted by $\boldsymbol{x_i}$. $p(X_I)$ represents the probability distribution of the conditionally independent variable X_I. $p(X_I|X_J)$ represents the probability distribution of conditionally dependent variable X_I given X_J.

Evaluation of an influence diagram gives the best policy and its expected utility value. Evaluation procedure requires three standard operations in influence diagram. Objective of each of these operations is to account for the effect of a variable in computing the best policy and remove it from the influence diagram. As a result of evaluation the influence diagrams are reduced to a single value node. The first operation is chance node removal which is to

take the expectation of the joint probability given the chance node. The second operation is the decision node removal which is to take the maximization of the expected value (value function in the value node) given the alternatives of the decision node. The last operation is arc reversal that changes the direction of an arc by application of Bayes rule.

Figure 1 shows an example of an influence diagram for an account treatment decision problem. In this example the order in which decisions are made is fixed, i.e. the decision to investigate is made before decision to take an action is made. An influence diagram that has a fixed temporal order on decision nodes is called a regular influence diagram.

Influence diagrams learned by our algorithm are efficient to evaluate because they are in an almost reduced form. They contain only the decision variables and their informational predecessors (variables that the decision maker can observe before making the decision. We will also call such variables as observable[1]). The impact of all other uncertain variables is represented in the final outcome distribution (value lottery).

In spite of their compact representation efficient influence diagrams allow full-fledged analysis, e.g., sensitivity analysis, the value of perfect information and the value of control.

Before proceeding we would like to remind the readers that an influence diagram represents an approximation and an abstraction of a decision problem. Traditionally in decision analysis, the creation of influence diagram is a very subjective process. It involves manually quantitatively specifying outcomes, selecting key variables and their dependencies, and assessing probabilities. In such setting it encourages clearer, more precise thinking, allows for the incorporation of the expert knowledge, and creates a common framework for understanding, analyzing, and communicating the decision problem. In learning an influence diagram from data, we replace this human judgement by the mutual information based algorithm. Both methods create an influence diagram which is an approximation of the decision problem. There's no "gold-standard" influence diagram given the decision problem.

1. Other variables may affect the final out come but may not be observable by the decision maker. Some of them are affected by the decision made by the decision maker and others not.

1.4 Evidence Propagation in Bayesian Networks and Influence Diagrams

The instantiation of evidence on a chance node and its propagation among chance nodes involves the following operations [4]. These definitions are slightly different from that of [13]:

- Evidence absorption: instantiation of evidence x_j on node J which is just the table lookup of the observed outcome, i.e., $p(X_j = x_j | X_{C(J)})$.
- Evidence propagation (forward): propagation of evidence x_j to its successor node *I*, which is the identification of still valid potential outcomes, i.e., $p(X_I | X_{C(I)} \wedge X_j = x_j)$.
- Evidence reversal (backward): evidence absorption of x_j on node *J* and arc reversal between *J* and its predecessor *C(J)* and the propagation of evidence x_j to *C(J)*.

2 Learning Efficient Influence Diagrams

In an influence diagram the outcome space of the value node with all its predecessors may become very large, and may need large storage. Furthermore, there is no use of keeping the unobserved variables in an unreduced form of influence diagram. We describe here an algorithm that makes a deliberate effort to learn an influence diagram from data in an almost reduced form.

2.1 Assumptions

Our algorithm for learning efficient influence diagrams, makes the following assumptions:

- Training data consists of decision, chance, and objective (value node) variables. Furthermore, the decision, chance, and objective variables are identified.
- All observable variables (informational predecessors) for each decision are known.
- Training data consists of all kinds of decisions, i.e. both good and bad decisions.

Next, we will provide the main algorithm, and illustrate the operation of our algorithm with the help of an example.

