This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier’s archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright
Efficient non-myopic value-of-information computation for influence diagrams

Wenhui Liao\textsuperscript{a,}\textsuperscript{*}, Qiang Ji\textsuperscript{b}

\textsuperscript{a}Research & Development, Thomson-Reuters Corporation, 610 Opperman Drive, Eagan, MN 55123, USA
\textsuperscript{b}Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA

Abstract

In an influence diagram (ID), value-of-information (VOI) is defined as the difference between the maximum expected utilities with and without knowing the outcome of an uncertainty variable prior to making a decision. It is widely used as a sensitivity analysis technique to rate the usefulness of various information sources, and to decide whether pieces of evidence are worth acquisition before actually using them. However, due to the exponential time complexity of exactly computing VOI of multiple information sources, decision analysts and expert-system designers focus on the myopic VOI, which assumes observing only one information source, even though several information sources are available. In this paper, we present an approximate algorithm to compute non-myopic VOI efficiently by utilizing the central-limit theorem. The proposed method overcomes several limitations in the existing work. In addition, a partitioning procedure based on the d-separation concept is proposed to further improve the computational complexity of the proposed algorithm. Both the experiments with synthetic data and the experiments with real data from a real-world application demonstrate that the proposed algorithm can approximate the true non-myopic VOI well even with a small number of observations. The accuracy and efficiency of the algorithm makes it feasible in various applications where efficiently evaluating a large amount of information sources is necessary.

1. Introduction

In a wide range of decision-making problems, a common scenario is that a decision maker must decide whether some information is worth collecting, and what information should be acquired first given several information sources available. Each set of information sources is usually evaluated by value-of-information (VOI). VOI is a quantitative measure of the value of knowing the outcome of the information source(s) prior to making a decision. In other words, it is quantified as the difference in value achievable with or without knowing the information sources in a decision-making problem.

Generally, VOI analysis is one of the most useful sensitivity analysis techniques for decision analysis [23,25]. VOI analysis evaluates the benefit of collecting additional information in a specific decision-making context [27]. General VOI analyses usually require three key elements: (1) A set of available actions and information collection strategies; (2) A model connecting the actions and the related uncertainty variables within the context of the decision; and (3) values for the decision outcomes. The methods of VOI analysis could be quite different when different models are used.

In this paper, we consider VOI analysis in decision problems modeled by influence diagrams. Influence diagrams were introduced by Howard and Matheson in 1981 [13] and have been widely used as a knowledge representation framework.
to facilitate decision making and probability inference under uncertainty. An ID uses a graphical representation to capture the three diverse sources of knowledge in decision making: conditional relationships about how events influence each other in the decision domain; informational relationships about what action sequences are feasible in any given set of circumstances; and functional relationships about how desirable the consequences are [21]. An ID can systematically model all the relevant random variables and decision variables in a compact graphical model.

In the past several years, a few methods have been proposed to compute VOI in IDs. Ezawa [8] introduces some basic concepts about VOI and evidence propagation in IDs. Dittmer and Jensen [7] present a method for calculating myopic VOI in IDs based on the strong junction tree framework [15]. Shachter [25] further improves this method by enhancing the strong junction tree as well as developing methods for reusing the original tree in order to perform multiple VOI calculations. Zhang et al. [28] present an algorithm to speed up the VOI computation by making use of the intermediate computation results, which are obtained when computing the optimal expected value of the original ID without the observations from the information sources. Instead of computing VOI directly, [22] describe a procedure to identify a partial order over variables in terms of their VOIs based on the topological relationships among variables in the ID. However, all these papers only focus on computing myopic VOI, which is based on two assumptions: (1) “No competition:” each information source is evaluated in isolation, as if it were the only source available for the entire decision; (2) “One-step horizon:” the decision maker will act immediately after consulting the source [21]. These assumptions result in a myopic policy: every time, the decision maker evaluates the VOI of each information source one by one, and chooses the one with the largest VOI. Then the observations are collected from the selected information sources, the probabilities are updated, and all the remaining information sources are to be reevaluated again, and a similar procedure repeats.

Obviously, the assumptions are not always reasonable in some decision circumstances. Usually, the decision maker will not act after acquiring only one information source. Also, although a single information source may have low VOI and is not worth acquisition compared to its cost, several information sources used together may have high VOI compared to their combined cost. In this case, by only evaluating myopic VOI, the conclusion will be not to collect such information, which is not optimal since its usage together with other information sources can lead to high value for the decision maker. Therefore, given these limitations in myopic VOI, it is necessary to compute non-myopic VOI.

Non-myopic VOI respects the fact that the decision maker may observe more than one piece of information before acting, thus requires the consideration of any possible ordered sequence of observations given a set of information sources. Unfortunately, the number of the sequences grows exponentially as the number of available information sources increases, and thus it is usually too cumbersome to compute non-myopic VOI for any practical use, and this is why the before mentioned work only focuses on myopic VOI. Given these facts, an approximate computation of non-myopic VOI is necessary to make it feasible in practical applications. To the best of our knowledge, [11] are the only ones who proposed a solution to this problem. In their approach, the central-limit theorem is applied to approximately compute non-myopic VOI in a special type of ID for the diagnosis problem, where only one decision node exists. Certain assumptions are required in their method: (1) all the random nodes and decision nodes in the ID are required to be binary; (2) the information sources are conditionally independent from each other given the hypothesis node, which is the node associated with the decision node and utility node.

Motivated by the method of Heckerman et al., we extend this method to more general cases1: (1) all the random nodes can have multiple states and the decision node can have multiple rules (alternatives); (2) the information sources can be dependent given the hypothesis node; and (3) the ID can have a more general structure. But same as Heckerman et al.’s method, we only discuss the VOI computation in terms of IDs that have only one decision node. This decision node shares only one utility node with another chance node. With the proposed algorithm, non-myopic VOI can be efficiently approximated. In order to validate the performance of the proposed algorithm, we not only perform the experiments based on the synthetic data for various types of IDs, but also provide a real-world application with real data.

Because of the efficiency and accuracy of the proposed method, we believe that it can be widely used to choose the optimal set of available information sources for a wide range of applications. No matter what selection strategies people use to choose an optimal set, such as greedy approaches, heuristic searching algorithms, or brute-force methods, the proposed method can be utilized to evaluate any information set efficiently in order to speed up the selection procedure.

The following sections are organized as follows. Section 2 presents a brief introduction to influence diagrams. The detail of the algorithm is described in Section 3. Section 4 discusses the experimental results based on synthetic data. And a real application is demonstrated in Section 5. Finally, Section 6 gives the conclusion and some suggestions for future work.

