A Method of Learning Implication Networks from Empirical Data: Algorithm and Monte-Carlo Simulation-Based Validation

Jiming Liu, *Member*, *IEEE*, and Michel C. Desmarais

Abstract—This paper describes an algorithmic means for inducing implication networks from empirical data samples. The induced network enables efficient inferences about the values of network nodes if certain observations are made. This implication induction method is approximate in nature as probablistic network requirements are relaxed in the construction of dependence relationships based on statistical testing. In order to examine the effectiveness and validity of the induction method, several Monte-Carlo simulations were conducted, where theoretical Bayesian networks were used to generate empirical data samples—some of which were used to induce implication relations, whereas others were used to verify the results of evidential reasoning with the induced networks. The values in the implication networks were predicted by applying a modified version of the Dempster-Shafer belief updating scheme. The results of predictions were, furthermore, compared to the ones generated by Pearl's stochastic simulation method [21], a probabilistic reasoning method that operates directly on the theoretical Bayesian networks. The comparisons consistently show that the results of predictions based on the induced networks would be *comparable* to those generated by Pearl's method, when reasoning in a variety of uncertain knowledge domains—those that were simulated using the presumed theoretical probabilistic networks of different topologies. Moreover, our validation experiments also reveal that the comparable performance of the implication-network-based-reasoning method can be achieved with much less computational cost than Pearl's stochastic simulation method; specifically, in all our experiments, the ratio between the actual CPU time required by our method and that by Pearl's is approximately 1:100.

Index Terms—Belief-network induction, probabilistic reasoning, learning algorithms, evidential reasoning, implication networks, implication-network induction, knowledge engineering, Monte-Carlo simulation, empirical validation.

◆

1 INTRODUCTION

As pointed out by Pearl [20], it would be impractical and inappropriate to represent real-world probabilistic knowledge by entries of a giant joint-probability distribution table. A more reasonable approach would be to opt for a network of probabilistic relationships among small clusters of nodes. A Bayesian network decomposes the joint-probability distribution with conditionals, and is usually defined in terms of a set of nodes representing assertions or variables and a set of connecting arcs signifying the independence relationships. The criterion for detecting these independence relationships embedded in the underlying probabilistic model is based on graph separation [2]. In the Bayesian network, the prior probabilities of all the root nodes and the conditional probabilities of other nodes given all the combinations of their parent nodes are specified. Hence, with the Bayesian network, if some information on the state of certain nodes or variables is obtained, the conditional probability distribution of other unobserved nodes can be updated, which is uniquely defined by the network. Bayesian networks have been used to model situations where causality is to be captured and reasoned about.

A number of algorithms for belief revision in Bayesian networks have been proposed in the past. Some are exact while others are approximate. The time complexity of them is in general problem-dependent. Comprehensive overviews of the existing methods can be found in [2]. Among those methods, Pearl's local propagation scheme handles a specific class of Bayesian networks, namely singly connected networks, in which any pair of nodes has at most one connecting path [20]. The probability distribution of a variable A in the network is updated based on three parameters, they are:

1) the current strength of the causal support contributed by each incoming link to A (i.e., for top-down predictive inference),
2) the current strength of the diagnostic support contributed by each outgoing link from A (i.e., for bottom-up diagnostic inference), and
3) the fixed conditional probability matrix relating the variable A to its immediate parents.

This scheme is strictly adherent to probability theory, and propagates harmoniously toward a stable equilibrium.

In order to handle multiply connected Bayesian networks, stochastic simulation methods have been widely used [16], [21], which eliminate the requirement of transforming a multiply connected network into a directed acyclic graph (DAG) as often done with an exact approach.

- *J. Liu is with the Department of Computing Studies, Baptist University, 224 Waterloo Rd., Kowloon Tong, Hong Kong. E-mail: jiming@comp.hkbu.edu.hk.*
- *M.C. Desmarais is with Centre de Recherche Informatique de Montreal, 1801 McGill College, Montreal, P.Q., Canada H3A 2N4.*

Manuscript received 6 Dec. 1995.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 104034.

In the stochastic methods (e.g., Henrion's logic sampling method [16] and Pearl's stochastic simulation method [21]), the precision of the reasoning is dependent on the size of stochastic samples that a simulation generates. Chin and Cooper [3] have noted that the stochastic simulation algorithms, when applied to certain networks, could lead to much slower than expected convergence to the true posterior probabilities, and proposed several possible forms of graph modification.

All the above mentioned evidential reasoning methods rely heavily on the availability of Bayesian networks. In real-life applications, however, it would be difficult to obtain a *real* Bayesian network mainly due to the insufficiency of empirical data and/or the complexity involved in the network induction. Generally speaking, constructing a valid knowledge representation is a time-consuming task, and often subject to opinion biases or logical inconsistency if it is built purely based on human heuristics. To overcome the difficulties in knowledge acquisition, several investigations have been carried out in recent years to explore the effectiveness and validity of automated means such as algorithms to perform this task [12].

Geiger [3] presented an algorithm for learning Bayesian networks that have a specific topology called a conditional tree. This algorithm combines an entropy-based optimization criterion with similarity networks [15]. Olesen et al. [19] considered the problem of modifying networks in changing environments and developed a tool for creating adaptive systems based on the compactly represented contingency table of imaginary counts. In addition, Geiger et al. [14] provided an algorithm for recovering structures such as trees, singly connected DAGs, and directed bipartite graphs.

Pitas et al. [22] proposed a method of learning general rules from specific instances based on a minimal entropy criterion. Cooper and Herskovits [5] developed an algorithmic method of empirically inducing probabilistic networks, which utilizes a Bayesian framework to assess the probability of a network topology given a distribution of cases. A heuristic technique was provided to optimize the search for probable topologies.

Another related work is the development of a *prediction logic* based on a contingency table of probabilities, as proposed by Hildebrand et al. [17]. In their work, the emphasis was on the computation of precision and accuracy of propositions represented. An analogy was made between the contingency table-based prediction logic and the formal proposition logic.

In this paper, we describe a new algorithm for inducing implication networks from a relatively small number of empirical data samples. The induced implication network can readily be used to compute the values of unobserved network nodes once a certain observation is made or a query is asserted. In this respect, our algorithmic implication induction method presents an alternative means for computationally deriving domain-knowledge structures essential for reasoning under uncertainty. The validity of the induced implication networks will be examined by way of Monte-Carlo simulation experiments.

The major difference between the previously mentioned approaches and ours is that the mentioned approaches focus on topological induction accuracy, while we focus on the accuracy of inferences based on an induced network without regards to the topological uniqueness. Our approach to implication induction draws on the previous work in empirical construction of inference networks [7], [8], [9], [10].

1.1 The Methodology for Investigation

Our study investigates the validity of an implication-network induction method with Monte-Carlo simulations. It starts with a given theoretical Bayesian network from which data samples can be generated and used to induce a set of new probabilistically significant implication relationships between pairs of individual variables. The belief revision in the induced networks rests on the assumption that the induced evidential sources are independent from one another. That is, combining evidence from multiple evidential sources assumes marginal independence for confirming or disconfirming evidence.

In this work, several experiments were carried out to compare the results of evidential reasoning in the induced networks with those generated by Pearl's stochastic simulation method—a multiply connected network solution based on local belief propagation algorithm [21], using the theoretical probabilistic networks.

The rationale behind the present Monte-Carlo simulations can be stated as follows: Since it is difficult to obtain a real Bayesian network from a set of empirical data samples, our validation of the induced networks needs to rely on a simulated data set generated from the "real" net—a given theoretical Bayesian network. This would ensure a fair comparison if the results of evidential reasoning in the induced networks are to be contrasted with the results obtained by an existing Bayesian-network-based-reasoning method, i.e., Pearl's stochastic simulation method.

1.2 The Organization of the Paper

The organization of the paper is as follows: Section 2 focuses on the algorithmic details of constructing an implication-network based on empirical data samples. Section 3 describes how the induced implication network is used to compute the values of unobserved network nodes once a certain observation is made or a query is asserted. Section 4 presents several Monte-Carlo experiments that examine the validity of the induced network, and compares the results with those of the commonly used Pearl's stochastic simulation method. Section 5 concludes the paper by highlighting the key findings of the present study.

2 IMPLICATION NETWORK INDUCTION

In the present work, we refer the term *implication network* to a directed graph in which each node represents an individual variable or hypothesis, and each arc signifies the existence of a direct implication (e.g., influence) between two adjacent nodes. The value taken on by one variable is dependent on the values taken on by all variables that influence it. Each value indicates the degree of belief that an unobserved variable is `TRUE`. This value is updated every time new information is obtained (e.g., some evidence is observed). The strengths of the node interdependencies are quantified by weights associated with the arcs.

