A Simple Method to Evaluate Influence Diagrams

Y. Xiang and C. Ye
Dept. of Computing and Information Science
University of Guelph, Guelph, Ontario, Canada N1G 2W1

(Published in Third International Conference On Cognitive Science, 2001)

August 9, 2001

Abstract

Influence diagrams (IDs) provide a concise graphical formulation for decision making in intelligent agents. In this work, we propose a simple method to evaluate IDs by first converting IDs into BNs and then applying a junction tree-based inference algorithm. The methods are more efficient than methods in [1, 5] and are simpler than [6]. (Keywords: Artificial intelligence, intelligent decision support systems, influence diagrams, Bayesian networks, uncertain reasoning.)

1 Introduction

Influence diagrams (IDs) [3] provide a concise graphical formulation of many decision problems and has been actively studied as a tool for decision making in intelligent systems. An ID represents a decision problem by three types of varialbes. Chance variables represent the uncertainty of the problem domain. Decision variables represent the alternative actions of the decision maker. A single value variable represent the preference of the decision maker for each possible consequence. The dependence among the variables are expressed as an acyclic graph where each node corresponds to a variable. The strength of the dependence between a chance variable and others is represented by a conditional probability distribution (CPT) associated with the chance node. The decision maker's preference is represented by a utility distribution associated with the value node. Once an ID is constructed, it can be used to derive a *policy* which specifies what action the decision maker should take for each decision in order to obtain the maximum expected utility (MEU). We refer to the computation of the optimal policy for an ID as *evaluating* the ID.

IDs are extensions of Bayesian networks (BNs) [4] which consist of chance nodes only. BNs provide a concise graphical representation for reasoning about uncertain domains. A set of algorithms have been developed for probabilistic inference in BNs. As BNs and IDs share many commonalities, methods have been proposed to use BN inference algorithms to evaluate IDs [1, 5, 6].

In this work, we propose a simple method to evaluate IDs by first converting IDs into BNs and then applying a junction tree-based inference algorithm [2]. The methods are more efficient than methods in [1, 5] and are simplier than [6].

2 Evaluating IDs with a single decision

First, we consider the problem of evaluating an ID where there is a single decision node d_1. We denote the space of d_1 by $D_{d_1} = \{d_{1,1}, d_{1,2}, ...\}$. We denote the parent nodes of d_1 in the ID by π_{d_1}. The π_{d_1} includes those variables whose values are observed before the decision d_1 is made. A single value node v is associated with a utility function $v(\pi_v)$, where π_v refers to the parent nodes of v. That is, $Max(v(\pi_v)) = 1$ and $Min(v(\pi_v)) = 0$.

To evaluate an ID using a BN inference algorithm, the ID needs be converted into a BN. We denote a BN by (V, G, P) where V is the set of variables, $G = (V, E)$ is a DAG, and P is a set of conditional probability distributions one associated with each node in G. Using the method proposed by Shachter and Peot [5], we convert an ID into a BN as follows:

- All existing chance nodes remain the same.
- Convert the decision node d_1 into a chance node. It has the same space D_{d_1} and the same set of parent nodes π_{d_1}. It is associated with a uniform distribution $P(d_1|\pi_{d_1}) = const$, where *const* stands for a normalizing constant such that $\sum_i P(d_{1,i}|\pi_{d_1}) = 1$ for each configuration of π_{d_1}.
- Convert the value node into a binary chance node u with a space $\{0, 1\}$, the same parents $\pi_u = \pi_v$, and a distribution $P(u = 1|\pi_u) = v(\pi_v)$.

The following proposition establishes how computation of posterior probability in a converted BN corresponds to the evaluation of expected utility in its deriving ID. It forms the basis of our method.

Proposition 1 *Let (V, G, P) be a BN converted from an ID with a single decision node d_1. For each configuration c of π_{d_1} and each value $d_{1,i}$ of d_1, we have*

$$P(u = 1|c, d_{1,i}) = E(v|c, d_{1,i}),$$

where $P(u = 1|c, d_{1,i})$ is a probability determined by the BN, and $E(v|c, d_{1,i})$ is the expected utility given c and $d_{1,i}$ determined by the ID.