2. Influence diagrams

An influence diagram (ID) is a graphical representation of a decision-making problem under uncertainty. Its knowledge representation can be viewed through three hierarchical levels, namely, relational, functional, and numerical. At the relational level, an ID represents the relationships between different variables through an acyclic directed graph consisting of various node types and directed arcs. The functional level specifies the interrelationships between various node types and defines the corresponding conditional probability distributions. Finally, the numerical level specifies the actual numbers associated with the probability distributions and utility values [6].

---

1 A brief version of this extension can be found in [18].
Specifically, an ID includes three types of nodes: decision, chance (random), and value (utility) nodes. Decision nodes, usually drawn as rectangles, indicate the decisions to be made and their set of possible alternative values. Chance nodes, usually drawn as circles/ellipses, represent uncertain variables that are relevant to the decision problem. They are similar to the nodes in Bayesian networks [14], and are associated with conditional probability tables (CPTs). Value nodes, usually drawn as diamonds, are associated with utility functions to represent the utility of each possible combination of the outcomes of the parent node. The arcs connecting different types of nodes have different meanings. An arc between two chance nodes represents probabilistic dependence, while an arc from a decision node to a chance node represents functional dependence, which means the actions associated with the decision node affect the outcome of the chance node. An arc between two decision nodes implies time precedence, while an arc from a chance node to a decision node is informational, i.e., it shows which variable will be known to the decision maker before a decision is made [21]. An arc pointing to a utility node represents value influence, which indicates that the parents of the utility node are those that directly affect its utility. Fig. 1 illustrates these arcs and gives corresponding interpretations.

Most IDs assume a precedence ordering of the decision nodes. A regular ID assumes that there is a directed path containing all decision nodes; a no-forgetting ID assumes that each decision node and its parents are also parents of the successive decision nodes; and a stepwise decomposable ID assumes that the parents of each decision node divide the ID into two separate fractions. In this paper, we consider IDs that have only one decision node, i.e., ignoring all previous decisions. The goal of ID modeling is to choose an optimal policy that maximizes the overall expected utility. A policy is a sequence of decision rules where each rule corresponds to one decision node. Mathematically, if there is only one decision node in an ID and assuming additive decomposition of the utility functions, the expected utility under a decision rule \( d \) given any available evidence \( e \), denoted by \( EU(d|e) \), can be defined as follows:

\[
EU(d|e) = \sum_{i=1}^{n} \sum_{X_i} p(X_i|e,d)u_i(X_i,d),
\]

where \( u_i \) is the utility function over the domain \( X_i \cup \{D\} \). For example, \( X_i \) could be the parents of the utility node that \( u_i \) is associated with. To evaluate an ID is to find an optimal policy as well as to compute its optimal expected utility [24,26]. More detail about IDs can be found in [17,14].

Generally, the advantages of an ID can be summarized by its compact and intuitive formulation, its easy numerical assessment, and its effective graphical representation of dependence between variables for modeling decision making under uncertainty. These benefits make ID a widely used tool to model and solve complex decision problems in recent years.

3. Approximate VOI computation

3.1. Value of information

The VOI of a set of information sources is defined as the difference between the maximum expected utilities with and without the information sources [17]. VOI can be used to rate the usefulness of various information sources and to decide whether pieces of evidence are worth acquisition before actually using the information sources [21].

We discuss the VOI computation in terms of IDs that have only one decision node. This decision node shares only one utility node with another chance node, as shown in Fig. 2. And the decision node and the chance node are assumed to be independent. In the ID, the chance node \( \Theta \), named as hypothesis node, represents a mutually exclusive and exhaustive set of possible hypotheses \( h_1, h_2, \ldots, h_q \); the decision node \( D \) represents a set of possible alternatives \( d_1, d_2, \ldots, d_q \); the utility node \( U \) represents the utility of the decision maker, which depends on the outcome of \( \Theta \) and \( D \); and the chance nodes \( O_1, O_2, \ldots, O_n \) represent possible observations from all kinds of information sources about the true state of \( \Theta \). And each \( O \) may have multiple states. Let \( O = \{O_1, \ldots, O_n\} \), the VOI of \( O \), VOI(\( O \)), w.r.t. the decision node \( D \), can be defined as follows:

\[
VOI(O) = EU(O) - EU(\emptyset),
\]

\[
EU(O) = \sum_{o \in O} p(o) \max_{d \in D} \sum_{h \in \Theta} p(h|o)u(h,d),
\]

\[
EU(\emptyset) = \max_{d \in D} \sum_{h \in \Theta} p(h)u(h,d),
\]

Fig. 1. Interpretations of arcs in an ID, where circles represent chance (random) nodes, rectangles for decision nodes, and diamonds for value (utility) nodes.
where \( u() \) denotes the utility function associated with the utility node \( U \), \( EU(O) \) denotes the expected utility to the decision maker if \( O \) were observed, while \( EU(O) \) denotes the expected utility to the decision maker without observing \( O \). Here the cost of collecting information from the information sources is not included; thus, the VOI can also be called perfect VOI \([11]\). The net VOI is the difference between the perfect VOI and the cost of collecting information \([12]\). Since after calculating the perfect VOI, the computation of the net VOI is just a subtraction of cost, we focus on the perfect VOI in the subsequent sections.

As shown in Eq. \((2)\), to compute \( VOI(O) \), it is necessary to compute \( EU(O) \) and \( EU(O) \) respectively. Obviously, \( EU(O) \) is easier to compute, whereas directly computing \( EU(O) \) could be cumbersome. If the decision maker has the option to observe a subset of observations \( \{O_1, \ldots, O_k\} \) and each \( O_i \) has \( m \) possible values, then there are \( m^k \) possible instantiations of the observations in this set. Thus, to compute \( EU(O) \), there are \( m^k \) inferences to be performed. In other words, the time complexity of computing \( VOI(O) \) is exponential. It becomes infeasible to compute \( VOI(O) \) when \( n \) is not small.