Formally, an implication network can be represented as an ordered quadruple:

$$Net = \langle \mathcal{N}, \mathcal{I}, p_{min}, \alpha_c \rangle \tag{1}$$

where $\mathcal{N}$ is a finite set of nodes and $\mathcal{I}$ is a finite set of arcs. p_{min} is the minimal conditional probability to be estimated in the arcs and α_c is the network-induction error allowed (to be explained later). Furthermore, each induced implication relation can be specified by the following quadruple:

$$I \in \mathcal{I}, I = \langle N_{ant}, N_{con}, W_I, \tilde{W}_I \rangle \tag{2}$$

where W_I and $\tilde{W}_I$ are weight functions that map the pairs of antecedent-consequent node states, i.e., N_{ant} and N_{con} (Note: They can be either `TRUE` or `FALSE`), and their negations to a real number between 0 and 1, respectively. That is,

$$W_I : N_{ant} \times N_{con} \rightarrow [0, 1] \tag{3}$$

$$\tilde{W}_I : \neg N_{con} \times \neg N_{ant} \rightarrow [0, 1] \tag{4}$$

2.1 The Implication Induction Algorithm

The basic idea behind the *empirical induction* of implication relationships is that in an ideal case, if there is an implication relation `A` ⇒ `B`, then we would never expect to find the co-occurrences as in Fig. 1 that `A` is true but not `B`, from the empirical data samples. This translates into the following two conditions:

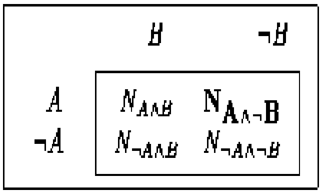
$$P(\mathtt{B} \mid \mathtt{A}) = 1 \tag{5}$$

$$P(\neg\mathtt{A} \mid \neg\mathtt{B}) = 1 \tag{6}$$

In reality, however, due to domain uncertainty or sampling errors, Conditions 5 and 6 may not be satisfied. Our implication induction algorithm takes into account the imprecise/inexact nature of implications and verifies the above conditions by computing the lower bound of a $(1 - \alpha_c)$ confidence interval around the measured conditional probabilities. If the verification succeeds, an implication relation between the two nodes is asserted. Two weights are associated with the implication,[1] expressing the degree of certainty in that relation. In the present implication-based representation, the weights are estimated based on conditional probabilities $P(\mathtt{B} \mid \mathtt{A})$ and $P(\neg\mathtt{A} \mid \neg\mathtt{B})$. Once an implication relation can be determined, another logical operator "⇔" is readily defined as follows:

$$(\mathtt{A} \Rightarrow \mathtt{B}) \Rightarrow ((\mathtt{B} \Rightarrow \mathtt{A}) \Rightarrow (\mathtt{B} \Leftrightarrow \mathtt{A})) \tag{7}$$

The elicitation of dependencies among the nodes requires considering the existence (or nonexistence) of direct relationships between pairs of random variables in a domain model. In theory, there exist six possible types of implications between any two nodes or events, the error cells corresponding to the uncertainty in these implication relations are summarized in Fig. 2.



	B	¬B
A	$N_{A \wedge B}$	$N_{A \wedge \neg B}$
¬A	$N_{\neg A \wedge B}$	$N_{\neg A \wedge \neg B}$

Fig. 1. A contingency table where each cell indicates the number of co-occurrences.

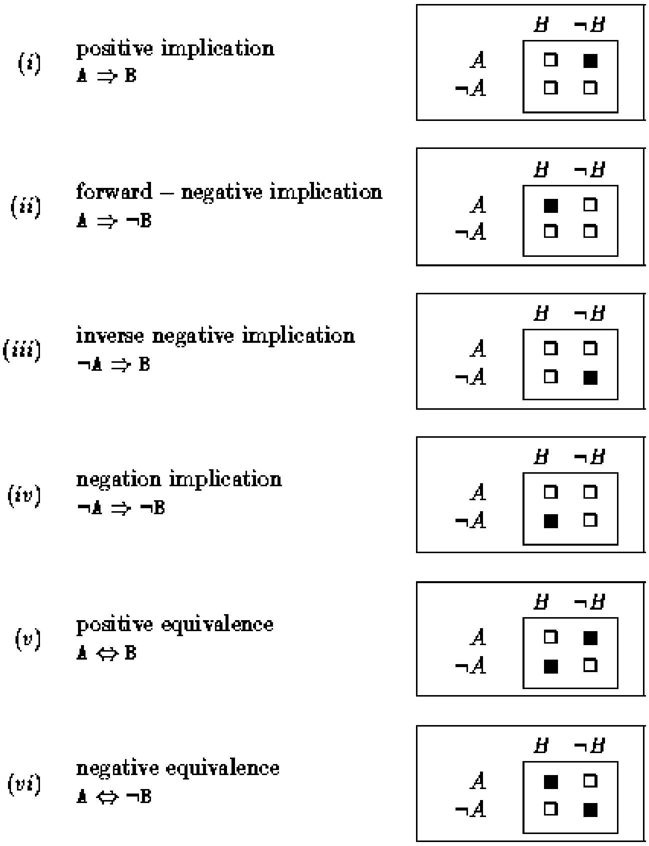


Fig. 2. The occurrences of errors in samples (denoted by ■), corresponding to the uncertainty in each of the six possible implications, are tested during the respective implication induction.

The implication induction algorithm can be stated as follows:

The Implication Induction Algorithm

```
Begin
set a significance level $\alpha_c$ and a minimal conditional probability $p_{min}$.
for node$_i$, $i \in [0, n_{max} - 1]$ and node$_j$, $j \in [i + 1, n_{max}]$
    for all empirical case samples
    compute a contingency table
```

$$T_{ij} = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}$$

where N_{11}, N_{12}, N_{21}, N_{22} are the numbers of occurrences with respect to the following combinations:

---

[1] With respect to the two directions of the inference, i.e., *modus ponens* vs. *modus tollens*.

```
            $N_{11}$ : nodei = TRUE ∧ nodej = TRUE
            $N_{12}$ : nodei = TRUE ∧ nodej = FALSE
            $N_{21}$ : nodei = FALSE ∧ nodej = TRUE
            $N_{22}$ : nodei = FALSE ∧ nodej = FALSE
        for each implication type $k$ out of the six possible cases (as in Fig. 2) test the following inequality:
                $P(x \le N_{error\_cell}) < \alpha_c$        (8)
            based on the lower tails of binomial distributions $Bin(N, p_{min})$ and $Bin(\tilde{N}, p_{min})$, where $N$ and $\tilde{N}$ denote the occurrences of antecedent satisfactions in the two inferences using a type $k$ implication relation, i.e., in modus ponens and modus tollens, respectively. $\alpha_c$ is the alpha error of the conditional probability test.
    if the test succeeds, then
        return a type $k$ implication relation.
End
```

In other words, we would like to test whether the probability of the errors as in the contingency table is less than a threshold. Suppose that the probability of committing an error in each single empirical data sample is p_e, and that all n samples are independent. If X is the frequency of the occurrence, then X satisfies a binomial distribution, whose probability function $p_X(k)$ and distribution function $F_X(k)$ are given below:

$$p_X(k) = \binom{n}{k} p_e^k \left(1 - p_e\right)^{n-k} \tag{9}$$

$$F_X(k) = p(X \le k) = \sum_{j=0}^{k} \binom{n}{j} p_e^i \left(1 - p_e\right)^{n-j} \tag{10}$$

A numerical example of implication induction is given in the Appendix.

2.2 An Example of an Implication Network

Fig. 3 gives a Bayesian network—a representation typically used by the existing probabilistic reasoning methods. In the present study, networks of this kind will serve as theoretical probabilistic relation networks for generating simulations about the states of the network nodes. Once the simulated data samples are generated, the implication induction algorithm can be applied. For this particular Bayesian network, we have generated 50 data samples, and used them to induce implication relationships. The criteria for accepting a relation are as follows: p_{min} = 0.8 and α_c = 0.2, which imply that we retain a relation involving conditional probabilities (corresponding to (5) and (6)) greater than 0.8 and that we tolerate a 20 percent error rate. Fig. 4 shows all the induced implication relationships under the above induction conditions; the new implication network contains 19 relations.

3 REASONING BASED ON THE INDUCED NETWORKS

Further to the construction of an implication network, inferences can be made by traversing the implication network and updating the belief values of the traversed nodes. As Charniak [2] points out, there might not exist an approximation belief updating scheme that works well for every situation, but it might be that in the end, we will simply have a library of algorithms. The current study on the validity of the induced network for evidential reasoning employs a modified version of the Dempster-Shafer method of evidential reasoning [6], [24] to propagate supports (whether confirming or disconfirming) throughout the implication network.

The Dempster-Shafer inferencing scheme may be regarded as a theoretical deviation from Bayesian theory. According to the Dempster-Shafer scheme, the set of possible outcomes of a node is called the *frame of discernment*, denoted by Θ. If the antecedents of a rule confirm a conclusion with degree m, the rule's effect on belief in the subsets of Θ can be represented by so-called probability masses. In our bivariate case of evidential reasoning, there are only two possible states for each node, q_i, that is, Θ = {`TRUE, FALSE`}.

