Explaining Inferences in Bayesian Networks

Ghim-Eng Yap   YAPG0001@NTU.EDU.SG

Ah-Hwee Tan   ASAHTAN@NTU.EDU.SG

School of Computer Engineering, Nanyang Technological University

Nanyang Avenue, Singapore 639798

Hwee-Hwa Pang   HHPANG@SMU.EDU.SG

School of Information Systems, Singapore Management University

80 Stamford Rd, Singapore 178902

Abstract

While Bayesian network (BN) can achieve accurate predictions even with erroneous or incomplete evidence, explaining the inferences remains a challenge. Existing approaches fall short because they do not exploit variable interactions and cannot account for compensations during inferences. This paper proposes the Explaining BN Inferences (EBI) procedure for explaining how variables interact to reach conclusions. EBI explains the value of a target node in terms of the influential nodes in the target's Markov blanket under specific contexts, where the Markov nodes include the target's parents, children, and the children's other parents. Working back from the target node, EBI shows the derivation of each intermediate variable, and finally explains how missing and erroneous evidence values are compensated. We validated EBI on a variety of problem domains, including mushroom classification, water purification and web page recommendation. The experiments show that EBI generates high quality, concise and comprehensible explanations for BN inferences, in particular the underlying compensation mechanism that enables BN to outperform alternative prediction systems, such as decision tree.

1. Introduction

Probabilistic reasoning systems like the Bayesian network (BN) [1] are developed to assist us with complex decisions. Given a set of input values, referred to commonly as the *findings* or *evidence*, BN derives the posterior probabilities for a target of interest. This is known as an *inference* on the target, and the target value with the highest belief or probability is the *prediction*. BN represents causal dependencies as directed arcs, and derives belief values by multiplying conditional probabilities. The encoded nodal dependencies in BN enable it to predict accurately even when important values are unavailable [2].

To gain user acceptance, however, the inferences made by the BN should be understandable to users. As the domain grows more complex, the gap between user and system
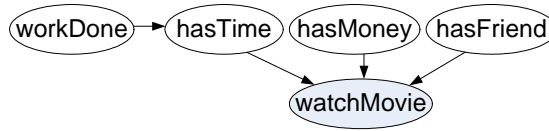
Figure 1: A Bayesian network (BN) for movie-watching.

knowledge widens, and the need for explanation increases. Indeed, a prominent user requirement of probabilistic systems is the ability to *explain* inferences [3]. Since BN infers based on probabilistic interactions among its variables, the explanations should reflect these interactions and the resulting compensation of missing and erroneous values.

Existing methods generally explain in terms of a set of input values that generates a target distribution similar to the inferred distribution. Figure 1 shows a BN for movie-watching. Suppose the boolean input values are given as {*workDone* = true, *hasMoney* = true, *hasFriend* = true}. Methods like INSITE [4] and BANTER [5] will check if individual input influences the target significantly, then search for the most influential path from each significant input to the target. A sample explanation for *watchMovie* may be:

```
Before presenting any evidence, the probability of watchMovie is p_a.
The following findings are considered important (in order of importance):
    • workDone results in a posterior probability of p_{b_1} for watchMovie.
    • hasMoney results in a posterior probability of p_{b_2} for watchMovie.
    • hasFriend results in a posterior probability of p_{b_3} for watchMovie.
Their influence flows along the following paths:
    • workDone influences hasTime, which influences watchMovie.
    • hasMoney influences watchMovie.
    • hasFriend influences watchMovie.
Presenting the evidence results in a posterior probability of p_b for watchMovie.
```

Such an explanation conceals the interactions among the parents into *watchMovie*, thus disregarding the fact that inferences for *watchMovie* should depend strongly upon its parents' interactions. There is also no mechanism for explaining missing and erroneous inputs.

Like the existing approaches, we believe that *probabilistic causal explanations* [6, 7] can elucidate the causal history behind inferences when formulated as *probabilistic rules* [8]:

$$if < conditions > then < conclusion > with\ probability < p > \qquad (1)$$

where p is given by $P(conclusion|conditions)$. In contrast to existing approaches that explain inferences in terms of individual evidence variables, we propose to explain the BN inferences on a target using a conjunction of the values in its *Markov blanket* [1], which includes the target's parents, children, and the children's other parents.

2

In the movie example, we would prefer to explain *watchMovie* in terms of the interactions among *hasTime*, *hasMoney* and *hasFriend*, rather than the inputs individually. In addition, suppose that *watchMovie*'s probabilities are independent of *hasFriend* when a user *hasTime* and *hasMoney*. This regularity is an instance of context-specific independence (CSI) [9], that we could exploit to generate a more succinct explanation such as the one below:

```
BN predicts watchMovie is true with probability p_b because
    hasTime is true with probability p_x, and hasMoney is true with probability p_y.
```

In this paper, we introduce the **E**xplaining **B**N **I**nferences (EBI) procedure. EBI considers just the target's Markov values during inference, because conditional independence implies that the Markov values fully explain the inference. To simplify the explanations, EBI exploits context-specific independencies reflected in the target's conditional probabilities.

The EBI procedure consists of three key steps described as follows. First, EBI restructures the BN (without changing its joint distribution) such that the target has its Markov nodes as parents. Next, EBI condenses the target's table of conditional probabilities into decision trees (DTs) [10]. Finally, the explanatory nodes under specific contexts are identified by traversing the DTs, and the corresponding EBI explanations are generated dynamically.

The EBI procedure is unique in that:

- Whereas most existing methods restrict their explanation to the evidence nodes, EBI explains the target node's value using the set of nodes in its Markov blanket that shield it from the rest of the network. This suits our human inclinations to reason in terms of a few specific heuristics [11], and to better accept causal stories as explanations [12].

- When observations on variables are presented as evidence for BN inference, some of the observed values may not be correct. While some explanation methods highlight erroneous values as conflicting findings, no existing method can explain how errors may be compensated during inference. Similarly, current methods do not explain how the missing values along the influence paths to the target node are derived. In contrast, EBI provides users with significant insights into these compensations, presented as layers of explanatory rules.

We evaluate EBI on a variety of problem domains, including mushrooms edibility classification [13], web page recommendation [14] and water purification [15]. The results show that the EBI procedure generates its explanations within seconds, and that the EBI explanations effectively account for the underlying BN compensation mechanism.

The rest of the paper is organized as follows. We introduce the relevant BN concepts and review the existing explanation methods in Section 2. Section 3 presents our EBI
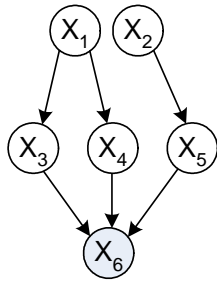
X_3	X_4	X_5	$P(X_6=0 \mid X_3, X_4, X_5)$	$P(X_6=1 \mid X_3, X_4, X_5)$
0	0	0	p1	1 - p1
0	0	1	p1	1 - p1
0	1	0	p1	1 - p1
0	1	1	p1	1 - p1
1	0	0	p2	1 - p2
1	0	1	p2	1 - p2
1	1	0	p3	1 - p3
1	1	1	p3	1 - p3

(a) Example BN.　　　　　　　　　　(b) CPT of node X_6.

Figure 2: Example Bayesian network (BN) and conditional probability table (CPT).

procedure, and Section 4 discusses how EBI explains BN compensations. Section 5 presents evaluations of EBI's efficacy. Section 6 concludes with a discussion of future work.

2. Preliminaries and Related Work

2.1 Bayesian Network (BN) and Variable Independencies

A Bayesian network (BN) is a directed acyclic graph, wherein each node represents a random variable and an edge indicates direct dependency between two variables. By its structure and conditional probabilities, a BN encodes a joint distribution that describes the domain's probabilistic semantics. We shall now introduce the key concepts of BN that are essential for understanding our approach to explaining BN inferences.