The key to computing \( VOI(O) \) efficiently is to compute \( EU(O) \), which can be rewritten as follows:

\[
EU(O) = \sum_{\omega \in D} p(\omega) \max_{d_i \in D} \sum_{h_i \in h} p(h_i|\omega) u(\theta, d_i) = \sum_{\omega \in D} \max_{d_i \in D} \sum_{h_i \in h} p(h_i|\omega) u(\theta, d_i) \tag{5}
\]

It is noticed that each instantiation of \( O \) corresponds to a specific optimal action for the decision node \( D \). We define the decision function \( \delta : O \rightarrow D \), which maps an instantiation of \( O \) into a decision in \( D \). For example, \( \delta(o) = d_k \) indicates when the observation is \( o \), the corresponding optimal decision is \( d_k \). \( d_k = \arg \max_{d_i \in D} \sum_{h_i \in h} p(h_i|\omega) u(\theta, d_i) \). Therefore we can divide all the instantiations of \( O \) into several subsets, where the optimal action is the same for those instantiations in the same subset. Specifically, if \( D \) has \( q \) decision rules, \( \{d_1, \ldots, d_q\} \), all the instantiations of \( O \) can be divided into \( q \) subsets, \( o_{d_1}, o_{d_2}, \ldots, o_{d_q} \), where \( o_{d_k} = \{o \in O|\delta(o) = d_k\} \). Fig. 3 illustrates the relationships between each instantiation and the \( q \) subsets. Thus, from Eq. \((5)\), \( EU(O) \) can be further derived as follows:

\[
EU(O) = \sum_{h_i \in h} \sum_{k=1}^q \sum_{o \in o_k} p(o|\theta) u(\theta, d_k). \tag{6}
\]

In the next several sections, we show how to compute \( EU(O) \) efficiently.

### 3.2. Decision boundaries

In Eq. \((6)\), the difficult part is to compute \( \sum_{o \in o_k} p(o|\theta) \) because the size of the set \( o_{d_k} \) could be very large based on the previous analysis. In order to compute it efficiently, it is necessary to know how to divide all the instantiations of \( O \) into the \( q \) subsets. We first focus on the case that \( \Theta \) has only two states, \( \theta_1, \theta_2 \) and then extend it to the general case in Section 3.4.

Based on the definition, the expected utility of taking the action \( d_k \) is \( EU(d_k) = p(\theta_1) * u_{1k} + p(\theta_2) * u_{2k} \), where \( u_{1k} = u(\theta_1, d_k) \) and \( u_{2k} = u(\theta_2, d_k) \). We can sort the index of all the decision rules based on the utility functions, such that \( u_{1k} > u_{1j} \) and \( u_{2k} < u_{2j} \) for \( k < j \). Fig. 4 gives an example of the utility function \( u(\theta, D) \). As shown in the figure, as \( k \) increases, \( u_{1k} \) decreases and \( u_{2k} \) increases. If there is an action \( d_i \) that cannot be sorted according to this criterion, it is either

![Fig. 2. An ID example for non-myopic VOI computation. \( \theta \) is the hypothesis node, \( D \) is the decision node, and \( U \) is the utility node. \( O_i \) represents possible observations from an information source. There could be hidden nodes between \( \theta \) and \( O_i \).](image)

![Fig. 3. (a) Each \( o_i \) corresponds to an instantiation; (b) all the instantiations can be divided into \( q \) subsets, where each instantiation in the set \( o_{d_k} \) corresponds to the optimal decision \( d_k \).](image)
dominated by another action, or it dominates another action. (If \( u(d_i, \Theta) \) is always larger than \( u(d_j, \Theta) \), no matter what state of \( \Theta \) is, we say \( d_i \) dominates \( d_j \).) Then the dominated action can be removed from the set of possible actions, without changing the optimal policy.

**Proposition 1.** Let \( r_{jk} = \frac{u_{ki} - u_{kj}}{u_{ki} - u_{kj}} \), \( p_{ki} = \max_{k' \neq k} r_{k'j} \), and \( p_{kj} = \min_{k' \neq k} r_{k'j} \), then \( d_k \) is the optimal action if and only if \( p_{ki} < p(\theta_1) < p_{kj} \). In addition, \( p_{ki} = 0 \) and \( p_{kj} = 1 \). (Here \( k \) is the index of an action.)

**Proof.** see Appendix. \( \square \)

**Proposition 1** presents that if the probability of \( \Theta \) being \( \theta_1 \) is between \( p_{ki} \) and \( p_{kj} \), \( d_k \) is the optimal decision. From this, we can further derive **Proposition 2**.

**Proposition 2**

\[
\sum_{o \in o_{d_k}} p(o) = p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj}).
\]  

(7)

**Proof.** see Appendix. \( \square \)

The **Proof of Proposition 2** establishes Eq. (7) by showing that both sides of this equation express the probability that \( d_k \) is the optimal decision for \( \theta_1 \). Based on **Proposition 2**, we can get the following corollary.

**Corollary 1**

\[
\sum_{o \in o_{\theta_1}} p(o | \theta_1) = p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_1).
\]

(8)

\[
\sum_{o \in o_{\theta_2}} p(o | \theta_2) = p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_2).
\]

(9)

The equations in **Corollary 1** indicate the probability that the decision maker will take the optimal decision \( d_k \) after observing new evidence, given the situation that the state of \( \Theta \) is \( \theta_1 \) before collecting the evidence.

Based on **Corollary 1**, the problem of computing \( \sum_{o \in o_{\theta_1}} p(o | \theta_1) \) is transferred to the problem of computing \( p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_1) \), which is the topic of the next section. We will focus on \( p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_1) \) only because the procedure of computing \( p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_2) \) is similar.

### 3.3. Approximation with central-limit theorem

#### 3.3.1. A partitioning procedure

To compute \( p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_1) \), one way is to treat \( p(\theta_1 | o) \) as a random variable. If the probability density function of this variable is known, it will be easy to compute \( p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_1) \). However, it is hard to get such a probability density function directly. But we notice that \( p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_1) = p\left(\frac{\theta_1}{\theta_1 - \theta_2} \leq \frac{p(\theta_1 | o)}{p(\theta_1 | o) - p(\theta_2 | o)} \leq \frac{\theta_1}{\theta_1 - \theta_2} \right) \). Based on the transformation property between a random variable and its function [2], it is straightforward that \( p(p_{ki} \leq p(\theta_1 | o) \leq p_{kj} | \theta_1) \)

\[
= p\left(\frac{\theta_1}{\theta_1 - \theta_2} \leq \frac{p(\theta_1 | o)}{p(\theta_1 | o) - p(\theta_2 | o)} \leq \frac{\theta_1}{\theta_1 - \theta_2} \right).
\]

Let us take a closer look at \( \frac{p(\theta_1 | o)}{p(\theta_1 | o) - p(\theta_2 | o)} \) because it is critical in the approximate algorithm.
If all the $O_i$ nodes are conditionally independent from each other given $\Theta$, based on the chain rule:

$$
\frac{p(\theta_1 | \Theta)}{p(\theta_2 | \Theta)} = \frac{p(O_1 | \theta_1)}{p(O_2 | \theta_2)} \cdot \frac{p(O_1 | \theta_1)}{p(O_2 | \theta_2)} = \frac{p(O_1 | \theta_1)}{p(O_2 | \theta_2)} \cdot \frac{p(O_1 | \theta_1)}{p(O_2 | \theta_2)}
$$

(10)

Usually some $O_i$ may not be conditionally independent given $\Theta$. We will show that $p(O_i | \Theta)$ is approximately distributed as a log-normal random variable. However, in order to prove it, it is necessary to obtain a format similar to Eq. (10) even when $O_i$s are not conditionally independent. We thus propose a partitioning procedure to partition $O$ into several groups based on the principle of $d$-separation [21], where the nodes in one group are conditionally independent from the nodes in other groups. This procedure consists of three steps.