Proof:

Denote $V' = V \setminus \{u, d_i, \pi_{d_1}\}$ and $V'' = V' \setminus \pi_u$. We have

$$\begin{aligned}
P(u = 1|c, d_{1,i}) &= \sum_{V'} P(u = 1|V', c, d_{1,i}) P(V'|c, d_{1,i}) \qquad \text{(marginalization)} \\
&= \sum_{\pi_u} \sum_{V''} P(u = 1|V'', \pi_u, c, d_{1,i}) P(V'', \pi_u|c, d_{1,i}) \\
&= \sum_{\pi_u} \sum_{V''} P(u = 1|\pi_u) P(V'', \pi_u|c, d_{1,i}) \qquad \text{(conditional independence)} \\
&= \sum_{\pi_u} P(u = 1|\pi_u) \sum_{V''} P(V'', \pi_u|c, d_{1,i}) \\
&= \sum_{\pi_u} P(u = 1|\pi_u) P(\pi_u|c, d_{1,i}) = \sum_{\pi_u} v(\pi_v) P(\pi_u|c, d_{1,i}) \quad \square
\end{aligned}$$

Given Proposition 1, it follows that the action $d_{1,*} \in D_{d_1}$ that maximizes $P(u = 1|c, d_{1,i})$ is the best action when c is observed, and $P(u = 1|c, d_{1,*})$ is then the maximum expected utility (MEU). This is the conclusion derived in [1] (using a slightly different ID conversion). However, neither [1] nor [5] shows Proposition 1 which is more general than the MEU conclusion itself.

According to [1], this method can be applied for each configuration of π_{d_1} and each action $d_{1,i}$ to obtain the optimal policy $\delta_1(\pi_{d_1})$ for decision d_1. Denote the *family* of d_1 by $fam(d_1) = \{d_1\} \cup \pi_{d_1}$, and denote the cardinality of the space of $fam(d_1)$ by n. Then the total number of inference computations of $P(u = 1|c, d_{1,i})$, one for each configuration of $fam(d_1)$, is n. Hence the complexity of this method is $2^{|fam(d_1)|}$. To avoid the exponential number of computations, we consider an alternative computation:

For each value $d_{1,i}$ of d_1 and for each configuration c of π_{d_1}, we have

$$P(u = 1|c, d_{1,i}) = \frac{P(c, d_{1,i}|u = 1)P(u = 1)}{P(d_{1,i}|c)P(c).} \tag{1}$$

Given c, $P(c)$ is a constant. $P(d_{1,i}|c)$ is a constant due to the uniform distribution assigned. $P(u = 1)$ is also a constant. Therefore, to find $d_{1,*}$ that maximizes $P(u = 1|c, d_{1,i})$ for a given c, we can alternatively maximizes $P(c, d_{1,i}|u = 1)$.

Computation of $P(c, d_{1,i}|u = 1)$ is more advantageous than that of $P(u = 1|c, d_{1,i})$ due to the following. We can convert the BN into a junction tree representation [2]. Given a BN over the set V of variables, a junction tree (JT) representation is a tree where each node is labeled by a subset of V, called a *cluster*. The conditional probability distributions in the BN are converted into the distribution over each cluster. An important property of the JT is that for each node in the BN, its family is contained in at least one cluster in the JT. Computation of posterior distributions over each cluster can be performed effectively by message passing between clusters, called *belief propagation*. For more details on the JT representation and belief propagation, see [2]. Since $fam(d_1)$ is contained in a single cluster, $P(\pi_{d_1}, d_1|u = 1)$ can be computed by just one belief propagation with observation $u = 1$. Then the optimal policy for d_1 is obtained by finding $d_{1,*}$ that maximizes $P(c, d_{1,i}|u = 1)$ for each c, and repeat the processing for each configuration c of π_{d_1}. This is the method suggested in [5]. Since only one belief propagation is needed, this is much more efficient than Cooper's method.

For each configuration c of π_{d_1}, the MEU of $d_{1,*}$ is $P(u = 1|c, d_{1,*})$. According to Equation 1, it can be obtained if, in addition to $P(c, d_{1,i}|u = 1)$, we have $P(d_1, \pi_{d_1})$ and $P(u = 1)$. Both can be obtained from the junction tree representation of the BN by one belief propagation without any observation. We denote the maximum expected utility distribution for the optimal policy $\delta_1(\pi_{d_1})$ as $meu(\pi_{d_1}, \delta_1(\pi_{d_1}))$.

3 Evaluating IDs with sequential decisions

In [5], the dynamic programming is suggested for evaluation of IDs with sequential decisions. We propose a new method that does not need dynamic programming and hence is simpler to implement and is more efficient to execute.