2.2 Algorithm

Our algorithm to learn efficient influence diagrams

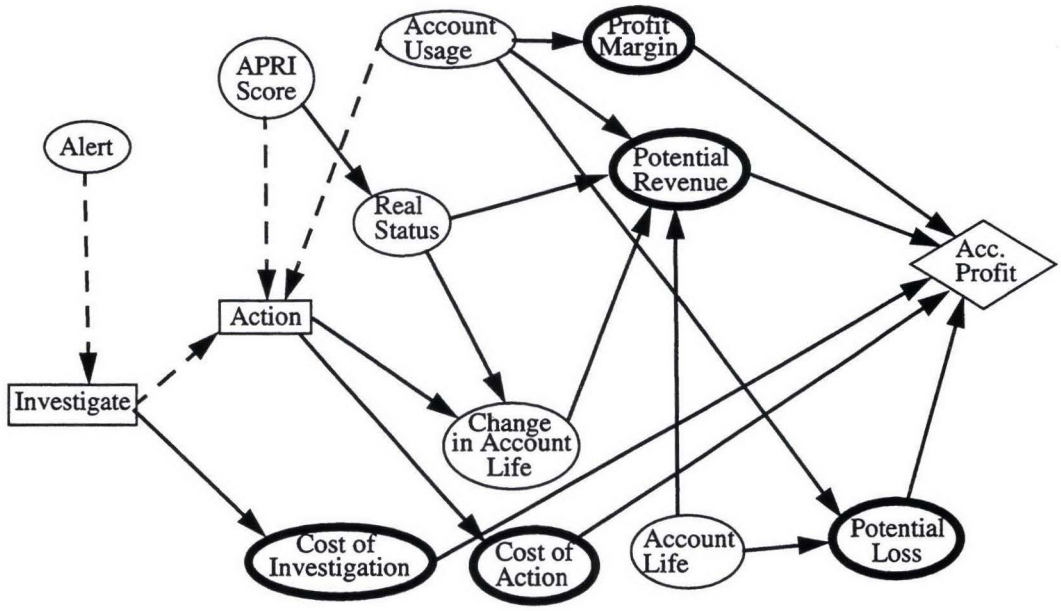

FIGURE 1. Account Treatment Decision Model: an example.

is as follows.

Algorithm 1:

Input: Training data
C: Chance variables
D: List of decision variables.
V: Value
***IP*:** Informational predecessors

Output: Influence diagram *G* with value Lottery on the value node.

1. Add a chance node corresponding to each variable in ***C, D*** and *V* to *G*.
2. Designate *V* as prime node (APRI terminology for the class variable).
3. Learn the goal oriented Bayes network with the constraint that no arcs can be directed towards the nodes corresponding to decisions. And no arc can be directed from the decision node towards its informational predecessors.
4. Reverse Arcs from the prime node to the nodes corresponding to decisions. Drop any introduced arcs that are directed towards any of the decision nodes
5. Convert nodes corresponding to decision nodes to decision node.

Due to step 3, arc reversal in step 4 will not introduce any cycles. Therefore the algorithm is guaranteed to produce a valid influence diagram.

We will illustrate this algorithm with the help of an example. In this example we intend to learn an influence diagram like the one shown in figure 1. Notice that the influence diagram in figure 1has a number of deterministic nodes. These variables will not be present in our data set.

Figure 2 shows the state of the influence diagram after step 3 in the algorithm. As mentioned in the algorithm, no arc is added that points towards the nodes "Investigate" or "Action". Also no arc is added from "Investigate" to "Alert" or from "Action" to "APRI score". Notice however, that to capture the conditional dependence embedded in the influence diagram in figure 1, APRI introduces arcs between "Account Usage" and "Account Life" and between "APRI Score" and "Account Usage"

In step 4 we perform arc reversal from the nodes "Investigate" and "Action" to the node "Acc. Profit". Finally after step 5 the output of the algorithm therefore will be as shown in figure 3.

There are two important features of the efficient influence diagram output by algorithm 1. First, there are no informational arcs present. Given informational predecessors ***IP***, it may even be possible to transform this influence diagram such that informational arcs could be added without introducing a directed cycle[1].