1. Decide whether two nodes, $O_i$, $O_j$, are conditionally independent given $\Theta$ by exploring the ID structure based on four rules: (i) if there is a directed path between $O_i$ and $O_j$ without passing $\Theta$, $O_i$ and $O_j$ are dependent; (ii) if both $O_i$ and $O_j$ are the ancestors of $\Theta$, $O_i$ and $O_j$ are dependent given $\Theta$; (iii) after removing the links to and from $\Theta$ from the original ID, if $O_i$ and $O_j$ have common ancestors, or $O_i$ is $O_j$’s ancestor, or vice versa, then $O_i$ and $O_j$ are dependent; and (iv) in all the other cases, $O_i$ and $O_j$ are conditionally independent given $\Theta$.
2. Build an undirected graph to model the relationships between the nodes. In such a graph, each vertex represents an $O_i$ node, and each edge between two vertices indicates that the two corresponding nodes are dependent according to the rules in Step 1.
3. Partition the graph into disjoint connected subgraphs. A depth-first search (DFS) algorithm [4] is used to partition the graph into several disjoint connected subgraphs (disjoint connected subgraphs) so that each component is disconnected from other components. The nodes in each connected component are conditionally independent from the nodes in any other connected components. Therefore, each connected component corresponds to one group.

For example, for the ID in Fig. 5a, with the partitioning procedure, the $O_i$ nodes can be divided into five groups, $\{O_1, O_2\}$, $\{O_3, O_4, O_5\}$, $\{O_6\}$, $\{O_7\}$, and $\{O_8, O_9\}$. Fig. 5b shows the graph built by the partitioning procedure.

3.3.2. Central-limit theorem

Generally, with the partition procedure presented in the previous subsection, $O$ can be automatically divided into several sets, named $O^1, O^2, \ldots, O^g$, where $g$ is the overall number of the groups. Thus, Eq. (10) can be modified as follows:

$$
\frac{p(\theta_1 | \Theta)}{p(\theta_2 | \Theta)} = \frac{p(O^1 | \theta_1)}{p(O^1 | \theta_2)} \cdot \frac{p(O^1 | \theta_1)}{p(O^1 | \theta_2)} \Rightarrow \ln \frac{p(\theta_1 | \Theta)}{p(\theta_2 | \Theta)} = \sum_{i=1}^{g} \ln \frac{p(O^i | \theta_1)}{p(O^i | \theta_2)} \Rightarrow \ln \phi = \sum_{i=1}^{g} w_i + c,
$$

where $\phi = \frac{p(\theta_1 | \Theta)}{p(\theta_2 | \Theta)}$, $w_i = \ln \frac{p(O^i | \theta_1)}{p(O^i | \theta_2)}$, $c = \ln \frac{p(\theta_1)}{p(\theta_2)}$.

(11)

In the above equation, $c$ can be regarded as a constant reflecting the state of $\Theta$ before any new observation is obtained and any new decision is taken. Here, we assume $p(\theta_2 | O)$, $p(O^i | \theta_2)$, and $p(\theta_2)$ are not equal to 0.

Let $W = \sum_{i=1}^{g} w_i$, be the sum of $w_i$. Following [11], we use the central-limit theorem to approximate $W$. The central-limit theorem [9] states that the sum of independent variables approaches a Gaussian distribution when the number of variables becomes large. Also, the expectation and variance of the sum is the sum of the expectation and variance of each individual random variable. Thus, regarding each $w_i$ as an independent variable, $W$ then follows a Gaussian distribution. Then, based on Eq. (11), $\phi$ will be a log-normal distribution. For a random variable $X$, if $\ln(X)$ has a Gaussian distribution, we say $X$ has a log-normal distribution. The probability density function is: $p(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$, denoted as $X \sim \text{LogN}(\mu, \sigma^2)$ [5], where $\mu$ and $\sigma$ are the mean and standard deviation of the variable’s logarithm [1]. In order to assess the parameters (mean and var-

![Fig. 5](image-url)
By the central-limit theorem, the expected value and the variance of \( W \) can be obtained by the following equations:

\[
\mu(W|\theta_1) = \sum_{i=1}^{r_1} p(\sigma_i^1|\theta_1) \ln \frac{p(\sigma_i^1|\theta_1)}{p(\sigma_i^2|\theta_2)},
\]

\[
\sigma^2(W|\theta_1) = \sum_{i=1}^{r_1} p(\sigma_i^1|\theta_1) \ln \frac{p(\sigma_i^1|\theta_1)}{p(\sigma_i^2|\theta_2)} - \mu^2(W|\theta_1).
\]

By the central-limit theorem, the expected value and the variance of \( W \) can be obtained by the following equations:

\[
\mu(W|\theta_1) = \sum_{i=1}^{s} \mu(W_i|\theta_1),
\]

\[
\sigma^2(W|\theta_1) = \sum_{i=1}^{s} \sigma^2(W_i|\theta_1).
\]

Therefore, based on Eq. (11), for \( W \sim N(\mu(W|\theta_1), \sigma^2(W|\theta_1)) \), we have \( \phi \sim \text{LogN}(\mu(W|\theta_1) + c, \sigma^2(W|\theta_1)) \), where \( \text{LogN} \) denotes the log-normal distribution. After getting the probability distribution function and the function parameters for \( \phi \) in Eq. (11), we are ready to assess the non-myopic VOI.

Before we go to the next section, we first analyze the computational steps involved in computing the parameters for the log-normal distribution, which is the most time-consuming part in the algorithm. Based on Eqs. (12) and (14), the overall number of the computational steps is \( 4 \sum_{i=1}^{d} r_i + 2g \). We will show that this number is much smaller than the overall number of the computational steps in the exact computational method during the algorithm analysis in Section 3.5.