The Dempster-Shafer scheme provides a means for combining beliefs from distinct sources, known as *Dempster's rule of combination*. This rule states that two assignments, corresponding to two independent sources of evidence, may be combined to yield a new one, that is,

$$m(X) = k \sum_{X_i \cap X_j = X} m_1\left(X_i\right)\, m_2\left(X_j\right) \tag{11}$$

where k is a normalization factor,

$$k = \frac{1}{1 - \sum_{X_i \cap X_j = \phi} m_1\left(X_i\right)\, m_2\left(X_j\right)} \tag{12}$$

Our belief revision algorithm works as follows: Starting from each of the observed nodes, q_i, it propagates the belief to its neighboring nodes based on the inference rules of *modus ponens* and *modus tollens*. In particular, if the observed value of q_i is `TRUE`, it performs forward chaining by following the positive implication $q_i \Rightarrow q_j$ and the forward-negative implication $q_i \Rightarrow \neg q_j$, and at the same time, backward chaining by following the forward-negative implication $q_j \Rightarrow \neg q_i$ and the negation implication $\neg q_j \Rightarrow \neg q_i$. Otherwise, vice versa. While doing so, it maintains a `queue` of next items from which the beliefs are to be propagated. The branching of a propagation stops whenever the path is terminated or the change in a belief value after updating is less than a threshold, θ (e.g., 0.1 percent). The belief revision algorithm can be stated as follows:

The Belief Revision Algorithm

{*Initially, all the observed nodes (i.e., the truth values of some nodes) are stored in a linked list,* `link`$_{observ}$. `insert` *and* `get_next_node` *are standard queuing functions.* `update_belief` *computes belief functions.* $\triangle$ *Bel*(•) *denotes the net change in beliefs after updating.*}

```
Begin
for each observed node, $q_i$ in link$_{observ}$, do
    insert($q_i$, queue);
    while queue is not empty, do
        node ← get_next_node(queue);
```

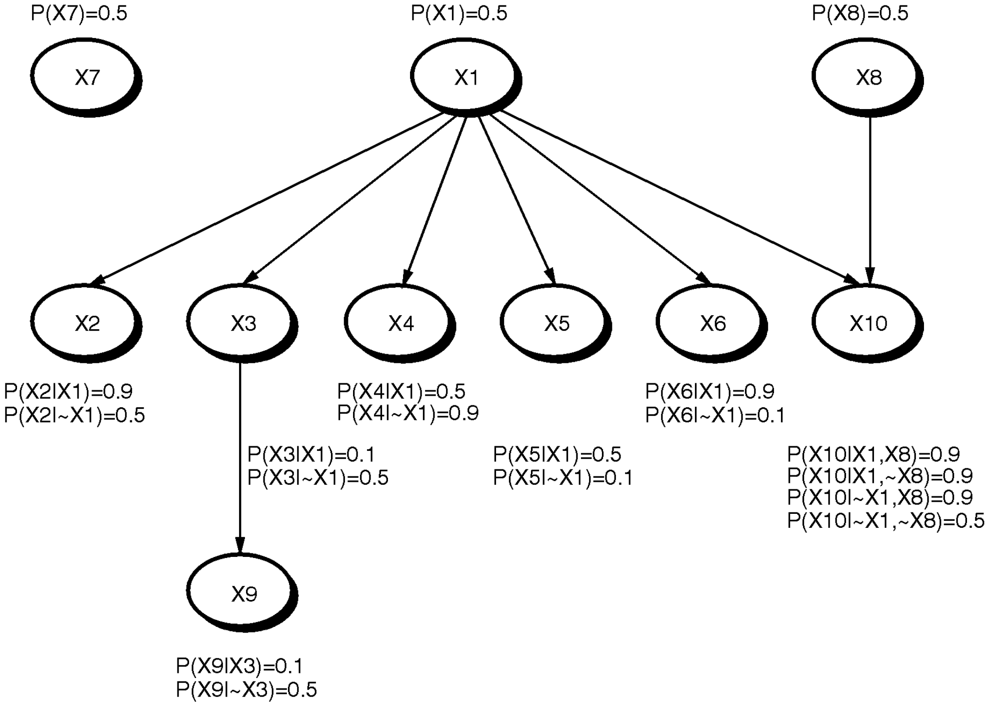
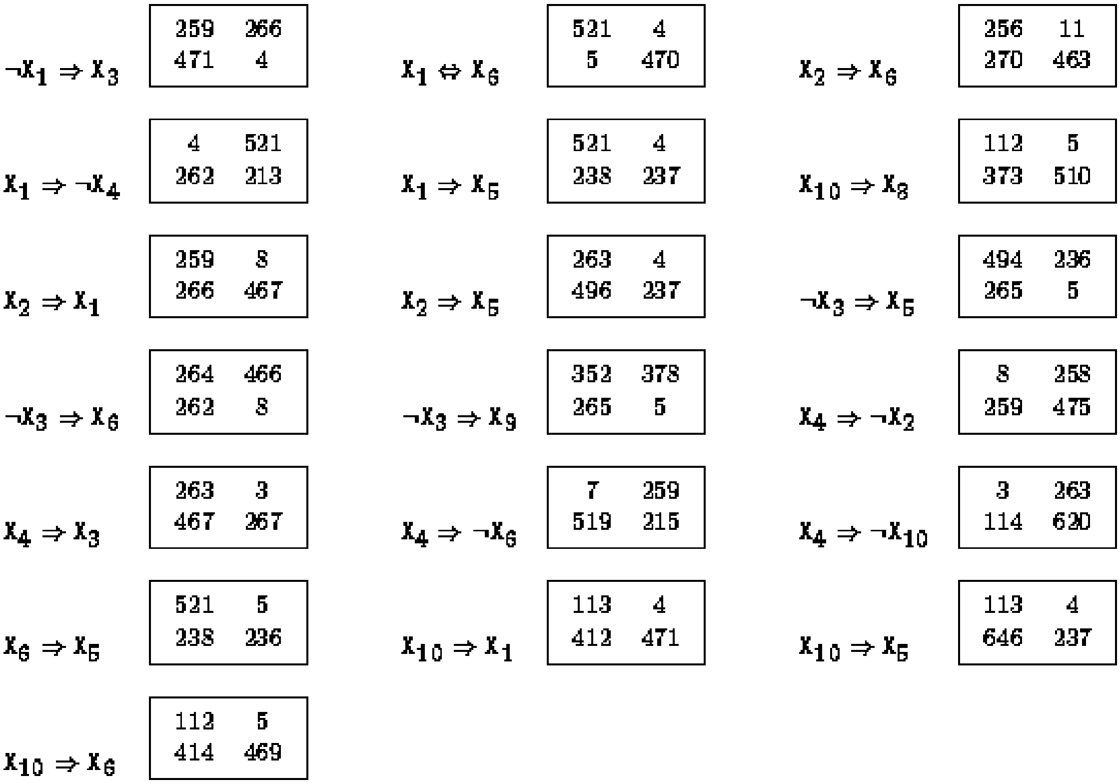


Fig. 3. A theoretical Bayesian network that is used to generate a collection of data samples upon which our implication induction algorithm operates.



$\neg X_1 \Rightarrow X_3$	259	266
	471	4
$X_1 \Leftrightarrow X_6$	521	4
	5	470
$X_2 \Rightarrow X_6$	256	11
	270	463
$X_1 \Rightarrow \neg X_4$	4	521
	262	213
$X_1 \Rightarrow X_5$	521	4
	238	237
$X_{10} \Rightarrow X_8$	112	5
	373	510
$X_2 \Rightarrow X_1$	259	8
	266	467
$X_2 \Rightarrow X_5$	263	4
	496	237
$\neg X_3 \Rightarrow X_5$	494	236
	265	5
$\neg X_3 \Rightarrow X_6$	264	466
	262	8
$\neg X_3 \Rightarrow X_9$	352	378
	265	5
$X_4 \Rightarrow \neg X_2$	8	258
	259	475
$X_4 \Rightarrow X_3$	263	3
	467	267
$X_4 \Rightarrow \neg X_6$	7	259
	519	215
$X_4 \Rightarrow \neg X_{10}$	3	263
	114	620
$X_6 \Rightarrow X_5$	521	5
	238	236
$X_{10} \Rightarrow X_1$	113	4
	412	471
$X_{10} \Rightarrow X_5$	113	4
	646	237
$X_{10} \Rightarrow X_6$	112	5
	414	469

Fig. 4. Given a set of 50 data samples generated from a Bayesian network, 19 implication relations can be derived using the induction algorithm. On the right-hand side of each implication relationship is the contingency table, □, tested during the induction of the respective implication.

```
if node = TRUE, then
    for each rule: node ⇒ $q_j$, node ⇒ ¬ $q_j$,
        $q_j$ ⇒ ¬ node, and ¬$q_j$ ⇒ ¬ node do
            $Bel(q_j)$ ← update_belief(node, $q_j$),
            if $\triangle Bel(q_j)$ > a threshold, $\theta$, then
                insert($q_j$, queue);
else
    for each rule: $q_k$ ⇒ node, ¬$q_k$ ⇒ node,
```

```
            $\neg$ node $\Rightarrow q_k$, and $\neg$ node $\Rightarrow \neg q_k$, do
                $Bel(q_k) \leftarrow$ update_belief
                    (node, $q_k$);
            if $\triangle Bel(q_k)$ > a threshold, $\theta$, then
                insert($q_k$, queue);
End
```

The basic probability assignment, *m*, augmented to each of the identified implication relations, utilizes the estimated conditional probability of the nonupdated node given an updated one.