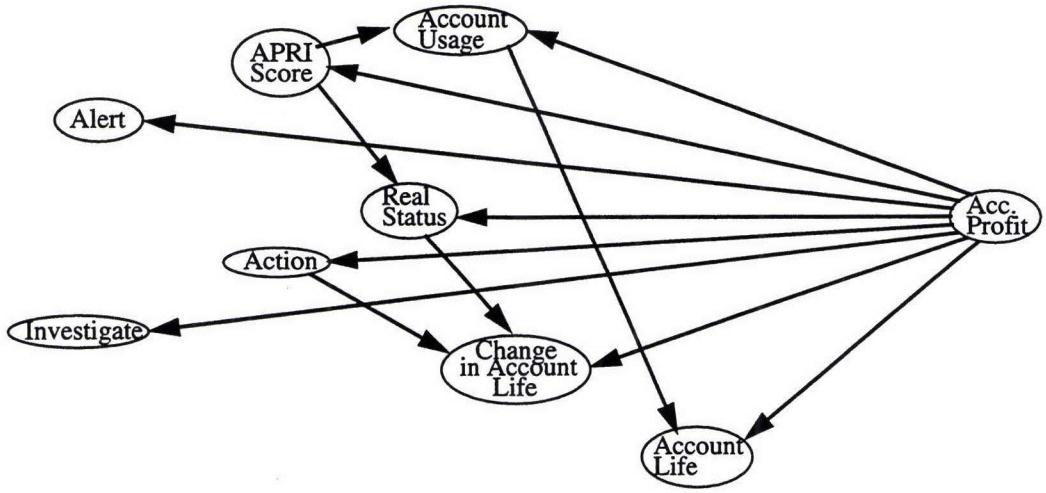

FIGURE 2. An example after execution of step 3 of algorithm 1.

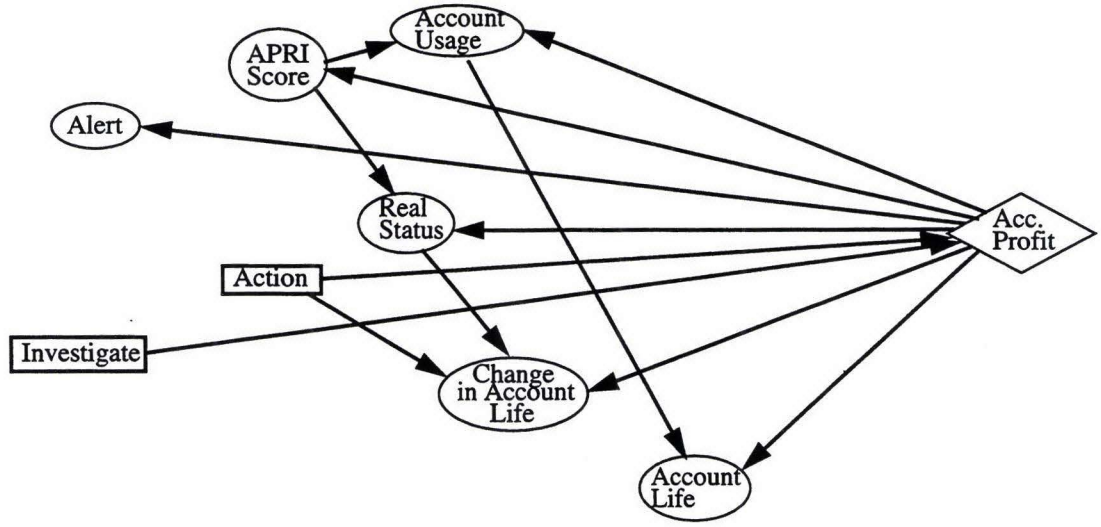

FIGURE 3. Influence diagram output by algorithm 1.

A two step evaluation algorithm (detailed in the next section) first instantiates the values of the informational predecessors, propagates them in the network and then reduces the influence diagram to deduce the best decision.

The second feature of the learned influence diagram is that all the chance nodes can be deleted by removing the barren nodes. As mentioned above, however, we need the informational predecessors for correct evaluation of the influence diagram. Uncertainty in other chance nodes, is already accounted for by the learned value lottery at the value node. In our example, if chance nodes that are not informational predecessors are removed, we will be left with the influence diagram shown in figure 4.

1. Notice that direct addition of the informational arcs in the influence diagram generated by algorithm1 may result in directed cycles.

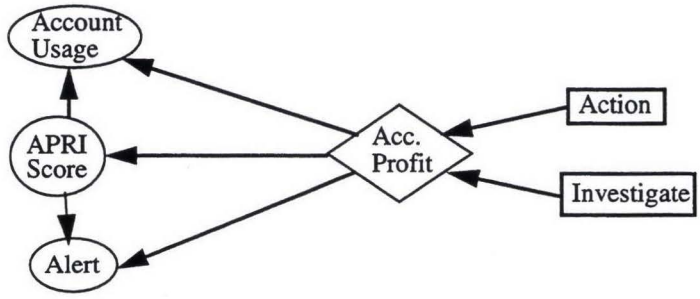

FIGURE 4. Influence diagram generated by removing the barron chance nodes that are not informationalpredecessors.

3 Evaluating Efficient Influence Diagrams

Efficient influence diagrams as shown in figure 4 become quite compact in comparison with the one in Figure 1. For use in a normative expert systems this compactness provides two advantages. First it requires much smaller space to store the model, and second it gains run-time efficiency for evaluation of efficient influence diagrams.