### 3.3.3. Approximate non-myopic value-of-information

Based on Proposition 1 in Section 3.2, we know that \( d_i \) is the optimal action with the probability \( p(p_i|o) \leq p(p_i|o) \leq p(p_o|o) \), which is equivalent to \( p(\phi_{ki}|\theta_1) \leq \phi \leq \phi_{ki}|\theta_1 \) as shown in Section 3.3.1. Let \( \phi_{ki} = \frac{\phi_{ki}}{\phi_{ki}} \) and \( \phi_{ki} = \phi_{ki} \), thus \( d_i \) is the optimal decision if and only if \( \phi_{ki} \leq \phi \leq \phi_{ki} \). Then, based on Corollary 1 in Section 3.2, the following equation stands:

\[
\sum_{\phi_{ki}} p(\phi_{ki}|\theta_1) = p(\phi_{ki}|\theta_1) \leq p(\phi_{ki}|\theta_1).
\]

Furthermore, from Section 3.3.2, we know that \( \phi \sim \text{LogN}(\mu(W|\theta_1) + c, \sigma^2(W|\theta_1)) \), thus,

\[
p(\phi_{ki} \leq \phi \leq \phi_{ki}|\theta_1) = \frac{1}{\sigma(W|\theta_1) \sqrt{2\pi}x} e^{-\frac{(\ln(x) - \mu(W|\theta_1))^2}{2\sigma^2(W|\theta_1)}},
\]

\[
p(\phi_{ki} \leq \phi \leq \phi_{ki}|\theta_2) \text{ can be computed in the same way by replacing } \theta_1 \text{ with } \theta_2 \text{ in the previous equations.}
\]

Therefore, VOI can be approximated by combining Eqs. (2), (6), (16), and (17). Fig. 6 shows the key equations of the algorithm when \( \Theta \) has only two states. In summary, to approximate VOI(\( \Theta \)) efficiently, the key is to compute \( EU(\Theta) \), which leads to an approximation of \( \sum_{\phi_{ki}} p(\phi|\theta_1) \) with the log-normal distribution by exploiting the central-limit theorem and the decision boundaries.

### 3.4. Generalization

In the previous algorithm, the node \( \Theta \) only allows two states, although the other random nodes and the decision node can be multiple states. However, in real-world applications, \( \Theta \) may have more than two states. In this section, we extend the algorithm to the case that \( \Theta \) can have several states too. Assume \( \Theta \) has \( h \) states, \( \theta_1, \ldots, \theta_h \), and still, \( d \) has \( q \) rules, \( d_1, \ldots, d_q \), similarly to Eq. (11), we have the following equations:

| Table 1 |
| --- | --- | --- |
| \( W_i \) | \( p(W_i|\theta_1) \) | \( p(W_i|\theta_2) \) |
| \( p[p_1^1|\theta_1] \) | \( p[p_1^2|\theta_2] \) | \( p[p_1^2|\theta_2] \) |
| \( p[p_1^1|\theta_2] \) | \( p[p_1^2|\theta_2] \) | \( p[p_1^2|\theta_2] \) |
| \( p[p_1^1|\theta_2] \) | \( p[p_1^2|\theta_2] \) | \( p[p_1^2|\theta_2] \) |

The probability distribution of Table 2

Accordingly, \( VOI(\mathcal{O}) = EU(\mathcal{O}) - EU(\mathcal{O}) \)

\[
EU(\mathcal{O}) = \sum_{\theta \in \Theta} p(\theta) \sum_{k=0}^{g} \sum_{o \in \mathcal{O}_k} p(o \mid \theta) \mu(\theta, d_k)
\]

\[
\sum_{o \in \mathcal{O}_k} p(o \mid \theta) = p(\phi^*_k \leq \phi < \phi^*_k \mid \theta)
\]

\[
p(\phi^*_k \leq \phi < \phi^*_k \mid \theta) = \frac{1}{\sigma(W \mid \theta) \sqrt{2\pi}} \int_{\phi^*_k}^{\phi^*_k} e^{-\frac{(x-\mu(W \mid \theta)-c)^2}{2\sigma^2(W \mid \theta)}} dx
\]

Fig. 6. The key equations to approximate VOI when \( \theta \) has only two states, \( D \) has multiple rules, and the other nodes have multiple states.

\[
p(\theta_i | O) = \frac{p(O^i \mid \theta_i)}{p(O^i \mid \theta_h)} \cdot \frac{p(O^h \mid \theta_h)}{p(O^h \mid \theta_h)} \quad i \neq h \Rightarrow \ln \frac{p(\theta_i | O)}{p(\theta_h | O)} = \sum_{k=1}^{g} \ln \frac{p(O^i \mid \theta_i)}{p(O^i \mid \theta_h)} + \ln \frac{p(\theta_i)}{p(\theta_h)} \Rightarrow \ln \phi_i
\]

\[
= \sum_{k=1}^{g} w_k^i + c_i, \quad \text{where} \quad \phi_i = \frac{p(\theta_i | O)}{p(\theta_h | O)}, \quad w_k^i = \ln \frac{p(O^i \mid \theta_i)}{p(O^i \mid \theta_h)}, \quad c_i = \ln \frac{p(\theta_i)}{p(\theta_h)}.
\]

(Later in the marching, Table 1 can be modified as follows (see Table 2).

Thus, we get the following equations:

\[
\mu(w_k^i | \theta_i) = \sum_{i=1}^{r_k} p(\phi_i^k | \theta_i) \ln \frac{p(O^i \mid \theta_i)}{p(O^i \mid \theta_h)}, \quad 1 \leq i < h, 1 \leq j \leq h, 1 \leq k \leq g.
\]

\[
\sigma^2(w_k^i | \theta_i) = \sum_{i=1}^{r_k} p(\phi_i^k | \theta_i) \ln^2 \frac{p(O^i \mid \theta_i)}{p(O^i \mid \theta_h)} - \mu^2(w_k^i | \theta_i).
\]

Similar to Eq. (14), the expected value and the variance of \( W_i \) can be obtained as we see here:

\[
\mu(W_i | \theta_i) = \sum_{k=1}^{g} \mu(w_k^i | \theta_i), \quad 1 \leq i < h, 1 \leq j \leq h.
\]

\[
\sigma^2(W_i | \theta_i) = \sum_{k=1}^{g} \sigma^2(w_k^i | \theta_i).
\]

Accordingly, \( \phi_i \) follows the log-normal distribution with \( S_k = \sigma(W_i | \theta_i) \) and \( M_k = \mu(W_i | \theta_i) + c_i \). We denote the probability density function of \( \phi_i \) given \( \theta_i \) as \( f_{\phi_i}(\phi_i) \). Eqs. (19) and (21) show that the overall number of the computational steps to assess the parameters for the log-normal distributions is \( 4h \sum_{k=1}^{g} r_k + 2(h - 1)g \) when \( h > 2 \).