Definition (Conditional Independence): A variable X is conditionally independent of its non-descendants (denoted hereafter as $NDes(X)$) given the values of its parents [1]:

$$P(X|Parents(X), NDes(X)) = P(X|Parents(X)). \tag{2}$$

Consider the BN in Figure 2(a) involving binary variables X_1 to X_6, ordered in such a way that all non-descendants of X_i are labelled with an index smaller than i. The BN's joint probability, denoted as $P(X_1 = x_1, X_2 = x_2, \ldots, X_6 = x_6)$, or just $P(x_1, x_2, \ldots, x_6)$, can be factorized as:

$$P(x_1, x_2, \ldots, x_6) = P(x_1) \times P(x_2|x_1) \times \ldots \times P(x_6|x_1, \ldots, x_5) = \prod_i P(x_i|x_1, \ldots, x_{i-1}) \tag{3}$$

By exploiting conditional independence (Equation 2), the joint probability reduces to

$$P(x_1, x_2, \ldots, x_6) = \prod_i P(x_i|Parents(x_i)), \quad Parents(x_i) \subseteq \{x_1, \ldots, x_{i-1}\} \tag{4}$$

which allows a joint probability to be specified as a product of *conditional probability tables*.

4

Definition (Conditional Probability Table): For each node X, a conditional probability table (CPT) tabulates X's distribution for possible value assignments to its parents [1].

For example, the CPT in Figure 2(b) tabulates the conditional distribution of node X_6. In the CPT, $P(X_6 = 0 \mid X_3 = 0, X_4 = 0, X_5 = 0)$ is given by the conditional probability p1.

In many practical applications [9, 16–18], the CPTs can be further decomposed for faster inferences where the target is independent of certain parents given that certain other parents are assigned specific values. Such regularities are known as *context-specific independencies*.

Context-specific independence (CSI) is an independence relation that holds only when given specific value assignments to certain variables [9]. A formal definition due to Boutilier et al. [9] is given below.

Definition (Context-Specific Independence): Let $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{C}$ be pairwise disjoint sets of variables. Sets $\boldsymbol{X}$ and $\boldsymbol{Z}$ are *context-specifically independent*, or *contextually-independent*, given $\boldsymbol{Y}$ and the *context* $c \in val(\boldsymbol{C})$, if the conditional probability $P(\boldsymbol{X} \mid \boldsymbol{Y}, c, \boldsymbol{Z}) = P(\boldsymbol{X} \mid \boldsymbol{Y}, c)$ whenever $P(\boldsymbol{Y}, c, \boldsymbol{Z}) > 0$.

To illustrate with the CPT in Figure 2(b), $P(X_6 = 0 \mid X_3 = 0, X_4, X_5)$ is p1 regardless of the values of X_4 and X_5, i.e., X_6 is contextually independent of X_4 and X_5 given $(X_3 = 0)$.

Note that conditional independence implies that given the parent values of a target node with no descendant, the target is independent of the other nodes. However, conditional independence is insufficient to "block" or separate a given variable from its descendants, because it only asserts that

$$P(X|NDes(X), Des(X)) = P(X|Parents(X), Des(X)) \tag{5}$$

where $Des(X_i)$ is the set $\{X_{i+1}, X_{i+2}, \ldots\}$ of node X_i's descendants. Completely predicting a node's behavior requires a knowledge of the larger set of nodes in its *Markov blanket* [1].

Definition (Markov Blanket): The Markov blanket for a node X is the set of nodes $MB(X)$ comprising X's parents, children, and spouses (the child nodes' other parents) [1].

Definition (Markov Node, Markov Value): Variables in the Markov blanket of X constitute the *Markov nodes* of X; the Markov nodes' values are the *Markov values* of X.

For example, node X_5 in Figure 2(a) has Markov nodes X_2, X_6, X_3 and X_4 (their values constitute the Markov values of X_5). The *Markov property* asserts that direct dependencies must be shown by arcs, i.e., there cannot be "backdoors" in a BN [19]. Every other node is thus independent of or *d-separated* from a node X when conditioned on $MB(X)$. Formally,

$$P(X|MB(X), MB'(X)) = P(X|MB(X)), \quad MB'(X) \cap (MB(X) \cup X) = \emptyset \tag{6}$$

where $MB'(X)$ comprises the set of remaining variables in the Bayesian network that do not fall within the Markov blanket of node X, and $X \notin MB'(X)$.

During an inference, the belief or probability of each nodal value is updated according to the evidence. Bounded by a common joint distribution, the values assigned to the target's Markov nodes explain the inferred target value, because given just these Markov values, the most-probable target value is consistent with its expected value under the evidence.

2.2 Related Work on Explaining BN Inferences

Existing BN tools generally display inferences using bars, meters, or some other graphical expressions of differences in probabilities among variable values. Despite the easier interpretation, they are limited to examining single variable's distributions, and it remains hard to comprehend the overall inference process. Lacave and Diez [20] note that no explanation method has controlled the level of details to cater to a user's knowledge, or how inquisitive a user is. There is no dialogue that allows a user to gather explanations progressively.

INSITE [4] and BANTER [5] are well recognized BN explanation approaches. Both methods identify the individually influential inputs and the paths along which their influences flow. The difference is that BANTER *instantiates inputs individually*, whilst INSITE *removes input values individually* to compare their effects on the target's probabilities. Haddawy et al. [5] highlight that both BANTER and INSITE cannot detect influential inputs that interact among themselves, because their exhaustive approach requires an exponential number of network evaluations to do so.

Chajewska and Draper [21] propose to reduce the computational complexity by searching for explanations that are good but not necessarily the best. They define the postulates for a measure of explanation quality and provide examples of such functions. Their algorithm searches for explanations that satisfy a quality threshold. It remains a challenge to determine whether there exists an explanation with a quality exceeding a specific threshold, because this still requires an examination of all possible subsets of evidence variables.

Zhou et al. [22] propose the extraction of classification and correlation rules based on the target's probabilities. The rule set is a classification model that is compliant with the BN. However, because they do not have a mechanism to isolate the influential variables from among all the input values, their explanations can only be in terms of all the inputs. Moreover, because both the order of inputs and their relations under specific contexts are never considered, interactions among inputs are neglected by the resulting rules.

The above approaches assume that the user prefers explanations that connect the evidence variables to the target. They suffer when the assumption fails for two reasons:

- Firstly, the number of variables is often overwhelming. For example, there are usually thousands of variables in domains such as biomedicine, text and multimedia. As a
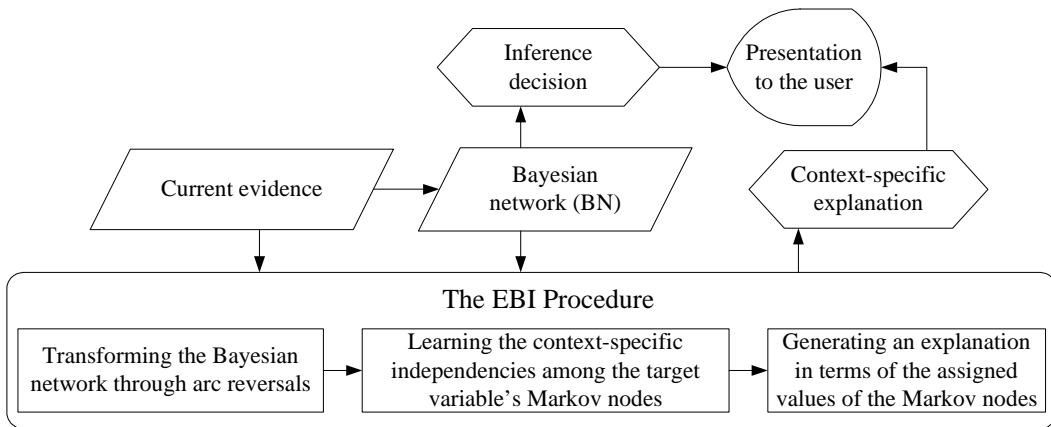
Figure 3: Overview of our proposed explanation procedure, EBI.

typical user understands only a handful of these, it is not always meaningful to explain inferences using variables that are part of the evidence.