To use an influence diagrams in a normative expert system, or to do sensitivity analysis we need to propagate evidence before the evaluation of the influence diagram. Algorithm 2 is used for evidence propagation and evaluation of influence diagrams both for the efficient influence diagrams and for sensitivity analysis.

Evidence propagation on a chance variable requires evidence reversal. It involves an arc reversal between the chance node and the classification (prime) node. After the arc reversal, the chance node will inherit the decision variables. To make sure that no cycles are introduced arc reversal should be performed in a sequence consistent with ***IP***. As mentioned above no special treatment is needed either for the observable or for the non-observable variables.

Algorithm 2: Evaluation of influence diagrams

Input: $\boldsymbol{x}$: Evidence chance nodes.
IP: Informational predecessors (observable chance nodes).
G. An influence diagram learned by algorithm 1 with ***C, D*** and *V* nodes.

Output: Value lottery on the node *V* that accounts for the resolved uncertainty in $\boldsymbol{x}$, and best set of decision given $\boldsymbol{x}$.

1. Generate an order $\{D_1, D_2, \ldots D_k\}$ on $\boldsymbol{D}$ which is consistent with ***IP***.
2. for $i = 1$ to k, do
3. for $E_{ij} \in \{(E_j = x_j) \in \boldsymbol{x} | (E_j \in C(D_i))\}$ do
4. If E_{ij} has a chance variable successor, perform forward evidence propagation.
5. If E_{ij} has a chance variable predecessor, perform evidence reversal.
6. for $i = k$ to 1,
7. Reduce D_i by setting
$$p(V|(\boldsymbol{D}/D_i)) = \underset{D_i}{Max}(p(V|\boldsymbol{D}))$$

This is equivalent to the evidence node reduction after the evidence absorption and arc reversal between the evidence node and the value node.

4 Discussion and Summary

This paper addressed the discovery of knowledge bearing on decisions in the form of Bayesian networks and influence diagrams using a novel supervised machine learning method that constructs Bayesian network models with and without decision variables, i.e., APRI, which creates a goal oriented Bayesian network. It is suitable for classification tasks and creation of influence diagrams. For decision support, this new form of influence diagram is efficient in both learning a model from a data as well as evaluating the model for decision recommendations given observed

information. Since the normative models learn the best strategies from data as well as integrates different experts best strategies, it may surpass any individual decision makers performance.

References

[1] Cooper, G. F. and Herskovits, E. (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning*, **9**, pp. 309-347.

[2] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*, John Wiley and Sons.

[3] Ezawa, K. J. (1986). Efficient Evaluation of Influence Diagrams, *Ph.D. Thesis*, Dept. of Engineering-Economic Systems, Stanford University.

[4] Ezawa, K. J. (1994). Value of Evidence on Influence Diagrams, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 212-220, Morgan Kaufmann.

[5] Ezawa, K. J., and Scherer, J. B. (1992). Technology Planning for Advanced Telecommunications Services: A Computer-Aided Approach. *Telematics and Informatics*, 9(2), pp. 101-112.

[6] Ezawa, K. J., and Schuermann, T. (1995). Fraud/ Uncollectible Debt Detection Using a Bayesian Network Based Learning System: A Rare Binary Outcome with Mixed Data Structures, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 157-166, Morgan Kaufmann.

[7] Ezawa, K. J., and Norton S. (1995). Knowledge Discovery in Telecommunication Services Data Using Bayesian Networks, Proceedings of the *First International Conference on Knowledge Discovery & Data Mining*, Montreal, Canada, August 1995.

[8] Ezawa, K. J., Singh M., and Norton, S. (1996). Goal Oriented Bayesian Network Model Comparisons Using Telecommunications Risk Management Data sets, *Proceedings of the 13th International Conference on Machine Learning*.

[9] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press.

[10] Jensen, V., Olesen K. G., and Anderson S. K. (1990) An Algebra of Bayesian Universes for Knowledge-Based Systems, *Networks*, 20, pp. 637-659.

[11] Lauritzen, S. L., and Spiegelhalter, D. J. (1988). Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems, *J. R. Statistics Society*, Vol. **50**, No. 2, pp. 157-224.

[12] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann.

[13] Shachter, R. D. (1990). Evidence Absorption and Propagation through Evidence Reversals, *Uncertainty in Artificial Intelligence*, Vol. **5**, pp. 173-190, North-Holland.

[14] Shachter R. D. (1986). Evaluating Influence Diagrams. *Operations Research*, Vol. **34**, No. 6,.