We denote the decision varialbes in an ID as

$$d_1, d_2, ..., d_{n\ \ 1}, d_n,$$

where d_1 is the first decision and d_n is the last decision. We denote the parent of d_i as π_i instead of π_{d_i} for simplicity. A common assumption for IDs is *non-forgetting* : For each decision d_i, $fam(d_j) \subset \pi_i$ holds for each $j < i$. This implies that the parent set π_n of the last decision d_n contains all other decision variables plus their parent sets.

Before presenting our method, we first review briefly how to evaluate an ID using dynamic programming [5]. To evaluate an ID using the junction inference algorithm, we convert the ID into a BN. The conversion is similar to the above except each decision node d_i is converted into a chance node. It has the same space D_{d_i} and the same set of parent nodes π_i. It is associated with a uniform distribution $P(d_i|\pi_i) = const$. Afterwards, we convert the BN into a JT representation.

To evaluate the ID, the dynamic program approach works backwards from the last decision to the first decision. We first compute the optimal policy for the last decision d_n as before by computing $P(d_n, \pi_n|u = 1)$, $P(d_n, \pi_n)$, $P(u = 1)$, the policy $\delta_n(\pi_n)$, and the max expected utility $meu(\pi_n, \delta_n(\pi_n))$. To derive the optimal policy for $d_{n\ 1}$, we first replace the uniform distribution $P(d_n|\pi_n)$ at d_n in the BN by a conditional probability distribution that is consistent with the policy $\delta_n(\pi_n)$:

$$P'(d_n|\pi_n) = \begin{cases} 1 & \text{if } d_n = \delta_n(\pi_n) \\ 0 & \text{otherwise} \end{cases}$$

We denote any distribution obtained from the new BN by $P'()$. We then convert the new BN into a JT representation and compute the optimal policy for $d_{n\ 1}$ in the same way as for d_n. This process is repeated until the optimal policy for d_1 is obtained. This is the approach of dynamic programming.

The dynamic programming approach requires a modification of the converted BN, the compilation of the BN into a JT, and two belief propagations in the JT for each decision node. In the following, we present a new method that computes $\delta_{n\ 1}(\pi_{n\ 1}), ..., \delta_1(\pi_1)$ directly from $P(d_n, \pi_n|u = 1)$, as described below:

In order to compute $\delta_{n\ 1}(\pi_{n\ 1})$, we need to obtain $P'(d_{n\ 1}, \pi_{n\ 1}|u = 1)$. Using the product rule, we have

$$\begin{aligned} P(d_n, \pi_n|u = 1) &= P(d_n, \pi_n)P(u = 1|d_n, \pi_n)/P(u = 1) \\ &= P(d_n|\pi_n)P(\pi_n)P(u = 1|d_n, \pi_n)/P(u = 1). \end{aligned}$$

We consider the impact of replacing $P(d_n|\pi_n)$ with $P'(d_n|\pi_n)$ in the dynamic programming. Since π_n contains ancestors of d_n, $P(\pi_n)$ is not affected by the replacement ($P'(\pi_n) = P(\pi_n)$). $P(u = 1|d_n, \pi_n)$ is also not affected ($P'(u = 1|d_n, \pi_n) = P(u = 1|d_n, \pi_n)$). $P(u = 1)$ is a normalizing constant. Therefore, after the replacement, we have

$$P'(d_n, \pi_n|u = 1) = P'(d_n|\pi_n)P(\pi_n)P(u = 1|d_n, \pi_n)/P'(u = 1).$$

Since $P'(d_n|\pi_n)$ is either 0 or 1, to obtain $P'(d_n, \pi_n|u = 1)$ from $P(d_n, \pi_n|u = 1)$, we have

$$P'(d_n, \pi_n|u = 1) = \begin{cases} \frac{P(d_n, \pi_n|u=1)P(u=1)}{P(d_n|\pi_n)P'(u=1)} & \text{if } d_n = \delta_n(\pi_n) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Since $P(d_n|\pi_n)$ is a uniform distribution, we have

$$P'(d_n, \pi_n|u=1) = \begin{cases} const * P(d_n, \pi_n|u=1) & \text{if } d_n = \delta_n(\pi_n) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $const$ is a normalizing constant. Since

$$P'(d_n, \pi_n|u=1) = P'(d_n, d_{n\ 1}, \pi_{n\ 1}, \pi_{n}|u=1),$$

where $\pi_{n} = \pi_n \setminus \{\pi_{n\ 1}, d_{n\ 1}\}$, we have

$$P'(d_{n\ 1}, \pi_{n\ 1}|u=1) = \sum_{d_n, \pi_n} P'(d_n, \pi_n|u=1).$$

From $P'(d_{n\ 1}, \pi_{n\ 1}|u=1)$, we can obtain $\delta_{n\ 1}(\pi_{n\ 1})$. Repeating the above process for $d_{n\ 2}, ..., d_1$, we can obtain $\delta_{n\ 2}(\pi_{n\ 2}), ..., \delta_1(\pi_1)$.