- Secondly, the aforementioned approaches require an exponential number of analyses for all the possible subsets of evidence variables. They are limited to analyzing findings individually, and are computationally intractable for reflecting variable interactions.

Scenario-based reasoning [23] argues that inferences can instead be explained by those variables that are "relevant" to the target, even though they may not be part of the evidence. Firstly, the relevant variables are selected using the criterion of *d-separation* [1]. Each combination of values for these variables constitutes one possible scenario, or causal story. As there may be a large number of scenarios, only the most probable ones are presented.

Similar reviews on BN explanation approaches are found in [20] and [24]. To the best of our knowledge, no prior work has exploited the variable independencies in BN to isolate the target's Markov values for explaining its inferences. In addition, no prior work has exploited variable interactions to simplify the explanations, and none of the existing approaches can explain BN compensations during inferences.

3. Our Approach to Explaining BN Inferences

We introduce the Explaining BN Inferences (EBI) procedure, as summarized in Figure 3. Recall from Section 2.1 that the Markov blanket is the minimal set of nodes that completely predicts the behavior of the target node. EBI first restructures the local dependencies around the target node through arc reversals, in such a way that its Markov nodes become its parents while maintaining the same joint distribution. The resulting conditional probability table (CPT) of the target node describes its probability distribution over all combinations

A	B	C	D	P(E=e|A,B,C,D)	P(E=$\bar{e}$|A,B,C,D)
a	b	c	d	p1	1 - p1
a	b	c	$\bar{d}$	p1	1 - p1
a	b	$\bar{c}$	d	p1	1 - p1
a	b	$\bar{c}$	$\bar{d}$	p1	1 - p1
a	$\bar{b}$	c	d	p2	1 - p2
a	$\bar{b}$	c	$\bar{d}$	p2	1 - p2
a	$\bar{b}$	$\bar{c}$	d	p2	1 - p2
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	p2	1 - p2
$\bar{a}$	b	c	d	p3	1 - p3
$\bar{a}$	b	c	$\bar{d}$	p3	1 - p3
$\bar{a}$	b	$\bar{c}$	d	p4	1 - p4
$\bar{a}$	b	$\bar{c}$	$\bar{d}$	p5	1 - p5
$\bar{a}$	$\bar{b}$	c	d	p3	1 - p3
$\bar{a}$	$\bar{b}$	c	$\bar{d}$	p3	1 - p3
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	p6	1 - p6
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	p5	1 - p5

(a) BN before arc reversal. (b) BN after the arc between nodes E and C is reversed. (c) CPT of node E after the arc reversal.
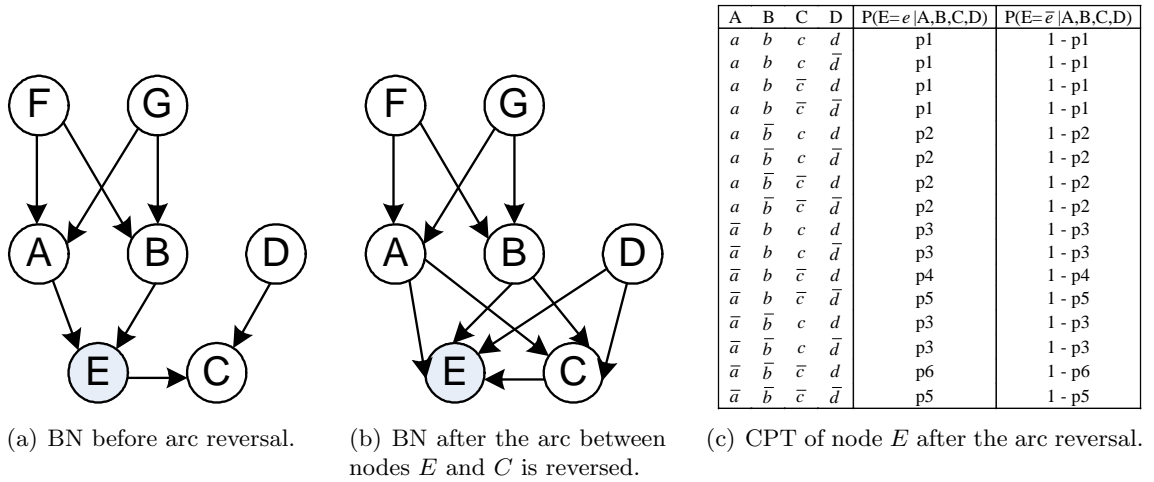
Figure 4: Running example for Section 3. The context-specific independencies for the target node E are captured as regularities in the conditional probabilities of Figure 4(c).

of Markov values. To reduce the complexity of the rules that EBI generates to explain the inference, EBI exploits context-specific independencies (CSI) and explains in terms of the significant Markov values. From the CPT, a decision tree (DT) [10] that represents the CSI among the Markov nodes is learned for each of the target values. During inference, EBI compares the assigned Markov values against the DT for the inferred target value, and explains using just the nodes on the path that matches the current context. In the remainder of this section, we elaborate on each of these component processes of EBI with Figure 4 as a running example.

3.1 Transforming the BN through Arc Reversals

Given a BN, we can reverse the arc from the target to each of its children, while readjusting their local structures and conditional probability tables such that the BN produces identical inferences as before. In the process, arcs are added from the other parents of each child node to the target node. Effectively, all the nodes in the Markov blanket of the target become its parents, allowing us to apply DT induction to the resulting target CPT.

Theorem (Arc Reversal): An arc (i,j) linking node i to node j can be replaced by the arc (j,i), provided that there is no other directed (i,j)-path in the network. After the reversal, both nodes inherit each other's conditional predecessors (the proof is given in [25]).

The requirement on the absence of alternate directed (i,j)-path is a necessary and sufficient condition to ensure that no cycle is created by the arc reversal transformation. For our purpose of reversing the arcs from the target t to each of its child nodes, this requirement is satisfied by sequencing the arc reversals; for a child c with an alternate directed (t,c)-path, the arc that links t to the child node on the alternative path is reversed before the arc (t,c).
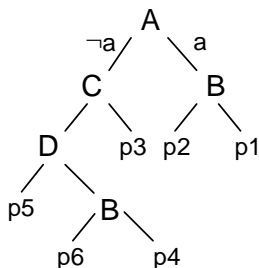
8

Figure 5: Decision tree (DT) induced from the CPT entries for value e in Figure 4(c).

We illustrate the arc reversal transformation [25, 26] with our running example. Figure 4(a) shows the original BN, and Figure 4(b) shows the BN after the arc from E to C is reversed. Before reversal, $Parents(E) - Parents(C) = \{A, B\}$ and $Parents(C) - Parents(E) - \{E\} = \{D\}$. The Markov blanket of E comprises parents $\{A, B\}$, child C, and the other parent of C, namely D. The reversal of $arc(E, C)$ involves setting $Parents(E) = Parents(C) = \{A, B\} \cup \{D\}$ so that E and C share a set of conditioning nodes, and computing their conditional probabilities as follows:

$$P(C|A, B, D) = \sum_E P(E|A, B)P(C|E, D) \tag{7}$$

$$P(E|C, A, B, D) = \frac{P(E|A, B)P(C|E, D)}{P(C|A, B, D)} \tag{8}$$

Equations 7 and 8 ensure that the joint distribution of the BN remains unchanged. Equation 7 computes $P(C|A, B, D)$ by summing out or marginalizing the node E from the joint distribution before arc reversal. Equation 8 computes $P(E|C, A, B, D)$ by equating the joint distributions before and after arc reversal, and dividing the joint distribution on each side by $P(C|A, B, D)$. The other child nodes of E and C are unaffected. Figure 4 shows that all the nodes in E's Markov blanket are transformed to become parents of E.

3.2 Learning Context-Specific Independencies among Markov Nodes

After arc reversal, the context-specific independencies (CSI) among the Markov nodes are learned from the resulting target conditional probability table (CPT). The CPT entries are grouped by the target values, and a decision tree (DT) [10] is learned from each group with the conditional probabilities as class labels. For the CPT in Figure 4(c), the Markov values and probabilities in the first two columns constitute the group of sixteen entries for value e of node E. Figure 5 shows the DT that is learned from this group.