It should be pointed out that an induced implication network may not always be a singly connected graph. In order to handle the problem of multiple transitivity in the network, our present implementation of the belief updating algorithm allows the traversal from one node to another to be performed only once, by randomly choosing one of the possible traversal paths. Thus, the path traversal in the multiple transitivity case may be regarded as being nondeterministic.

4 EMPIRICAL VALIDATION WITH DATA GENERATED FROM THEORETICAL NETWORKS

Our empirical validation study begins with a theoretical Bayesian network to generate samples from which the implication induction algorithm is applied. Thereafter, inferences can be made using the induced implication network whenever a node is observed.

This section presents the results of some experiments which take several Bayesian networks of difference characteristics and generate data samples for carrying out the proposed implication induction and evidential reasoning task. The respective networks used in the experiments are:

1) a single-chain serially connected network,
2) a two-layer parallelly connected network,
3) a five-node multiply connected network, and
4) a ten-node multiply connected network.

Apart from evaluating the validity of our network-induction method by way of examining the results of evidential reasoning in the induced networks, we also conduct comparisons with one of the commonly used probabilistic reasoning approaches, namely *Pearl's stochastic simulation algorithm* for handling general multiply connected Bayesian networks [21]. Pearl's simulation method generates a series of network samples that are consistent with the probabilities of the root nodes and the conditional probabilities of the nonroot nodes in the network. The generated samples (viewed as simulated scenarios) allow for the counting of occurrence frequencies of specific events that in turn give the estimates of the probabilities of those events (or node variables). In reasoning with Pearl's method, whenever a node variable is newly observed, the posterior probabilities of other nodes are estimated through such a stochastic simulation process. One of the major features of Pearl's stochastic simulation method lies in that the samples as generated from the network have already taken into account the effect of node dependencies (as in multiply connected networks). However, to yield precise estimates of the node probabilities, Pearl's method requires a large number of iterations.

In all the experiments that we have conducted, we run 1,000 iterations for each of Pearl's simulations.[2]

4.1 The Experimental Procedure

In all the experiments conducted, the general procedure for validating our algorithm can be summarized into the following basic steps (see also Fig. 5):

1) **Theoretical network specification.** Define a theoretical Bayesian network with prior probabilities of its root nodes and the conditional probabilities of its nonroot nodes given all the combinations of their predecessors.
2) **Simulated data sample generation.** Apply *logic sampling* algorithm (see [16]) to generate two sets of data samples from the network as defined in Step 1; one for constructing implication network and the other for validating the evidential reasoning results. The states of the root nodes are generated based on their prior probabilities as given in the Bayesian network, while the states of the nonroot nodes are generated based on their probabilities conditioned on their immediate parent nodes.
3) **Implication network construction.** Induce implication relationships between pairs of variables by applying the induction algorithm (see Section 2.1) to the simulated data sample set from Step 2. *This step represents the probabilistic knowledge originally modeled by the Bayesian network into a set of implication relationships.*
4) **Implication network propagation.** For each of the testing data samples, randomly select an unknown (i.e., unobserved) node and use its value in the sample as a new observation, and thereafter propagate the belief values for other nodes reachable from the observed one.
5) **Reasoning validation.** For each of the updated nodes, compare the belief value computed based on the propagation and the one given in the testing sample.[3] Output the difference (whose computation scheme will be defined later).
6) If there exists any unobserved node in the testing data sample, then go to Step 4. Otherwise exit.

With regard to Pearl's stochastic simulation scheme, Step 3 is eliminated since the procedure directly involves the theoretical network.

Throughout the experiments, we generate 50 samples from the given Bayesian networks for constructing the implication networks, and another 100 samples for empirically validating the evidential reasoning results. During the statistical testing for implication networks, the criteria for accepting a single relation are: $p_{\min} = 0.5$ and $\alpha_c = 0.4$. This set of parameters enables us to derive implication networks composed of moderate-influence links. Here it should be pointed out that the effects of the $p_{\min}$ and α_c values on the

---

[2] A stochastic simulation of 1,000 iterations, as has been experimentally demonstrated by Shachter and Peot [23], will yield reasonably stabilized prediction results.

[3] In the experiments, if the value of a node `A` as given in the binary data sample is 1, then the belief value $Bel(A)$ is regarded as 1, else $Bel(A) = 0$.

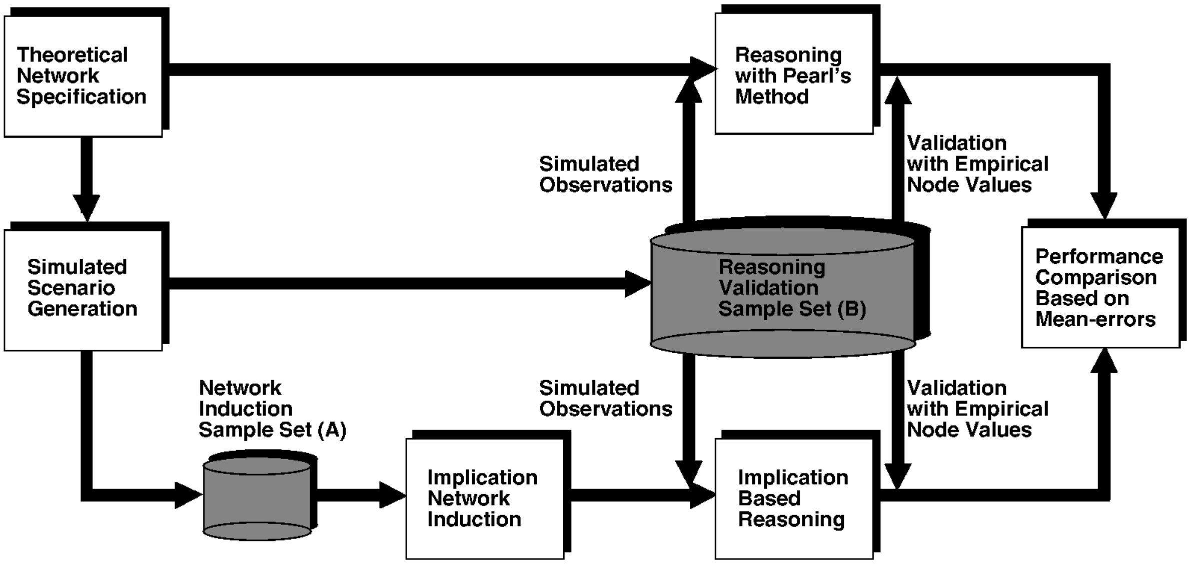


Fig. 5. A flowchart summarizing the major steps involved in our experimentation.

induced network topologies as well as the evidential inference performance could be, to a certain extent, a domain-specific matter depending on the nature of available empirical data. Recent studies that, in part, addressed this issue have been reported in [11].

The validation results for our induced implication networks will also be compared to those produced by Pearl's method. In our present study, we shall utilize the same set of testing sample data to validate Pearl's stochastic simulation-based evidential reasoning method that directly utilizes the given theoretical Bayesian networks.

4.2 The Metrics of Evaluations

In order to evaluate the reasoning performance, we use a set of testing data samples to simulate the observations of the network nodes, and at the same time, let our reasoning program estimate the belief values for other unobserved nodes. After each observation-and-updating session, the following error is calculated, namely, the absolute difference between the actual value in the data sample and the updated belief value. This error can be stated as follows:

$$\Delta_X = | Bel_{emp}(X) - Bel_{est}(X) | \tag{13}$$

where $Bel_{emp}(X)$ denotes the belief value obtained from the empirical (simulated) testing sample in the following manner: If the actual value of the node X in the testing sample is equal to `TRUE`, this belief value is set to 1; else if the observed state is `FALSE`, it is set to 0.

Based on the calculated differences, we can further derive the mean error ($\overline{\Delta}$) as well as the standard deviation (σ_Δ) of the testing data samples; these two metrics are defined as follows:

$$\overline{\Delta} = \frac{1}{N_s \times n_{max}} \sum_{i=1}^{N_s} \sum_{j=1}^{n_{max}} \Delta_{ij} \tag{14}$$

and

$$\sigma\Delta = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} \left(\Delta_i - \overline{\Delta}\right)^2} \tag{15}$$

where n_{max} is the number of nodes in the network. N_s is the number of simulated empirical data samples for the testing.

4.3 Experimental Results

This section presents the results obtained from the empirical validation experiments.

4.3.1 Experiments with Serially Connected and Parallelly Connected Bayesian Networks

Our first two experiments are concerned with the performance of our implication-based evidential reasoning method as well as Pearl's in dealing with topologically simple, semantically clear theoretical Bayesian networks; namely, a single-chain, serially connected network and a two-layer, parallelly connected network. These experiments were designed to see how each method would predict in the case of serial-propagation dominant or parallel-propagation dominant networks, and at the same time, to study how sensitive each method would be to a set of biased empirical testing samples.