Even though \( f_{\phi_i}(\phi_i) \) can be easily obtained, it is still necessary to get the decision boundaries for each optimal decision in order to efficiently compute \( \sum_{\theta \in \Theta} p(o | \theta) \). Therefore, a set of linear inequality functions need to be solved when \( \theta \) has more than two states. For example, if \( d_k^i \) is the optimal action, \( EU(d_k) \) must be larger than the expected utility of taking any other action. Based on this, a set of linear inequality functions can be obtained:

| \( w_k^i \) | \( p(W_i^k | \theta_i) \) | \dots | \( p(W_i^k | \theta_h) \) |
|-----|-----|-----|
| \( \ln(p(\phi_i^k | \theta_i)) \) | \( p(\phi_i^k | \theta_i) \) | \( \dots \) | \( p(\phi_i^k | \theta_h) \) |
| \( \ln(p(\phi_i^k | \theta_i)) \) | \( p(\phi_i^k | \theta_i) \) | \( \dots \) | \( p(\phi_i^k | \theta_h) \) |
| \( \ln(p(\phi_i^k | \theta_i)) \) | \( p(\phi_i^k | \theta_i) \) | \( \dots \) | \( p(\phi_i^k | \theta_h) \) |
We assume $u_{ij} - u_{ik} > 0$; otherwise, “$>$” is changed to “$\leq$” in the last inequality.

Let $A_k$ be the solution region of the above linear inequalities, then

$$\sum_{\phi \in A_k} p(\phi|\theta) = \int_{A_k} f_i(\phi_1) \cdots f_i(\phi_n) \, dA_k, \quad 1 \leq j \leq h, \ 1 \leq k \leq q.$$  \hspace{1cm} (24)

The right side of Eq. (24) is an integral over the solution region $A_k$ decided by the linear inequalities. We first demonstrate how to solve the integral when $\Theta$ has three states, and then introduce the method for the case that $\Theta$ has more than three states.

When $\Theta$ has three states, Eq. (23) can be simplified as follows:

$$p(\theta_1)u_{1k} + p(\theta_2)u_{2k} + p(\theta_3)u_{3k} \geq p(\theta_1)u_{1j} + p(\theta_2)u_{2j} + p(\theta_3)u_{3j} \Rightarrow x_{1kj} p(\theta_1)/p(\theta_j) + x_{2kj} p(\theta_2)/p(\theta_j) + x_{3kj} p(\theta_3)/p(\theta_j) \geq 1,$$

where $x_{1kj} = u_{jk} - u_{1j}$ and $x_{2kj} = u_{jk} - u_{2j}$.

In the above, it is assumed that $u_{ij} > u_{kk}$; if $u_{ij} < u_{kk}$, then “$>$” is changed to “$\leq$” in the last inequality.

And Eq. (24) can be simplified as follows:

$$\sum_{\phi \in A_k} p(\phi|\theta) = \int_{A_k} f_i(\phi_1) f_j(\phi_2) \, dA_k, \quad 1 \leq k \leq q, \ 1 \leq j \leq 3.$$  \hspace{1cm} (26)

$A_k$ is decided by $(q - 1)$ linear inequalities and each inequality has two variables $\phi_1$ and $\phi_2$ as defined in Eq. (25). We use the following steps to solve this integral when $A_k$ is a finite region.

1. Identify all the lines that define the inequalities and find all the intersection points between any two lines as well as the intersection points between any line and the x (or y) axis.
2. Choose the intersection points that satisfy all the linear inequalities, and use them as vertices to form a polygon.
3. Divide the polygon into several simple regions: Specifically, for each vertex, we generate a line crossing this vertex and parallel to the y-axis. The lines then divide the polygon into several simple regions.
4. Evaluate the integral in each simple region and sum the values together.

An example of the solution region is shown in Fig. 7. In this example, if $x_{1kj} > x_{1k}(i \neq j)$, then $x_{2kj} > x_{2k}$ too. Therefore, the solution region can be decided by the intersection points of the lines that are defined by the linear inequalities and the axes. For example, in Fig. 7, $A_k$ is decided by $a$–$d$, which are selected from the intersection points $\{(1/\lambda_{1k}, 0), (0, 1/\lambda_{2kj}), j = 1, \ldots, q, j \neq k\}$. Based on [3], the time complexity of solving $m$ linear inequalities with $n$ variables (each inequality only has two variables) is $O(mn \log m + mn^2 \log^2 n)$. In this case, $n$ is 2 and $m$ is $q - 1$. 

![Fig. 7. A solution region of a group of linear inequalities.](image)
When $\Theta$ has more than three states, the integral needs to be performed in a high-dimension space (dimension is larger than 2). Therefore, we solve it with Quasi-Monte Carlo integration [10,16], which is a popular method to handle multiple integral. Quasi-Monte Carlo integration picks points based on sequences of quasirandom numbers over some simple domain $A_k$ which is a superset of $A_k$, checks whether each point is within $A_k$, and estimates the area (n-dimensional content) of $A_k$ as the area of $A_k$ multiplied by the fraction of points falling within $A_k$. Such a method is implemented by Mathematica [20], which can automatically handle a multiple integral with a region implicitly defined by multiple inequality functions.

Fig. 8 shows the key equations of the algorithm when $\Theta$ has multiple states. The main equations are similar to those in Fig. 6. However, since $\Theta$ has multiple states, it becomes more complex to obtain the parameters of the log-normal distribution and perform the integration.

3.5. Algorithm analysis

Now, we analyze the computational complexity of the proposed approximation algorithm compared to the exact computational method. For simplicity, assume that the number of the state of each $O_i$ node is $m$, and there are $n$ nodes in the set $O$. Assume we only count the time used for computing expected utilities. Then the computational complexity of the exact VOI computational method is approximately $h^m n^n$, where $h$ is the number of the state of the $\Theta$ node. With the approximation algorithm, the computational complexity is reduced to $h^k n^k$, where $h$ is the number of the state of the $\Theta$ node, and $k$ is the number of $O_i$ nodes in the maximum group among $O^1, \ldots, O^g$. In the best case, if all the $O_i$ nodes are conditionally independent given $\Theta$, the time complexity is about linear with respect to $m$. In the worst case, if all the $O_i$ nodes are dependent, the time complexity is approximately $m^n$. However, usually, in most real-world applications, $k$ is less than $n$, thus, the approximate algorithm is expected to be more efficient than the exact computational method, as will be shown in the experiments. For example, for the ID in Fig. 5, $n = 9$, $m = 4$, $h = 3$, and $q = 3$. Then, for the exact computation, the number of computations is around $\binom{3}{3} / C^4 9 = 786432$, while using the approximate algorithm, the number of computations is only around $3^4 + 4^4 = 192$.