DT induction is suitable as EBI's learning method because it is fast and straightforward. More importantly, each DT contains an *exhaustive* set of *mutually exclusive* paths. These

9

properties guarantee that within the learned DT, there is always a unique path that matches the current Markov value combination. Nevertheless, the quality of the decision tree induced from the CPT could affect the extent to which contextual independencies among the Markov nodes can be extracted. A discussion about representation of CPTs by decision trees is given in Boutilier et al [9]. Assuming that the CPTs are not revised incrementally by incoming examples, the entire process of reversing the arcs and learning the DTs can be completed off-line during the preprocessing. The DTs are then referred to during prediction to identify the minimal set of Markov nodes that explains the inference in the current context.

3.3 Generating Context-Specific Explanations based on Markov Nodes

When the BN predicts a value for the target, EBI dynamically generates an explanation for the inference in the current context. From the DT that corresponds to the predicted target value, EBI selects the path that matches the target's assigned Markov values, and forms the explanation with the nodes on this path. The most time-consuming task during explanation is the identification of the matching path. Zhang [27] points out that the matching operation is much more efficient with trees of paths than with sets of rules. Letting n be the number of Markov values that constitute the current context of the target variable, the worst-case time complexity of generating the EBI explanation is merely of order $O(n)$. As we shall discuss in the next section, EBI can be applied recursively to explain compensations during inference; the use of trees for representing CSI makes EBI efficient even in recursive application.

Consider the running example in Figure 4. Given the evidence, the BN infers a value for target E and similarly for each of its Markov nodes A, B, C and D. Without using the DTs, we can explain the inference on E by matching the sixteen entries (rules) in its CPT to the Markov values. For instance, if the Markov values are $\bar{a}$, b, c and $\bar{d}$, respectively, an explanation can be formulated from the matching CPT entry in row 10 of Figure 4(c). If p3=0.7, BN infers E as e with the following explanation:

- E is e with probability 0.7 because A is $\bar{a}$, B is b, C is c, and D is $\bar{d}$.

Instead of matching on the sixteen CPT entries, EBI traverses the DT shown in Figure 5 to find the path that contains the important Markov values within the current context. The matching process proceeds by descending from the tree root to a leaf node through following branches compatible with the Markov values. For the Markov values of $\bar{a}$, b, c and $\bar{d}$, the resulting EBI explanation for P(e) is independent of B and D in the context ($A=\bar{a}$, $C=c$):

- E is e with probability 0.7 because A is $\bar{a}$, and C is c.

Correspondingly, the EBI explanation for $E=\bar{e}$ is given as:

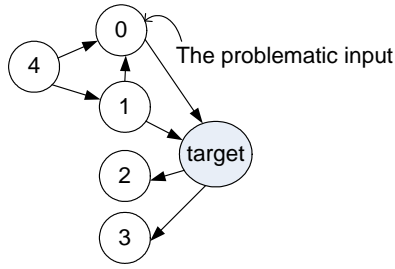- E is $\bar{e}$ with probability 0.3 because A is $\bar{a}$, and C is c.

Figure 6: Example for explaining BN compensations.

The explanations extracted using EBI are independent of the inference algorithm. Furthermore, because the inference result and the assigned Markov values account for the same evidence, EBI explanations are consistent with the inferences.

4. Explaining BN Compensations during Inferences

A significant advantage of EBI is that it can be extended to explain the mechanism by which a BN compensates for missing and erroneous inputs during inferences. EBI presents interested users with explanations on how the BN computes the distribution of one or more variables whose values are not supplied. In addition, EBI highlights those situations in which the network detects apparent errors in one or more variables in the target variable's Markov blanket, and explains how the network compensates for them during inferences. In this section, we discuss how the EBI procedure explains these hidden BN processes.

4.1 Explaining BN Compensation of Missing Inputs

An input is *missing* if its value is not given as an evidence. During BN inference, the node with the missing value is assigned a Markov value based on the interaction with its Markov nodes. Our treatment of missing inputs is to first explain the inference on the target variable as before using the assigned Markov values, and then provide an option for the users to retrieve further explanations for each missing Markov node.

Figure 6 shows a BN where the target is explained using the values of its Markov nodes *0*, *1*, *2* and *3*. If node *0* is missing during an inference, its assigned value is in turn explained based on its own Markov nodes. EBI highlights each missing Markov value as a *weakness* in the evidence. Procedure 1 summarizes EBI's explanation of missing value compensations.

While the basic EBI procedure emphasizes explanation in terms of the target's Markov nodes, *Recursive-EBI* allows the explanation process to propagate through the BN. This provides an interactive mechanism to explain how specific missing input values are compensated by the BN. Alternatively, one can use a graph tracing algorithm to explain *all* the missing inputs, while still benefiting from EBI's exploitation of variable independencies.

Procedure 1   Recursive-EBI (BN, *V*, *n*)

Input: BN, evidence *V*, and the node *n* to be explained.
Output: Explanation for the BN inference on *n*.

    if *n* is not the target node **then**
        Explain the BN's computed posterior distribution for *n* as a missing value compensation.
    else
        Explain *n* as the target node.
    Highlight any missing Markov node *m*.
    if the user requests an explanation on *m* **then**
        Recursive-EBI (BN, *V*, *m*).

4.2 Explaining BN Compensation of Erroneous Inputs

An input is potentially *erroneous* if its observed values carry a certain probability of being incorrect. For specific applications like those involving sensor measurements, it is possible to specify a sensible threshold to filter away low natural background variations in readings, such that we are able to identify those inputs with a higher inconsistency as having a greater potential to be erroneous. BN handles a potentially erroneous input by entering its value as a *likelihood finding* instead of a *specific finding*. It is the likelihood ratio of the input states, and not the specific likelihood of each state, that is taken as evidence [19]. The likelihood ratio represents the relative probability of observing a specific value of a variable given each of its possible states. This soft evidence allows the beliefs of that input to be affected by other nodes. Such exploitation of variable dependencies to correct inconsistent observations is related to the detection of inconsistencies in sensor data by Ibarguengoytia et al [28].

BN supports two types of error compensation. With reference to Figure 6, suppose that the value for node *0* is potentially in error and its value is entered as a *likelihood finding*. Its beliefs are left open to "correction" by the other nodes, under which its value may be revised to be more consistent with the other input findings. We term this as *Type I compensation.*

Definition (Type I Compensation): We say that Type I compensation of an erroneous input occurs, when its value is corrected to an assigned value after the BN inference.

Suppose that the value of node *0* is not corrected but the BN nevertheless produces the "right" inference. In other words, the effect of the erroneous node *0* has been compensated by nodes *1*, *2* and *3* in deriving the target value. We term this as *Type II compensation.*

Definition (Type II Compensation): We say that Type II compensation of an erroneous input occurs, when its effect on the value of a target node has been overcome by other inputs to the target after the BN inference.

EBI explains Type I and Type II compensations differently. To explain Type I compensation when an assigned Markov value differs from its finding, Recursive-EBI is enhanced to highlight the correction and explain the underlying compensation mechanism. We present

---

Procedure 2   Enhanced-EBI (BN, *V*, *n*)

Input: BN, evidence *V* and the node *n* to be explained.
Output: Explanation for the BN inference on *n*.

    if *n* is not the target node **then**
        if *n* is missing from the evidence **then**
            Explain the assigned value for *n* as a missing value compensation.
        else if *n* has its value corrected **then**
            Explain the correction on *n* as an erroneous value compensation.
    else
        Explain *n* as the target node.
    for each corrected Markov node *c* **do**
        Enhanced-EBI (BN, *V*, *c*).
    Highlight any missing Markov node *m*.
    if user requests an explanation on *m* **then**
        Enhanced-EBI (BN, *V*, *m*).

---

this as *Enhanced-EBI* in Procedure 2. The procedure presents users with further explanations when there are corrections in the target's Markov values due to Type I compensations.