Fig. 6 shows a serially connected network used for validating serial propagation performances of the two methods. In dealing with such a serial network, our induced network contains 14 implication rules, as listed in Fig. 7, and its topology resembles closely that of the theoretical network.

Figs. 8 and 9 present the mean prediction accuracy of the two methods tested using the following three testing data sets, respectively:

1) 100 data samples generated using the theoretical Bayesian network.
2) 100 data samples generated using the theoretical Bayesian network. But, the values of some randomly selected nodes are changed *from 0 to 1*. This is to examine the sensitivity of the methods to *positively biased* cases.

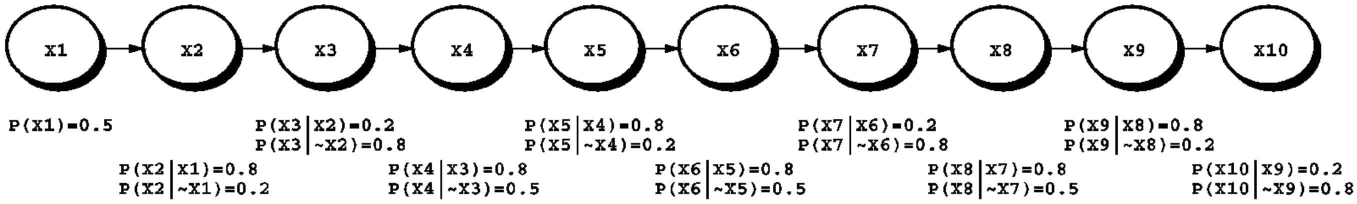


Fig. 6. A single-chain, serially connected Bayesian network where both strong confirming and disconfirming links exist.

$X_1 \Rightarrow X_2$	425	100
	86	389
$\neg X_1 \Rightarrow X_3$	164	361
	326	149
$\neg X_2 \Rightarrow X_3$	100	411
	390	99
$\neg X_3 \Rightarrow \neg X_4$	252	238
	99	411
$X_4 \Rightarrow X_5$	284	67
	140	509
$X_4 \Rightarrow \neg X_8$	125	226
	246	403
$\neg X_5 \Rightarrow \neg X_6$	221	203
	119	457
$X_6 \Rightarrow \neg X_7$	69	271
	520	140
$X_6 \Rightarrow \neg X_8$	91	249
	280	380
$\neg X_7 \Rightarrow \neg X_8$	290	299
	81	330
$\neg X_7 \Rightarrow \neg X_9$	292	297
	132	279
$X_8 \Rightarrow X_9$	300	71
	124	505
$X_8 \Rightarrow \neg X_10$	112	259
	425	204
$X_9 \Rightarrow \neg X_10$	71	353
	466	110

Fig. 7. Based on 50 data samples generated from the theoretical Bayesian network (as shown in Fig. 6) 14 implication relations can be derived. On the right-hand side of each implication relationship is the contingency table, ☐, tested during the induction of the respective implication.

3) 100 data samples generated using the theoretical Bayesian network. But, the values of some randomly selected nodes are changed *from 1 to 0*. This is to study the sensitivity of the methods to *negatively biased* cases.

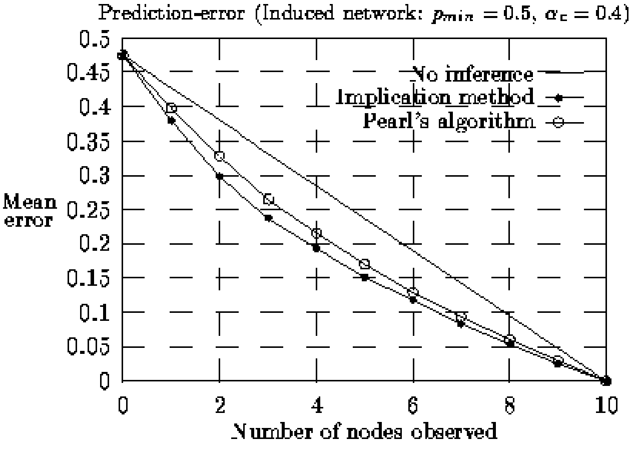


Fig. 8. The mean prediction-error among 100 testing samples, obtained by the implication-based method and Pearl's stochastic simulation methods. The theoretical Bayesian network is single-chain, serially connected. The criteria for the implication induction are: p_{min} = 0.5, α_c = 0.4.

Note that in the latter two validation experiments, the positive and negative biases were introduced in two of the total 10 nodes, hence the mean errors in these two cases were both at 20 percent when all nodes were observed.

Fig. 10 gives a two-layer, parallelly connected theoretical Bayesian network used to validate the performances of our method as well as Pearl's. Several characteristics may be observed from such a parallel network; they are:

1) the conditional influences from the top three nodes toward the bottom three nodes are much stronger than the influences from the bottom to the top, as it is revealed in the prior and posterior probabilities of the network, and
2) the top three nodes are joined at the bottom nodes by disjunctions.

Note that the relations of this network are complex as only the combined negation of the three parent nodes have an effect on the children. Moreover, it has a positive effect on X4 *and* X6 *and a negative effect on* X5. *This type of relation is not adequately captured by binary relation induction techniques such as ours. Thus, this case is playing against the implication network-induction technique.*

Figs. 11 and 12 present the results of experiments with three different sets of testing samples, generated in a similar way as in the preceding serial network case. As it is shown in the figures, the positively biased errors in the testing data would have greater impact than negatively

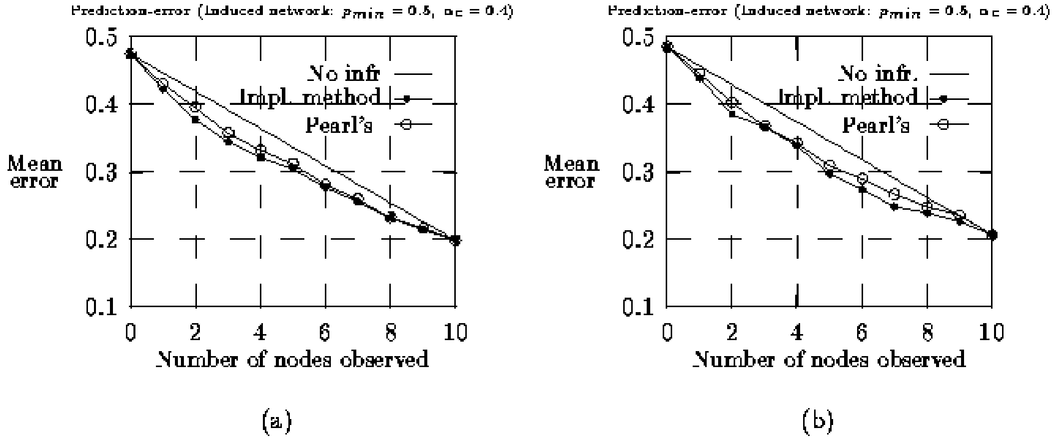


Fig. 9. The mean prediction-error among 100 testing samples, obtained by the implication-based method and Pearl's stochastic simulation method. The theoretical Bayesian network is single-chain, serially connected. The criteria for the implication induction are: p_{min} = 0.5, α_c = 0.4. (a) In the testing samples, the values of 20 percent nodes are changed *from* 0 *to* 1; (b) In the testing samples, the values of 20 percent nodes are changed *from* 1 *to* 0.