However, in addition to the cost of computing expected utilities, the approximation algorithm also includes some extra costs: sorting the utility functions (Section 3.2), partitioning the $O$ set (Section 3.3.1), and deciding the decision boundaries (Section 3.2) when $\Theta$ has two states, or performing the integral when $\Theta$ has more than two states (Section 3.4). These costs are not included in the above analysis. In general, the extra time in these steps is much less than the time used for computing expected utilities. For example, the time complexity of sorting is $O(q \log(q))$, the time complexity of the partition procedure is $O(|V| + |E|)$ ($V$ is the set of vertex, and $E$ is the set of edges in an ID), and the time complexity in deciding the decision boundaries when $\Theta$ has two states is $O(q^2)$. When $\Theta$ has more than two states, deciding the decision boundaries needs additional time. Empirically, it does not affect the overall speed, as will be shown in the experiments. In addition, most steps in computing expected utilities involve performing inferences in an ID, which is usually NP-hard and thus consumes much more time than a step in the procedures of sorting, partitioning, and integrating.

![Fig. 8. The key equations to compute VOI when $\Theta$ has multiple states.](image-url)
4. Experiments

The experiments are designed to demonstrate the performance of the proposed algorithm compared to the exact VOI computation. We limit the ID test model with at most 5 layers\(^2\) and up to 11 information sources due to the exponential computational time behind the exact computation. Ten different ID models are constructed, where in one of the IDs the H node has four states.

Table 3 describes the structures of these IDs. The IDs are parameterized with 150 sets of different conditional probability tables and utility functions, a process which yields 1500 test cases. In each the one-third computational time behind the exact computation. Ten different ID models are constructed, where in one of the IDs the H node has four states.

When the set size of the O nodes increases, the error rate decreases. When the state number of H is the same, the error rates of the dependent cases are larger than the error rates of the conditional independent cases. This can be explained by the reason that the IDs in the dependent cases have fewer independent O subsets than the ID in the independent groups. Since the central-limit theorem is the basis of our algorithm, it works better when the number of w_i increases, which corresponds to the number of independent O subsets. Even when the size of O set is as small as 6, the average error is less than or around 0.1 for all the cases. We could run several larger IDs with much more O nodes, and the error curve would be progressively decreasing. Here, we intend to show the trend and the capability of this algorithm.

Charts (c) and (d) show the average computational time with the exact computation and the approximation computation. When the set size of the O nodes is small, the computational time is similar. However, as the size becomes larger, the computational time of the exact computation increases exponentially, while the computational time of the approximation algorithm increases much slower. Thus, the larger the O set size is, the more time the approximation algorithm can save. Likewise, as the number of the state of each O node further increases, the computational saving would be more significant. As the number of states of H increase, the computational time also slightly increases.

5. An illustrative application

We use a real-world application in human computer interaction to demonstrate the advantages of the proposed algorithm. Fig. 10 shows an ID for user stress recognition and user assistance. The diagram consists of two portions. The upper portion, from the top to the “stress” node, depicts the elements that can alter human stress. These elements include the workload, the environmental context, specific character of the user such as his/her trait, and importance of the goal that

\[ k \text{ is the size of the biggest group after partitioning.} \]

\[ \text{Table 3} \]

<table>
<thead>
<tr>
<th>k</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of IDs</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \text{Table 4} \]

| Testing cases | 50 test cases, where O_i nodes are conditionally independent given H whose state is binary | 50 test cases, where O_i nodes are conditionally independent given H who has three states | 50 test cases, where O_i nodes are conditionally independent given H who has four states | 450 test cases, where O_i nodes are conditionally independent given H whose state is binary | 450 test cases, where O_i nodes are conditionally independent given H who has three states | 450 test cases, where O_i nodes are conditionally independent given H who has four states |

\[ \text{Fig. 9} \]

\[ \text{Err} = \frac{|\text{VOI_t} - \text{VOI}|}{\text{VOI}} \]

The 1500 test cases described previously are divided into six groups, named as \text{ID}_\text{indep}: 2\text{-state}, \text{ID}_\text{indep}: 3\text{-state}, \text{ID}_\text{indep}: 4\text{-state}, \text{ID}_\text{dep}: 2\text{-state}, \text{ID}_\text{dep}: 3\text{-state}, and \text{ID}_\text{dep}: 4\text{-state}. Table 4 describes the six groups.
he/she is pursuing. This portion is called predictive portion. On the other hand, the lower portion of the diagram, from the “stress” node to the leaf nodes, depicts the observable features that reveal stress. These features include the quantifiable measures on the user physical appearance, physiology, behaviors, and performance. This portion is called diagnostic portion.
The hybrid structure enables the ID to combine the predictive factors and observable evidence in user stress inference. For more detail please refer to [19].

To provide timely and appropriate assistance to relieve stress, two types of decision nodes are embedded in the model to achieve this goal. The first type is the assistance node associated with the stress node, which includes three types of assistance that have different degrees of impact and intrusiveness to a user. Another type of decision nodes is the sensing action node ($S_i$ node in Fig. 10). It decides whether to activate a sensor for collecting evidence or not. Through the ID, we decide the sensing actions and the assistance action sequentially. In order to first determine the sensing actions (which sensors should be turned on), VOI is computed for a set $S$ consisting of $S_i$. Using the notations defined before, we have $\text{VOI}(S) = \text{VOI}(E) - \sum_{i \in S} u_i(S_i)$, where $E$ is the set of observations corresponding to $S$ and $\text{VOI}(E) = \text{EU}(E) - \text{EU}(\emptyset)$.

Fig. 11 shows the experimental results for the stress model. We enumerate all the possible combinations of sensors and then compute the value-of-information for each combination. Chart (a) illustrates the average VOI errors for different sensor sets with the same size. And Chart (b) displays the Euclidean distance between the true and estimated probabilities $\sum_{i \in O} p(\theta_i)$ (Eq. (26)). Similarly to the simulation experiments, the error decreases as the size of $O$ set increases, and the computational time increases almost linearly in the approximation algorithm.