For Type II compensations, it suffices for Enhanced-EBI to explain the target value as per Procedure 1. The reason is that EBI only considers the *assigned* Markov values after the beliefs in the BN have been updated based on the current evidence, regardless of whether they are likelihood findings, specific findings, or are inferred from other inputs.

4.3 Discussion

The simple strategies of EBI give rise to a number of apparent limitations, though most of the concerns can be addressed through straightforward extensions. They include:

- Explaining with most-probable values might be insufficient when there are more than one most-probable value, or when variable dimensions are numerous and the highest posterior probabilities are too low for the explanation to be convincing. In the former case, presenting more than one possible explanation could be a viable solution. In the latter case, the main concern is whether the presented posterior of the target variable is too low to be convincing. However, considering that this low posterior has provided the very basis for the inference, presenting the low posterior can in fact alert the user to the low degree of confidence in the BN conclusion, thus aiding the user's judgement.

- EBI explains both the inferred target value and the underlying compensation processes, using a local view of the target variable and its Markov variables. Nevertheless, it is possible for an individual user to prefer (even an incomplete) global view of the inference process instead of a local view when working in a given context. A truly

user-centric solution could entail a hybrid or combination of global and local views of the inference process.

- EBI is designed to effectively explain the BN inference on a single target node, but a user might require explanations for the value assignments in multiple targets instead. A straightforward solution would be to generate an independent EBI explanation for each of these target variables, by treating the rest of the variables as missing values in each round. However, EBI's recursive treatment of missing value explanation may then generate a large number of complex explanations. An alternative solution would be to take advantage of the recursion capability of the procedure, by presenting first the explanations for a small subset of the targets, then allowing the users to recursively obtain explanations for the remaining related target variables.

Keeping in view that these and other limitations may arise during user interactions, we proceed to empirically validate EBI's explanation efficacy in natural problem domains.

5. Experimental Validation

In Section 5.1, we present our experimental procedures and describe the problem domains. Sections 5.2 begins by comparing the prediction performance of BN and DT when inputs are missing. The purpose for the comparison is to provide motivation for improving explanation of BN inferences rather than predicting with DT, which are known to be reliable classifiers that also provide users with comprehensible rules. The remainder of Section 5.2 discusses examples from various domains to validate if EBI can effectively explain the missing inputs.

Following a similar organization, Section 5.3 compares BN against DT, and validates the EBI explanations when there are erroneous inputs. Section 5.4 summarizes the explanation efficacy of EBI in the various experiments.

5.1 Methods and Procedures

We use the CaMML program [29] to learn BN from data and the Netica-J API [30] to predict with BN. We present results on the mushrooms data set from UCI [13], the Syskill-Webert web page ratings data set [14], and the water network [15]. The data is preprocessed so that BN and DT are compared fairly on the same partitions of training and testing examples. The experimental results in this paper are obtained on a Pentium-4 3.0 GHz PC installed with Windows XP and 1.0 GB of physical memory.

The mushrooms data set consists of 8124 examples. For each mushroom sample, given a set of six observational input variables, the target variable *Edibility* predicts whether the mushroom is *poisonous* or *edible*. Both BN and DT learn on inputs *Odor*, *Spore-print-color*, *Stalk-surface-below-ring*, *Stalk-color-above-ring*, *Habitat* and *Cap-color* because these
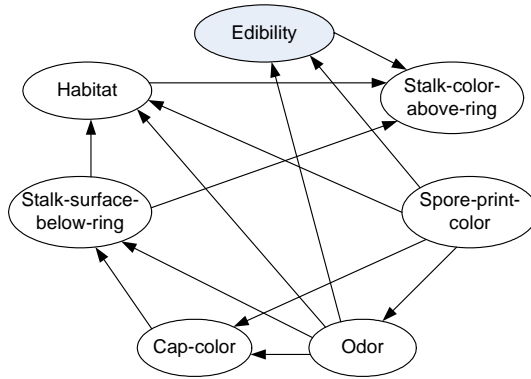
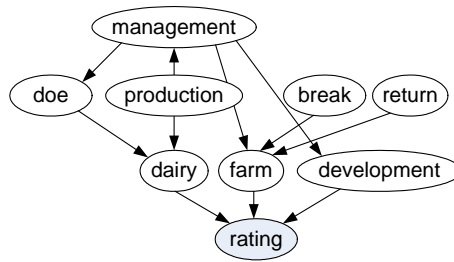Figure 7: A BN learned from the mushrooms data.

Figure 8: A BN learned from the web page data.

are known to have the highest classification power. As *Odor* is the most important input (with a classification accuracy of 98.52%), we compare BN and DT with *Odor* missing or in error. Figure 7 shows a BN learned from the mushrooms data.

The Syskill-Webert web page ratings data set comprises the HTML sources of real web pages and their corresponding ratings ("medium/cold", "hot") by a human subject. Web pages on the topic of goats provide seventy examples for our experiments. The attributes are English words extracted from the web pages, excluding stopwords from the SMART list [31]. We conduct multiple trials by training on randomly-chosen examples and testing on the rest. In each trial, we use the information gain criterion from the original study [32] to select the 32 most informative attributes. Figure 8 shows a BN that is learned from the web page data based on the minimal set learning procedure of Yap et al. [2].

The water network models the processes inside a water purification plant. As shown in Figure 9, eleven inputs from the 0th and 15th time slices are used to predict *CBODD_12_30*. A total of 1000 examples are generated using the Netica-J API. As DT uses only *CKNI_12_15* and *CBODD_12_15* in determining the value of target *CBODD_12_30*, we compare BN and DT when these inputs are missing or erroneous.

Prediction performance is measured by *prediction accuracy*, defined as the percentage of correctly classified test examples. (The mean absolute error (MAE) for a discrete class is
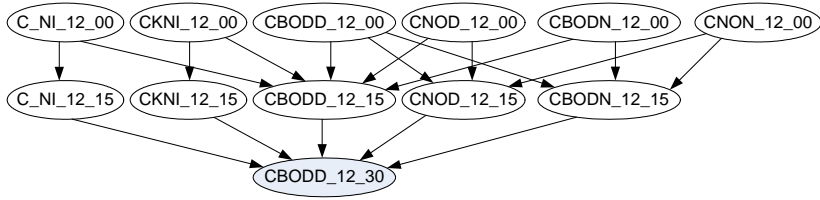
Figure 9: The water BN in our experiments.

Table 1: Prediction accuracy under missing inputs. *Complete*: all input values are present; *Missing*: important input values are missing.

	BN (%)		DT (%)	
	Complete	Missing	Complete	Missing
Mushrooms	99.45	95.76	99.90	54.39
Water	92.30	70.20	92.30	38.80
Web Page	75.00	71.15	72.69	66.92

one minus its prediction accuracy.) For the mushrooms data set, we perform five rounds of two-fold cross-validations to compare BN and DT. For the web page data set, we perform ten trials of sixty training and ten test examples. For the water data, we compare BN and DT over ten test sets of 100 examples each. For BN, the missing values are left out during prediction, whilst for DT we substitute missing values with the most frequent values in the corresponding training sets. For the error compensation experiments, training sets have no error while test sets have erroneous inputs with error rates ranging from 0% to 50%.

We need a measure of explanation quality to evaluate the EBI explanations. A suitable measure is given by Chajewska and Draper [21] to be the ratio of absolute differences:

$$f(o) = 1 - |\frac{P(o|\delta_B) - P(o|\delta_X)}{P(o|\delta_B) - P(o|\delta_A)}| \tag{9}$$

where δ_A is the state of the BN before inference, δ_B is the state after inference, δ_X is the state in the explanation, o is the predicted target value, and $P(o|\delta)$ is the predicted target value's probability given δ. Equation 9 evaluates to zero when $P(o|\delta_X)$ equals $P(o|\delta_A)$ (the explanation is worthless as it cannot explain the difference in $P(o|\delta_A)$ and $P(o|\delta_B)$); it peaks if $P(o|\delta_X)$ equals $P(o|\delta_B)$, is symmetrical about the peak, and falls linearly away from it.