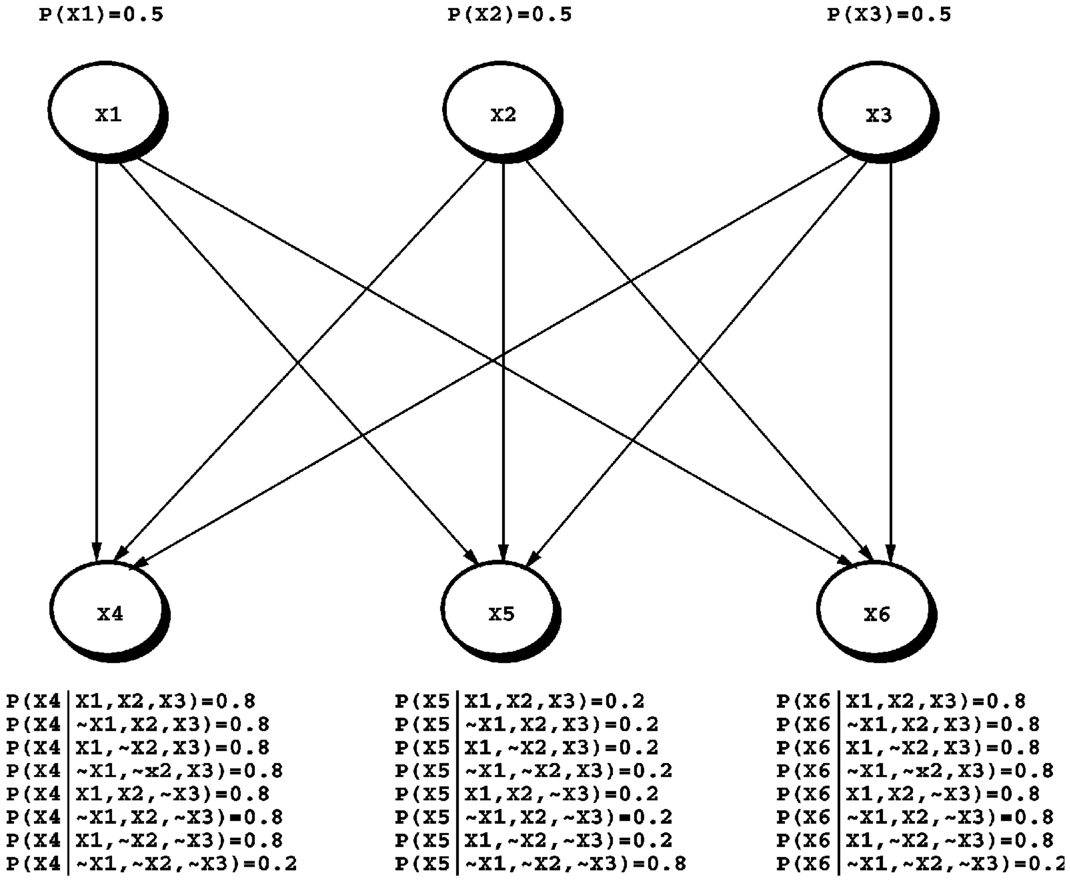


Fig. 10. A two-layer, parallelly connected Bayesian network used to validate the performances of parallel propagations with our method and Pearl's stochastic simulation method.

biased errors. That is, the testing with negatively biased cases would yield slightly better results for both methods. This phenomenon may be understood in view of the characteristics of the original Bayesian network. For instance, if one of the top nodes is TRUE, all three bottom nodes, except node X5, would tend to become TRUE. However, if one of the top nodes is biased toward FALSE, the other two top nodes will still retain their influences on the bottom nodes.

Since in the parallelly connected network testing, both positive and negative biases were introduced at one of the total six nodes, the resulting mean errors at 100 percent observation are around 16 percent.

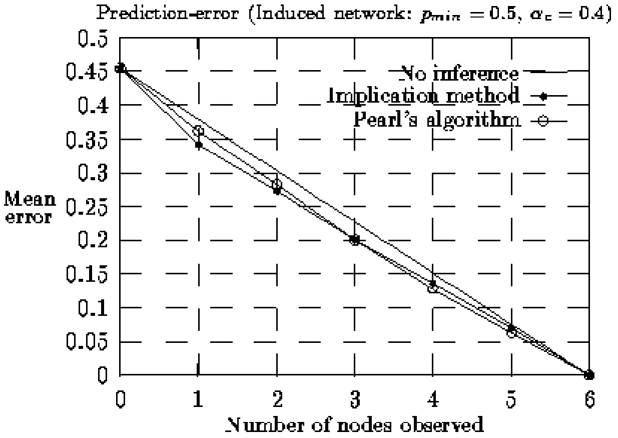


Fig. 11. The mean prediction-error among 100 testing samples, obtained by the implication-based method and Pearl's stochastic simulation method. The theoretical Bayesian network is two-layer, parallelly connected. The criteria for the implication induction are: p_{min} = 0.5, α_c = 0.4.

In both the serially connected and parallelly connected network testing, the difference in performance between our method and Pearl's may be understood by noticing the ways in which the two methods propagate evidential supports. In our method, we propagate the weights following the induced implication rules, whereas in Pearl's method, bidirectional propagations were performed in the theoretical Bayesian network. Thus, Pearl's updating scheme may be more prone to the probabilities of negative events, while ours is more ready to neglect (or "forget" about) the random sampling errors by means of localizing the propagation.

In all the results presented above, it is shown that our implication-based reasoning aggregates evidential weights relatively closer toward the actual ones than Pearl's method, whether it is serial propagation or parallel propagation.

4.3.2 Experiments with Multiply Connected Networks

The first multiply connected network tested is a five-node loop network as proposed in [4] and studied by a number of researchers [23]. This network is given in Fig. 13. The posterior probabilities are set to 0.7.

The implication-based method and Pearl's stochastic simulation method were tested using 100 empirical testing samples generated from the theoretical five-node Bayesian network. Fig. 14 presents the mean errors of the two methods.

Our second multiply connected network is more complicated than the five-node one in a sense that it contains 10 nodes and 12 interconnected conditional links, as shown in Fig. 15.

Fig. 16 shows the experimental results obtained by testing and contrasting the predictions made on the induced networks and with the theoretical network, using a separate set of generated data samples (100 in total). The experiments have shown that while both exhibit a similar behavior, the reasoning based on the induced networks gives slightly better performance with respect to the earlier defined evaluation metrics.

4.3.3 Experiments on Implication-Network Induction with Different α_c

As may be noted from the preceding descriptions of our experiments, the error probability allowed in the network induction (i.e., α_c) was set to 0.4. In order to further get an insight into the effect of α_c setting, let us now take a close look at the performances of the implication-based method with induced networks of different α_c values.

Figs. 17a to 17d present the mean errors of the implication-based method and Pearl's stochastic simulation method. In the figures, the induced networks for the implication-based methods have adopted the following respective α_c values:

(a) $\alpha_c = 0.1$;
(b) $\alpha_c = 0.2$;
(c) $\alpha_c = 0.3$; and
(d) $\alpha_c = 0.5$.

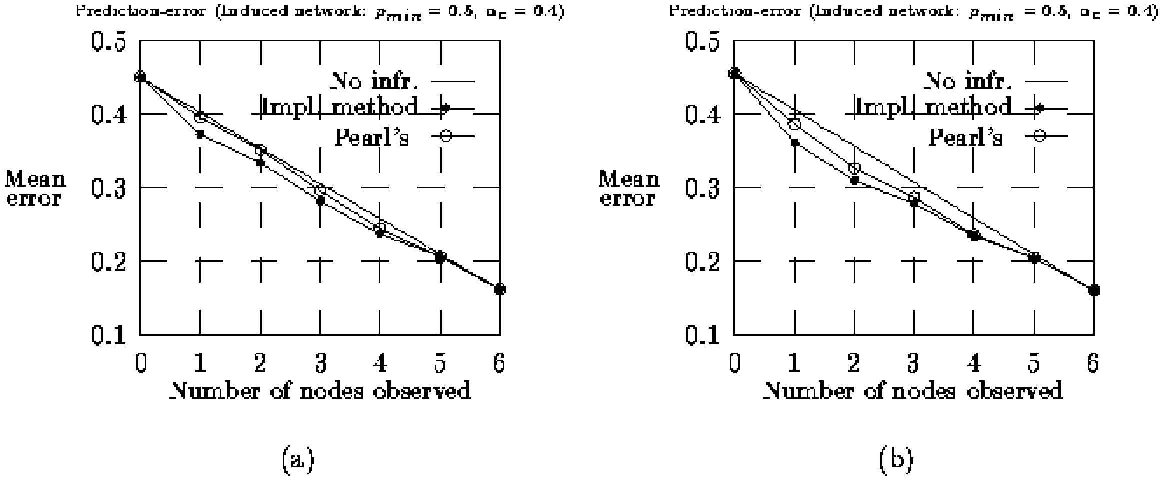


Fig. 12. The mean prediction-error among 100 testing samples, obtained by the implication-based method and Pearl's stochastic simulation method. The theoretical Bayesian network is two-layer, parallelly connected. The criteria for the implication induction are: p_{min} = 0.5, α_c = 0.4. (a) The value of one randomly selected node is changed *from* 0 *to* 1; (b) The value of one randomly selected node is changed *from* 1 *to* 0.