### 6. Conclusions and future work

As a concept commonly used in influence diagrams, VOI is widely used as a criterion to rate the usefulness of various information sources, and to decide whether pieces of evidence are worth acquiring before actually using the information sources. Due to the exponential time complexity of computing non-myopic VOI for multiple information sources, most researchers focus on the myopic VOI, which requires the assumptions (“No competition” and “One-step horizon”) that may not meet the requirements of real-world applications.

We thus proposed an algorithm to approximately compute non-myopic VOI efficiently by utilizing the central-limit theorem. Although it is motivated by the method of [11], it overcomes the limitations of their method, and works for more general cases, specifically, no binary-state assumption for all the nodes and no conditional-independence assumption for the

![Fig. 11. Results for the stress modeling: (a) average errors with the approximation algorithm; (b) Euclidean distance between the true and approximated $\sum_{i \in O} p(\theta_i)$; (c) computational time (log(t), unit is second); (d) true VOI vs. approximated VOI.](image-url)
information sources. Table 5 compares our method with the method in [11]. Due to the benefits of our method, it can be applied to a much broader field. The experiments demonstrate that the proposed algorithm can approximate the true non-myopic VOI well, even with a small number of observations. The efficiency of the algorithm makes it a feasible solution in various applications when efficiently evaluating a lot of information sources is necessary.

Nevertheless, the proposed algorithm focuses on the influence diagrams with one decision node under certain assumptions. For example, currently, we assume the hypothesis node $\theta$ and the decision node $d$ are independent. If $d$ and $\theta$ are dependent, but conditionally independent given the observation set $O$, Eqs. (5) and (6) will not be affected, so our algorithm can still apply. However, if $D$ and $\theta$ are dependent given $O$, it may be difficult to directly apply our algorithm. Another scenario is that when there are more than one hypothesis node and/or utility nodes. One possible solution is to group all these hypotheses nodes into one. We would like to study these issues in the future.

Appendix

**Proposition 1.** Let $r_{jk} = \frac{u_{j}-u_{k}}{u_{j}+u_{k}+u_{j}-u_{k}}$, $p_{\theta}^{d} = \max_{k<\theta<q} r_{jk}$, and $p_{\theta^{\prime}}^{o} = \min_{1<\theta<q} r_{jk}$. then $d_{k}$ is the optimal action if and only if $p_{\theta}^{d} \leq p(\theta_{1}) \leq p_{\theta}^{\prime}$.

**Proof of Proposition 1.** $\Rightarrow$ In this direction, we prove that if $d_{k}$ is the optimal action, $p(\theta_{1}) \geq \max_{k<\theta<q} r_{jk}$ and $p(\theta_{1}) \leq \min_{1<\theta<q} r_{jk}$.

If $d_{k}$ is the optimal action, $EU(d_{k})$ must be larger than or equal to the expected utility of any other action. Based on the definition, the expected utility of taking the action $d_{k}$ is $EU(d_{k}) = p(\theta_{1}) \cdot u_{1k} + p(\theta_{2}) \cdot u_{2k}$, where $u_{1k} = u(\theta_{1}, d_{k})$, and $u_{2k} = u(\theta_{2}, d_{k})$. Therefore, we get the equations as follows:

$$EU(d_{k}) \geq EU(d_{j}) \ \forall j, j \neq k,$$

$$p(\theta_{1}) \cdot u_{1k} + p(\theta_{2}) \cdot u_{2k} \geq p(\theta_{1}) \cdot u_{1j} + p(\theta_{2}) \cdot u_{2j},$$

$$p(\theta_{1}) \geq \frac{u_{j} - u_{k}}{u_{j} + u_{j} + u_{j} - u_{k}} = r_{jk} \ \text{if} \ j > k,$$

$$p(\theta_{1}) \leq \frac{u_{j} - u_{j}}{u_{j} + u_{j} + u_{j} - u_{j}} = r_{jk} \ \text{if} \ j < k.$$  

Thus, based on the above equations, $p(\theta_{1}) \geq \max_{k<\theta<q} r_{jk}$ and $p(\theta_{1}) \leq \min_{1<\theta<q} r_{jk}$.

$\Leftarrow$ In this direction, we prove that if $p(\theta_{1}) \geq \max_{k<\theta<q} r_{jk}$ and $p(\theta_{1}) \leq \min_{1<\theta<q} r_{jk}$, then $d_{k}$ is the optimal action. If $p(\theta_{1}) \geq \max_{k<\theta<q} r_{jk} \ \forall j, k < j \leq q$, we get

$$p(\theta_{1}) \geq r_{jk} = \frac{u_{j} - u_{j}}{u_{j} + u_{j} + u_{j} - u_{j}},$$

$$p(\theta_{1})(u_{1k} - u_{1j} + u_{2j} - u_{2k}) \geq u_{2j} - u_{2k},$$

$$p(\theta_{1}) \cdot u_{1k} + (1 - p(\theta_{1})) \cdot u_{2k} \geq p(\theta_{1}) \cdot u_{1j} + (1 - p(\theta_{1})) \cdot u_{2j},$$

$$EU(d_{k}) \geq EU(d_{j}).$$

Similarly, for $\forall j, 1 < j < k$, we can get $EU(d_{k}) \geq EU(d_{j})$. Therefore, $d_{k}$ has the maximal expected utility and thus is the optimal decision.

**Proposition 2.** $\sum_{o \in o_{k^{\prime}}} p(o) = p(p_{\theta}^{d} \leq p(\theta_{1}|o) \leq p_{\theta}^{\prime}).$

**Proof of Proposition 2.** Based on Proposition 1, $d_{k}$ is the optimal decision if and only if the value of $p(\theta_{1})$ is between $p_{\theta}^{d}$ and $p_{\theta}^{\prime}$. Therefore, given an instantiation $o$, the probability that $d_{k}$ is the optimal decision is equal to the probability that $p(\theta_{1}|o)$ is between $p_{\theta}^{d}$ and $p_{\theta}^{\prime}$, i.e., $p(p_{\theta}^{d} \leq p(\theta_{1}|o) \leq p_{\theta}^{\prime}).$

On the other hand, we know that $o_{k^{\prime}}$ is a subset of instantiations, each of which corresponds to the optimal action $d_{k}$. Therefore, as long as $o$ belongs to the set of $o_{k^{\prime}}$, $d_{k}$ will be the optimal decision. In other words, the probability of $d_{k}$ being the optimal decision is the sum of the probability of each $o \in o_{k^{\prime}}$, which is $\sum_{o \in o_{k^{\prime}}} p(o)$. Therefore, $\sum_{o \in o_{k^{\prime}}} p(o) = p(p_{\theta}^{d} \leq p(\theta_{1}|o) \leq p_{\theta}^{\prime}).$
References