5.2 Observations under Missing Inputs

The prediction accuracies of BN and DT for the scenarios with and without missing input in the three data sets are summarized in Table 1. The results agree with observations in Yap
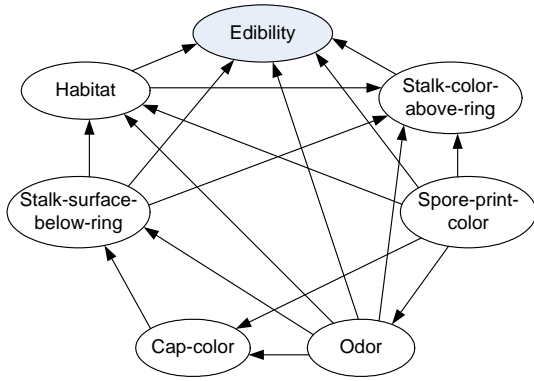
Figure 10: BN after reversal of the arc coming out of *Edibility* in Figure 7.
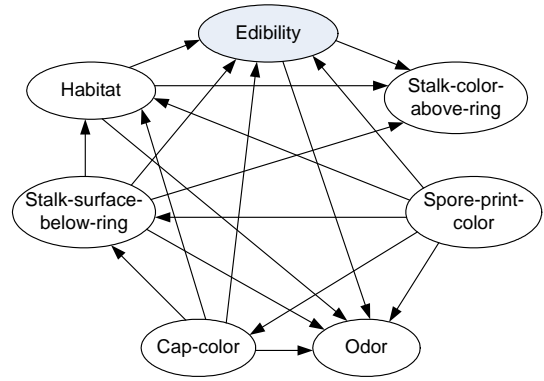
Figure 11: BN after reversal of arcs coming out of *Odor* in Figure 7.

Table 2: An explanation generated by EBI for mushrooms. *Odor*'s value is missing.

BN Inference with EBI Explanation	Quality of Explanation
BN predicts Mushroom is poisonous with probability 1.000 because Odor is foul with probability 1.000, Stalk-surface-below-ring is silky with probability 1.000, Habitat is grasses with probability 1.000, and Stalk-color-above-ring is brown with probability 1.000.	1.000
Missing values compensated: BN infers Odor as foul with probability 1.000 because Spore-print-color is chocolate with probability 1.000, Cap-color is yellow with probability 1.000, and Stalk-surface-below-ring is silky with probability 1.000.	

et al. [2] that BN and DT perform similarly when given complete data for prediction, but BN outperforms DT when important inputs are missing. The reason is that BN encodes its variable dependencies such that the missing values can be compensated by the other inputs.

For the mushrooms data set, the BNs learned in the various validation cycles have a structure similar to that in Figure 7. Knowing that missing values for *Odor* can be explained by the values of its Markov nodes, the structure does not tell us *how* these values explain the assigned value of *Odor*, and *how* this in turn explains the inference on *Edibility*. EBI brings out the missing insights.

Figures 10 and 11 show the networks after the BN in Figure 7 is restructured to explain the values of *Edibility* and *Odor*, respectively. Table 2 shows an EBI explanation when *Odor* is missing. For our illustration, the line under *Odor* indicates that compensation has taken place. Only the explanation for *Edibility* is given in the initial feedback. If the user is interested in the reason for the assigned *Odor* value, EBI presents an explanation for the

Table 3: An explanation generated by EBI for water. *CKNI_12_15*'s value is missing.

| BN Inference with EBI Explanation	Quality of Explanation

BN predicts CBODD_12_30 is 25mg/l with probability 0.549 because
    C_NI_12_15 is 6 with probability 1.000,
    CBODD_12_15 is 30mg/l with probability 1.000,
    <u>CKNI_12_15</u> is 30mg/l with probability 0.600, and
    CBODN_12_15 is 5mg/l with probability 1.000.
 	
Missing value compensated:
BN infers CKNI_12_15 is 30mg/l with probability 0.600 because
    CKNI_12_00 is 30mg/l with probability 1.000,
    C_NI_12_15 is 6 with probability 1.000, and
    CBODD_12_15 is 30mg/l with probability 1.000.
	1.000

Table 4: An explanation generated by EBI for web page. *development*'s value is missing.

| BN Inference with EBI Explanation	Quality of Explanation

BN predicts rating is cold/medium with probability 0.814 because
    dairy is absent with probability 1.000,
    <u>development</u> is absent with probability 0.936, and
    farm is absent with probability 1.000.
 	
Missing values compensated:
BN infers development is absent with probability 0.936 because
    management is absent with probability 1.000,
    dairy is absent with probability 1.000, and
    farm is absent with probability 1.000.
	1.000

compensation as shown. Although we can present the posteriors for all the Markov nodes of the missing input, we have removed the target node from the missing value explanation to avoid confusing the users. In the example, *Edibility* is independent of *Spore-print-color* and *Odor* is independent of *Habitat* given the rest of their respective Markov values. These variable interactions among the Markov nodes allow EBI to simplify its explanation.

Table 3 shows an EBI explanation for the water network's prediction on *CBODD_12_30* where *CKNI_12_15* is missing. With reference to Figure 9, *CBODD_12_30* and *CKNI_12_15* have five and six Markov nodes, respectively. EBI deduces that not all the Markov nodes are needed to explain the two values and presents its simplified explanations as shown. The missing status of *CKNI_12_15* is highlighted in the EBI explanation for *CBODD_12_30*, and only interested users are presented with the missing value explanation in the table.
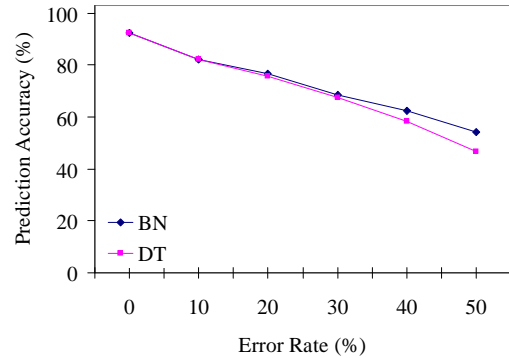
Table 4 shows an EBI explanation for the web page data. With reference to Figure 8, *rating* has *dairy*, *development* and *farm* as its Markov nodes. The explanation in Table 4 is
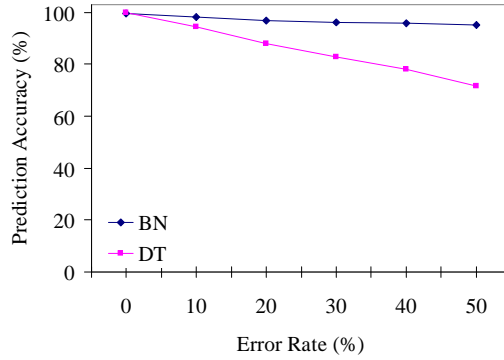
(a) For mushrooms.
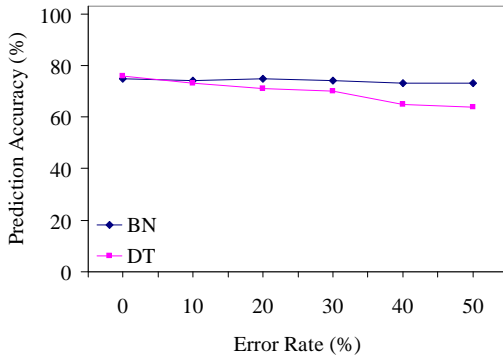
(b) For web page.

(c) For water.

Figure 12: Prediction accuracy with erroneous inputs in the test data.

for the case where *development* is missing from the evidence. Likewise, the missing Markov value is highlighted and the further explanation is presented only upon user request.

As shown, EBI explains missing inputs via a flexible and user-centered mechanism, thus helping users to better appreciate how the missing values are compensated during inferences.