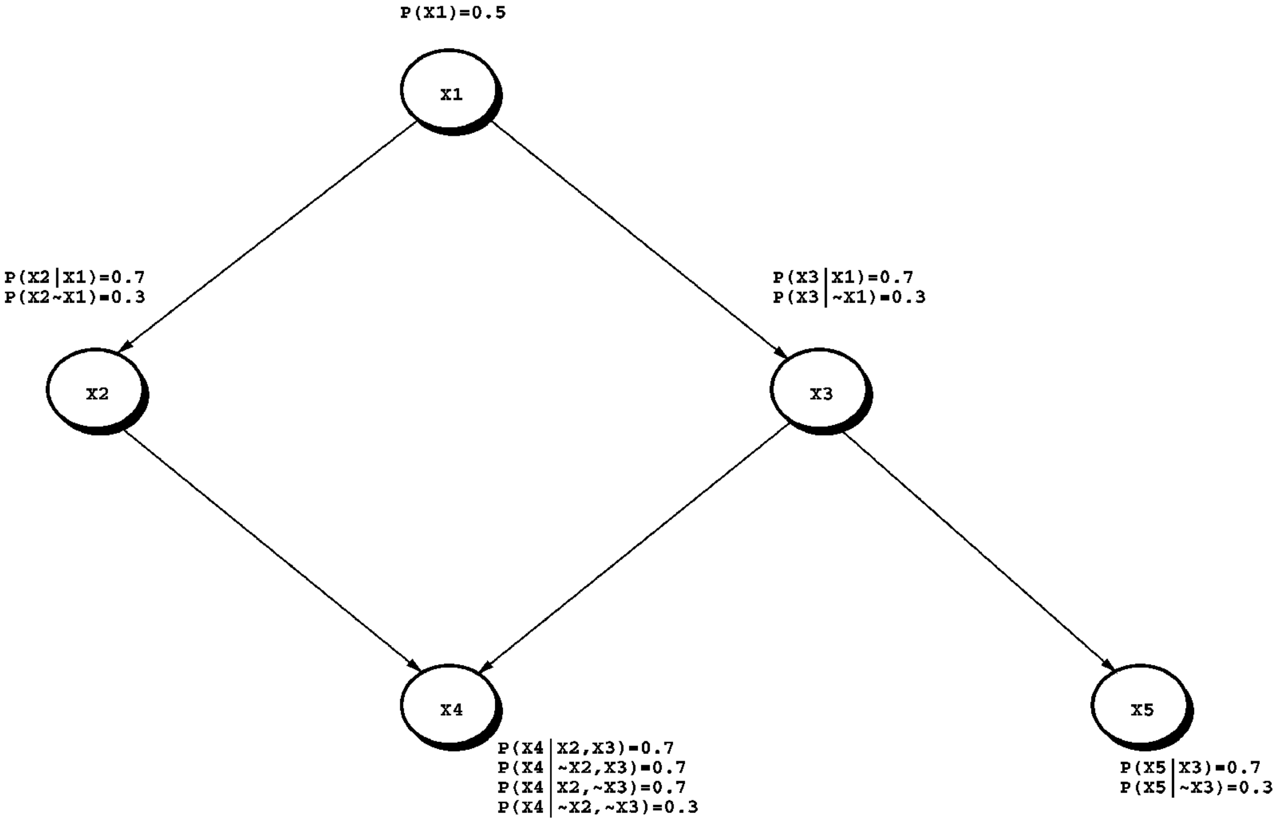


Fig. 13. A five-node, multiply connected Bayesian network as originally proposed and studied in [4].

What is interesting to point out in these experiments is the following: If the α_c value is extremely low, such as 0.1, the performance of the implication-based method would become slightly inferior to Pearl's method. However, as the α_c values increased to 0.2, 0.3, etc., the performances of the two methods would be comparable. Furthermore, when the α_c values exceeded 0.5, the performance of the implication-based method could be far more effective in reducing the prediction errors than Pearl's method. Generally speaking, we can note that when the induced network contains an enough number of implication rules as in Fig. 17d, the implication-based local evidential reasoning scheme as used in our work is capable of propagating evidential supports, and subsequently making correct inferences.

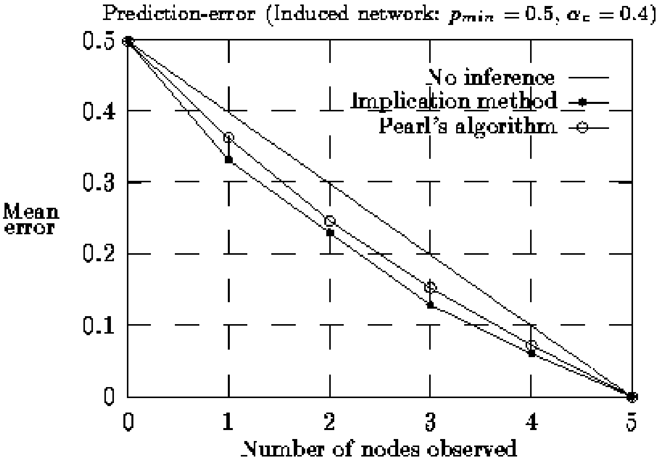


Fig. 14. The mean prediction-error among 100 testing samples, obtained by the implication-based method and Pearl's stochastic simulation method. The theoretical Bayesian network is shown in Fig. 13. The criteria for the implication induction are: $p_{min} = 0.5$, $\alpha_c = 0.4$.

4.4 Discussion

So far, the results of our validation experiments have shown that the performances of the two methods are in general quite similar, with respect to the mean prediction error measure—an evaluation at a global network level. In addition, it should also be mentioned that the comparable performance of the proposed method can be achieved with much less computational cost than Pearl's stochastic simulation method More specifically, the ratio between the actual CPU time required by our method and that by Pearl's is approximately 1:100. For instance, we have conducted the experiments (as mentioned in Section 4.3.2) on a Sun SPARC 20/612MP workstation, and found that it would take Pearl's stochastic simulation method 2,040 msec to complete a 100-iteration—normally it requires 1,000 iterations for reasonable precision—simulation in deriving the posterior probabilities of the network nodes upon an observation. On the other hand, however, it took only 29.75 msec for our implication-based method to generate the predictions about the unobserved nodes.

Several experiments were also carried out on other stochastic simulation methods, such as Shachter's Basic Algorithm with Markov Blanket [23], our initial results

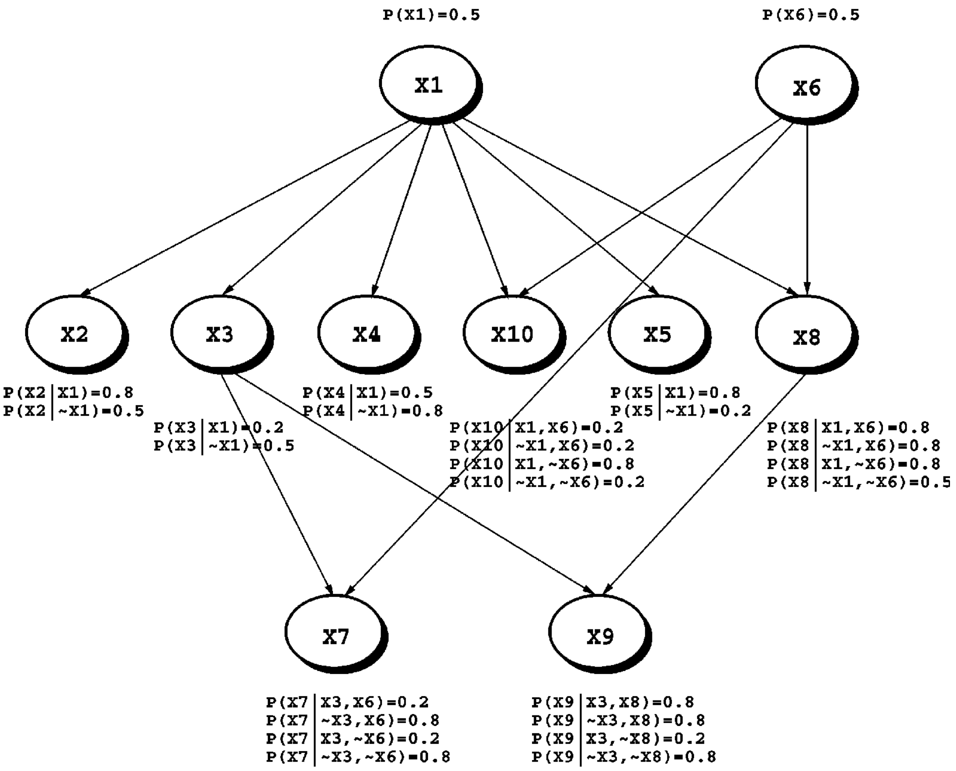


Fig. 15. A multiply connected Bayesian network used for generating samples upon which implication networks were induced and validated.

showed that the asymptotic behaviors of different stochastic simulation methods, by and large, tended to be quite close.

One of the important characteristics of our implication-based method is that the induced implication network contains the major conditional dependence relationships as in the given theoretical network, and when both the theoretical network and the induction condition get stronger, our induced implication network would contain fewer implication relations.

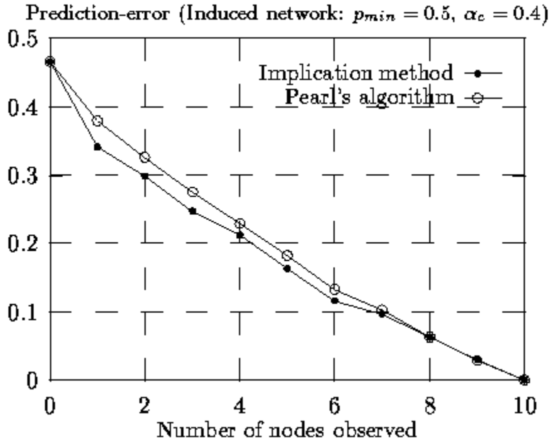


Fig. 16. The mean prediction-error among 100 testing samples, obtained by the implication-based method and Pearl's stochastic simulation method. The theoretical Bayesian network is shown in Fig. 15. The criteria for the implication induction are: $p_{min} = 0.5$, $\alpha_c = 0.4$.