In order to compute $meu(\pi_{n\ 1}, \delta_{n\ 1}(\pi_{n\ 1}))$ for $\delta_{n\ 1}(\pi_{n\ 1})$, we use

$$P'(u=1|\pi_{n\ 1}, d_{n\ 1}) = \frac{P'(\pi_{n\ 1}, d_{n\ 1}|u=1)P'(u=1)}{P'(\pi_{n\ 1}, d_{n\ 1})}. \quad (4)$$

Since $P'(\pi_{n\ 1}, d_{n\ 1}) = P(\pi_{n\ 1}, d_{n\ 1})$, it can be obtained from the original BN. Comparing Equations 2 and 3, we can obtain $P'(u=1)$ from $const$, $P(d_n|\pi_n)$ and $P(u=1)$ as follows:

$$P'(u=1) = P(u=1)/(const * P(d_n|\pi_n)). \quad (5)$$

Note that $P(d_n|\pi_n) = 1/|D_{d_n}|$ where D_{d_n} is the space of d_n, since $P(d_n|\pi_n)$ is uniform. We define

$$P^*(d_n, \pi_n|u=1) = \begin{cases} P(d_n, \pi_n|u=1) & \text{if } d_n = \delta_n(\pi_n) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

which can be obtained by setting some terms of $P(d_n, \pi_n|u=1)$ to zero without normalization. By combining Equations 3 through 6, we can compute $P'(u=1|\pi_{n\ 1}, d_{n\ 1})$ as follows:

$$P'(u=1|\pi_{n\ 1}, d_{n\ 1}) = \frac{P^*(\pi_{n\ 1}, d_{n\ 1}|u=1)P(u=1)\ |D_{d_n}|}{P(\pi_{n\ 1}, d_{n\ 1})}. \quad (7)$$

For each decision d_i, although $\delta_i(\pi_i)$ specifies the action for each configuration of π_i, some configurations of π_i will never be used due to the policies for $\delta_{i\ 1}(\pi_{i\ 1}), ..., \delta_1(\pi_1)$. After $\delta_n(\pi_n), ..., \delta_1(\pi_1)$ have been obtained, we compute the *optimal execution policy* $\delta'_1(\pi_1), ..., \delta'_n(\pi_n)$ as follows:

Let $\delta'_1(\pi_1) = \delta_1(\pi_1)$. For the second decision d_2,

$$\delta'_2(\pi_2) = \delta_2(d_1, \pi_1, \pi_{2\ }) \ \text{ if } d_1 = \delta_1(\pi_1)\ .$$

Otherwise, $\delta'_2(\pi_2)$ is undefined.

For the i'th decision d_i $(i > 1)$,

$$\delta'_i(\pi_i) = \delta_i(d_{i\ 1}, \pi_{i\ 1}, \pi_{i\ })$$

if $\delta'_{i\ 1}(\pi_{i\ 1})$ is well-defined and $d_{i\ 1} = \delta'_{i\ 1}(\pi_{i\ 1})$. Otherwise, $\delta'_i(\pi_i)$ is undefined.

Acknowledgement

This work is supported by NSERC. Assistance in implementation from X. An, Y. Wang and Q. Chao is acknowledged.

References

[1] G.F. Cooper. A method for using belief networks as influence diagrams. In R.D. Shachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, editors, *Proc. 4th Workshop on Uncertainty in Artificial Intelligence*, pages 55–63, 1988.

[2] F.V. Jensen. *An Introduction To Bayesian Networks*. UCL Press, 1996.

[3] R.M. Oliver and J.Q. Smith, editors. *Influence Diagrams, Belief Nets and Decision Analysis*. John Wiley, 1990.

[4] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[5] R.D. Shachter and M.A. Poet. Decision making using probabilistic inference methods. In D. Dubois, M.P. Wellman, B. D'Ambrosio, and P. Smets, editors, *Proc. 8th Conf. on Uncertainty in Artificial Intelligence*, pages 276–283, Standford, CA.

[6] N.L. Zhang. Probabilistic inference in influence diagrams. In G.F. Cooper and S. Moral, editors, *Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, pages 514–522, Madison, Wisconsin, 1998.