5.3 Observations under Erroneous Inputs

Figures 12(a), 12(b) and 12(c) summarize the results for the three domains under the effect of erroneous inputs. BN outperforms the DT under identical error conditions because it captures and exploits the dependencies among its variables. Figures 12(a) and 12(b) show that BN maintains its accuracy despite the errors, whilst DT suffers as errors intensify. BN's robustness is most pronounced for mushrooms and less so for web pages, possibly due to a weaker dependency among words. Likewise, Figure 12(c) suggests that the dependencies among water plant processes are weak but BN outperforms DT when the error rate is high.

Table 5: An example from mushrooms data where a Type II compensation gives a different prediction when *Odor* is in error.

(a) No Error Among the Input Values.

Odor	*Spore-print-color*	*Edibility*
foul	chocolate	poisonous

(b) Wrong *Odor* - As Specific Finding.

Odor	*Spore-print-color*	*Edibility*
none	chocolate	edible

(c) Wrong *Odor* - As Likelihood Finding.

Odor	*Spore-print-color*	*Edibility*
foul	chocolate	poisonous

Table 6: An example from mushrooms data where a Type II compensation gives an identical prediction when *Odor* is in error.

(a) No Error Among the Input Values.

Odor	*Spore-print-color*	*Edibility*
none	black	edible

(b) Wrong *Odor* - As Specific Finding.

Odor	*Spore-print-color*	*Edibility*
fishy	black	edible

(c) Wrong *Odor* - As Likelihood Finding.

Odor	*Spore-print-color*	*Edibility*
none	black	edible

Table 5 presents an example from the mushrooms experiments where an error in *Odor* leads to a different inference on *Edibility*. The inference based on the correct *Odor* value of *foul* is that the mushroom is *poisonous* (Table 5(a)). However, when *Odor* is wrongly entered as *none*, BN predicts that the same mushroom is *edible* (Table 5(b)). This example corresponds to the first explanation in Table 7. In this case, when the erroneous *Odor* is entered as a specific finding, EBI explains the inferred *Edibility* value as follows:

```
BN predicts Mushroom is edible with probability 0.965 because
    Odor is none with probability 1.000,
    Spore-print-color is chocolate with probability 1.000,
    Stalk-surface-below-ring is silky with probability 1.000,
    Habitat is woods with probability 1.000, and
    Stalk-color-above-ring is brown with probability 1.000.
```

If we enter the erroneous value of *none* for *Odor* as a likelihood finding, *Odor* is corrected to *foul* during the inference and we get the same prediction as without error (Table 5(c)). EBI highlights this compensation as shown in the first explanation of Table 7, where the value of *Habitat* is found to be insignificant for explaining the correction in *Odor*.

Corresponding to the second explanation in Table 7, Table 6 shows another example from the mushrooms experiments. Just as when *Odor* is *none* (Table 6(a)), the BN predicts that the mushroom is *edible* when the wrong *Odor* value of *fishy* is entered as specific finding (Table 6(b)). Type II compensation, which involves inputs to the target interacting to yield

Table 7: Two explanations generated by EBI for mushrooms. *Odor*'s value is in error.

BN Inference with EBI Explanation	Quality of Explanation

BN predicts Mushroom is poisonous with probability 1.000 because
    <u>Odor is foul</u> with probability 1.000,
    Spore-print-color is chocolate with probability 1.000,
    Stalk-surface-below-ring is silky with probability 1.000,
    Habitat is woods with probability 1.000, and
    Stalk-color-above-ring is brown with probability 1.000.
 	
 Erroneous values compensated:
 BN corrects Odor from none to foul because given
    Spore-print-color is chocolate, Cap-color is yellow,
    and Stalk-surface-below-ring is silky,
       Odor is none with probability 0.000, and
       Odor is foul with probability 1.000.
	1.000

BN predicts Mushroom is edible with probability 0.999 because
    <u>Odor is none</u> with probability 0.999,
    Spore-print-color is black with probability 1.000,
    Stalk-surface-below-ring is fibrous with probability 1.000, and
    Habitat is grasses with probability 1.000.
 	
 Erroneous values compensated:
 BN corrects Odor from fishy to none because given
    Spore-print-color is black, Cap-color is white,
    and Stalk-surface-below-ring is fibrous,
       Odor is fishy with probability 0.000, and
       Odor is none with probability 0.999.
	1.000

a right inference, helps the BN in this case to maintain the same prediction even when *Odor* is in error. EBI explains this inference as follows:

```
BN predicts Mushroom is edible with probability 0.523 because

   Odor is fishy with probability 1.000,

   Spore-print-color is black with probability 1.000,

   Stalk-surface-below-ring is fibrous with probability 1.000,

   Habitat is grasses with probability 1.000, and

   Stalk-color-above-ring is white with probability 1.000.
```

When the wrong *Odor* value of *fishy* is entered as a likelihood finding, the BN corrects the *Odor* value to *none*, in addition to making the right inference that the mushroom is edible (Table 6(c)). EBI highlights this compensation as shown in the second explanation of Table 7. In the example, EBI simplifies its explanation for the inference on *Edibility* by exploiting context-specific independencies among the nodes in *Edibility*'s Markov blanket (the

Table 8: An explanation generated by EBI for water. *CBODD_12_15*'s value is in error.

BN Inference with EBI Explanation	Quality of Explanation
BN predicts CBODD_12_30 is 25mg/l with probability 0.878 because	1.000
C_NI_12_15 is 6 with probability 1.000,	
<u>CBODD_12_15 is 25mg/l with probability 0.984</u>,	
CKNI_12_15 is 20mg/l with probability 1.000, and	
CBODN_12_15 is 20mg/l with probability 1.000.	
Erroneous values compensated:	
BN corrects CBODD_12_15 from 30mg/l to 25mg/l because given	
CKNI_12_00 is 20mg/l, CBODD_12_00 is 25mg/l,	
and CKNI_12_15 is 20mg/l,	
CBODD_12_15 is 30mg/l with probability 0.016, and	
CBODD_12_15 is 25mg/l with probability 0.984.	

Table 9: An explanation generated by EBI for web page. *farm*'s value is in error.

BN Inference with EBI Explanation	Quality of Explanation
BN predicts rating is cold/medium with probability 0.984 because	1.000
dairy is absent with probability 1.000, and	
<u>farm is absent</u> with probability 0.997.	
Erroneous values compensated:	
BN corrects farm from present to absent because given	
pigs is absent, kids is absent, animal is absent,	
and dairy is absent,	
farm is present with probability 0.003, and	
farm is absent with probability 0.997.	

nodes shown as the parents of *Edibility* in Figure 10) to omit *Stalk-color-above-ring*. Likewise, EBI deduces from the discovered interactions among *Odor*'s Markov nodes (*Odor*'s parents in Figure 11) that *Habitat* is not necessary for explaining the correction in *Odor*.

Table 8 shows an EBI explanation for water. With reference to Figure 9, *CBODD_12_30* has five Markov nodes and the erroneous *CBODD_12_15* has ten. EBI exploits CSI to omit *CNOD_12_15* when explaining *CBODD_12_30*, and it uses just four of the ten Markov nodes for *CBODD_12_15* to explain its corrected value. Table 9 shows an EBI explanation for the web page problem. Likewise, EBI not only explains the inference on the target *rating*, but it also highlights and explains the correction in the value of *farm* during the inference.

5.4 Explanation Efficacy

We summarize the empirical statistics from the experiments. Table 10 presents the average time taken to generate an EBI explanation, and Tables 11 and 12 present the savings due to CSI. We quantify the savings by measuring the explanation complexity (defined as the

Table 10: Average time (in units of seconds) taken for generating each EBI explanation.

Mushrooms	1.397
Water	2.657
Web Page	0.001
Average	1.352

Table 11: Percentage of cases with a reduction in explanation complexity after exploiting CSI.

Mushrooms	62.98
Water	71.14
Web Page	10.00
Average	48.04

Table 12: Explanation complexity. *Before CSI*: Average number of antecedents before exploiting CSI; *After CSI*: Average number of antecedents after exploiting CSI.

	Before CSI	After CSI	% reduction
Mushrooms	6.393	5.615	12.17
Water	7.355	5.629	23.47
Web Page	3.471	3.306	4.757
Average	5.740	4.850	13.47

number of rule antecedents as per [33]) before and after exploiting CSI. Table 11 presents the percentage of all cases that experience a reduction in explanation complexity after exploiting CSI, and Table 12 presents the reduction in average explanation complexity across all cases.

The EBI explanations have an average quality (see formula (9)) of 100.0%. Table 10 shows that EBI generates its explanations within a matter of seconds, and Table 11 shows that as many as half of all test cases have benefitted from EBI's exploitation of CSI to simplify explanations. Table 12 shows that on average, EBI explanations enjoy a substantial reduction in complexity by exploiting CSI to identify the minimal set of important nodes.

The results demonstrate that EBI produces concise rules to explain BN inferences in real time. Most importantly, EBI is unique in its ability to explain missing and erroneous inputs. As illustrated by experiments on the various data sets, EBI effectively explains the intrinsic processes of BN compensations to the users.

6. Conclusion and Future Work

As Bayesian network (BN) exploits interactions among its variables during inferences, explanations for the inferences should reflect these interactions. To this end, our EBI procedure explains the inferred value of a node using its Markov values rather than the evidence variables. To make its explanations more concise, EBI exploits context-specific independencies to highlight the important Markov values in specific contexts. It then generates probabilistic explanations that show how variable interactions lead to the inferences. Experiments using

real-world data sets show that BN outperforms DT when inputs are missing or erroneous; EBI produces concise and comprehensible explanations for the underlying compensation mechanism, thus promoting user appreciation of the BN predictions.

The search for a natural and user-centric BN explanation process is a very hard problem, and the work reported here constitutes a step towards that end. There are several avenues for future work. The EBI procedure uses decision tree rules to represent variable interactions, but alternative forms of representation like M-of-N rules [34, 35] may provide a richer semantics. Besides context-specific independence, we can investigate other types of variable interactions that may be useful for explaining BN inferences. For instance, *causal independence* [36], defined as a situation where multiple causes contribute independently to a common effect, has known applicability to noisy OR-gate problems [1]. Yet another interesting future direction is to transform the induced decision tree data structures into their equivalent decision diagrams [37], which would reduce the space complexity for the representation of context-specific independencies. Finally, further studies are necessary to conceive and evaluate ways to enhance EBI for specific domains including biomedicine. While our research demonstrates the general feasibility of EBI, the collection of qualitative user evaluations is an important follow-on work.

References

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

[2] G.-E. Yap, A.-H. Tan, and H.-H. Pang. Discovering causal dependencies in mobile context-aware recommenders. In *Proceedings of MDM'06*, pages 4 (in CD–ROM Procs), Nara, Japan, May 2006. IEEE Computer Society.

[3] R. L. Teach and E. H. Shortliffe. An analysis of physician attitudes regarding computer-based clinical consultation systems. *Computers & Biomed. Research*, 14:542–558, 1981.

[4] H. J. Suermondt. *Explanation in Bayesian Belief networks.* PhD thesis, Medical Information Sciences, Stanford University, Palo Alto, California, March 1992.

[5] P. Haddawy, J. Jacobson, and C. Kahn(Jr.). BANTER: A Bayesian network tutoring shell. *Artificial Intelligence in Medicine*, 10:177–200, 1997.

[6] P. Humphreys. *The Chances of Explanation: Causal Explanation in the Social, Medical, and Physical Sciences.* Princeton University Press, Princeton, New Jersey, 1989.

[7] M. Strevens. Scientific explanation. *Macmillan Encyclopedia of Phil., 2nd Ed.*, 2006.

[8] S. K. M. Wong and W. Ziarko. INFER - An adaptive decision support system based on the probabilistic approximate classification. In *Proceedings of the 6th International Workshop on Expert Systems and Their Applications*, pages 713–725, Avignon, 1986.

[9] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Proceedings of UAI'96*, pages 115–123, Reed College, Portland, Oregon, USA, August 1996. Morgan Kaufmann.

[10] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, Ca, 1993.

[11] A. Tversky and D. Kahneman. Judgement under uncertainty: Heuristics and biases. *Readings in Uncertain Reasoning*, pages 32–39, 1974.

[12] N. Pennington and R. Hastie. Explanation-based decision making: Effects of memory structure on judgement. *JEP: Learning, Memory, and Cognition*, 14(3):521–533, 1988.

[13] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases. Irvine: University of California, Department of Information and Computer Sciences, 1998. URL `http://www.ics.uci.edu/∼mlearn/MLRepository.html`.

[14] M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting web sites. In *Proceedings of AAAI'96, IAAI'96*, pages 54–61, Portland, Oregon, 1996.

[15] F. V. Jensen, U. Kjærulff, K. G. Olesen, and J. Pedersen. An expert system for control of waste water treatment - A pilot project. Technical report, Judex Datasystemer A/S, Aalborg, Denmark, 1989. In Danish.

[16] N. L. Zhang and D. Poole. On the role of context-specific independence in probabilistic inference. In *Proceedings of IJCAI'99*, pages 1288–1293, Stockholm, Sweden, 1999.

[17] C. J. Butz. Exploiting contextual independencies in web search and user profiling. In *Proceedings of WCCI'02*, pages 1051–1056, Honolulu, Hawaii, USA, 2002.

[18] D. Poole and N. L. Zhang. Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research*, 18:263–313, 2003.

[19] K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence.* CRC Press, 2003.

[20] C. Lacave and F. J. Diez. A review of explanation methods for Bayesian networks. *Knowledge Engineering Review*, 17:107–127, 2002.

[21] U. Chajewska and D. L. Draper. Explaining predictions in Bayesian networks and influence diagrams. In *AAAI Spring Symposium*, Stanford Univ., Palo Alto, Ca., 1998.

[22] Z. Zhou, H. Liu, S. Z. Li, and C. S. Chua. Rule mining with prior knowledge: A belief networks approach. *Intelligent Data Analysis*, 5(2):95–110, 2001.

[23] M. Druzdzel and M. Henrion. Using scenarios to explain probabilistic inference. In *Working Notes of AAAI'90 Workshop on Explanation*, pages 133–141, 1990.

[24] J. R. Koiter. *Visualizing Inference in Bayesian Networks*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2006.

[25] R. D. Shachter. Evaluating influence diagrams. *Operations Res.*, 34(6):871–882, 1986.

[26] S. M. Olmsted. *On Representing and Solving Decision Problems*. PhD thesis, Dept. of Engineering-Economic Systems, Stanford University, Palo Alto, California, 1983.

[27] N. L. Zhang. Inference in Bayesian networks: The role of context-specific independence. Technical Report HKUST-CS98-09, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, 1998.

[28] P. Ibarguengoytia, L. Sucar, and S. Vadera. A probabilistic model for sensor validation. In *Proceedings of UAI'96*, pages 332–339, Reed College, USA, 1996. Morgan Kaufmann.

[29] C. S. Wallace and K. B. Korb. Learning linear causal models by MML sampling. In *Causal Models and Intelligent Data Management*. Springer, 1999.

[30] Norsys Software Corporation. Netica-Java Application Programmer Interfaces (Netica-J API), 2005. URL `http://www.norsys.com/netica-j.html`.

[31] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1971.

[32] M. J. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27:313–331, 1997.

[33] S. M. Weiss and N. Indurkhya. Reduced complexity rule induction. In *Proceedings of IJCAI'91*, pages 678–684, Sydney, Australia, August 1991. Morgan Kaufmann.

[34] G. G. Towell and J. W. Shavlik. Interpretation of artificial neural networks: Mapping knowledge-based neural networks into rules. *Advances in NIPS*, 4:977–984, 1992.

[35] G. G. Towell and J. W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13:71–101, 1993.

[36] N. L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.

[37] B. M.E. Moret. Decision trees and diagrams. *Computing Surveys*, 14(4):593–623, 1982.