4.4.1 Experiments with Networks Composed of High-Influence Links

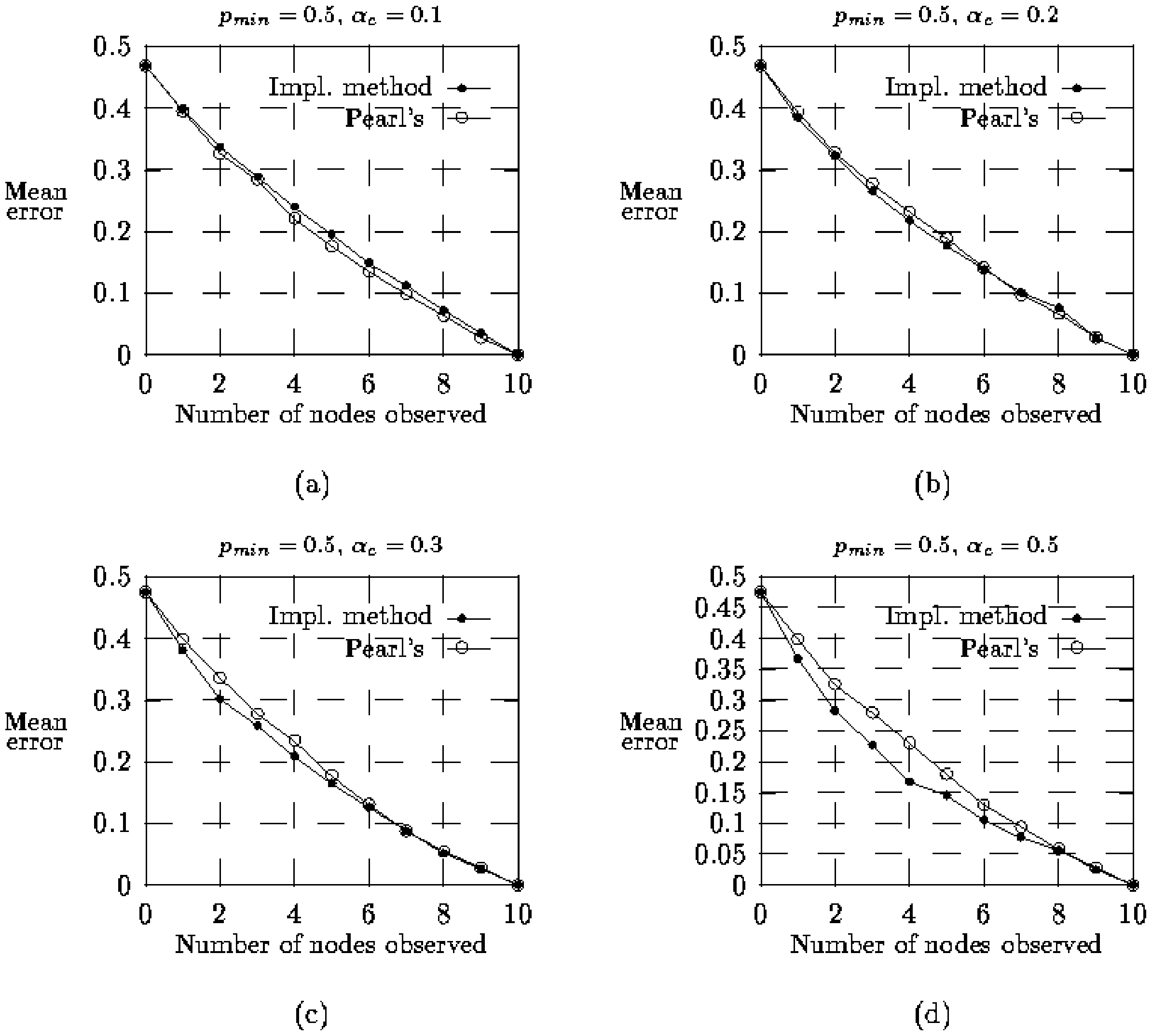
In our present study, we also examined the effect of the conditional probability strength in theoretical networks on the accuracy of evidential reasoning. In order to test the performances of the two methods with networks composed of high-influence links, we modified the multiply connected network as in Fig. 15 by changing the conditional probabilities from previous 0.8 to 0.9.

The results of prediction validation are given in Fig. 18. What is interesting to note is that in this case, when the posterior probabilities increased to 0.9, our method would *not* give performance as good as Pearl's. The explanation for such a phenomenon might be:

1) When the causal nodes in the network are fewer than the diagnostic nodes, our method would not make as strong evidential inferences (e.g., diagnostic inferences) as Pearl's method does.
2) Pearl's method is more accurate to account for the posterior probabilities when the conditionals are strong. Or, in other words, it performs well if the given Bayesian network is less uncertain.

4.4.2 Notes on Network Construction and Reasoning

There are several advantages in our present pairwise implication induction method. First, estimating the conditional probability of A based only on the information of B, $P(A \mid B)$, requires less empirical data than estimating, say, the conditional probability $P(A|B, C, D)$. One may observe that it is usually more convenient to assess, either statistically or subjectively, the conditional probabilities of a certain variable, given fewer contributing sources.



(a) (b)

(c) (d)

Fig. 17. The mean errors of the implication-based method and Pearl's stochastic simulation method. The induced networks for the implication-based methods have adopted the following respective α_c values: (a) $\alpha_c = 0.1$, (b) $\alpha_c = 0.2$, (c) $\alpha_c = 0.3$, and (d) $\alpha_c = 0.5$.

Second, the induced implication network provides a sufficient knowledge representation for carrying out efficient evidential reasoning. Sometimes it might not be necessary to involve *all* the conditional probabilities in the step-by-step unfolding of other probabilities, even if a complete set of the conditionals could perfectly be constructed. For instance, suppose that our empirical data has strongly indicated that marginal probability of a variable A to be 0.7. As we are observing some variables, we update this probability to 0.5, based on both the incoming and outgoing supports. This could be caused by some of the weak links whose conditional probabilities are not so significant. The question now is whether we should still aggregate this updating result even when it fails to give us a better estimate on the posterior probability of A given certain observation. With an exact probabilistic reasoning scheme such as Pearl's local constraint propagation, the answer would be affirmative, as it is believed that any perturbation of the probability would eventually be stabilized at a certain value and that such a value can tell us which value the variable is more likely to take. When maintaining the consistency of

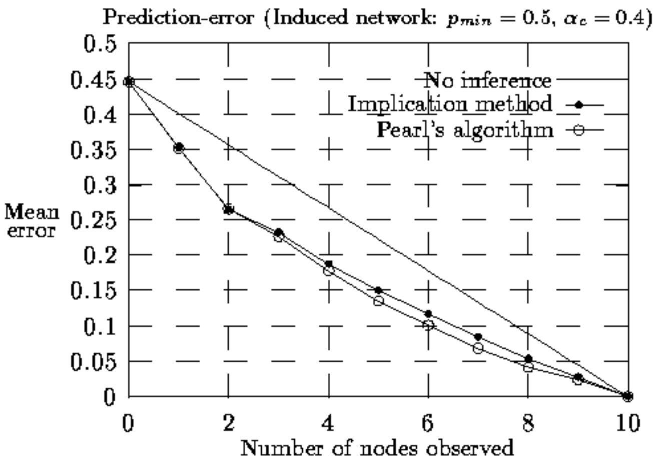


Fig. 18. The mean prediction-error among 100 testing samples, obtained by the implication-based method and Pearl's stochastic simulation method. The theoretical Bayesian network is multiply connected with strong conditional links. The criteria for the implication induction are: $p_{min} = 0.5$, $\alpha_c = 0.4$.

the posterior probabilities, such an approach overlooks the fact that in doing so, much of the error as signified by the weak conditionals (e.g., 0.49) gets propagated to other variables. In other words, the strict adherence to probability theory as in the exact approaches guarantees consistent probabilistic results, but not necessarily the informative predictions or diagnoses.

One of the arguments for involving both top-down predictive and bottom-up diagnostic inferences in evidential reasoning is based on the intuition that a stronger belief in a given hypothesis means a greater expectation for the occurrence of its various manifestations, and a greater certainty in the occurrence of these manifestations adds credence to the hypothesis [20]. But the question that remains is what if either the belief or the certainty is extremely weak. In our opinion, the weak evidential sources should be discarded in order to avoid the aggregation of errors.

Since the goal of evidential reasoning is to make judgments about the unobserved variable, i.e., the information in terms of what is *more plausible* based on some observations, our present implication-based evidential reasoning explicitly represents the strong links of the networks and propagates the belief only along these links. At the same time, it discards the weak links that cause the erroneous prediction perturbation.

5 CONCLUSIONS

This paper has presented a computationally efficient means for automatically inducing implication networks from empirical data samples. The proposed network-induction algorithm was validated through Monte-Carlo simulations in which data samples were stochastically generated from theoretical Bayesian networks. The performances of evidential reasoning in the implication networks as induced from a variety of theoretical Bayesian networks were studied at the global network level. Furthermore, the prediction-error results were contrasted with those of the commonly used stochastic simulations such as Pearl's [21]. Several general observations about the effectiveness of the proposed and Pearl's methods could be made. Generally speaking, the results of reasoning based on the induced networks is comparable to those of Pearl's method, particularly when the uncertainty inherent in the domain knowledge is not neglectable. However, what is most significant is that our method is well suited for applications where Bayesian networks are not known as a priori knowledge.

Wise and Henrion [25] have studied the behavior of a single rule in isolation using some of the commonly applied probabilistic and nonprobabilistic inference methods [1], [6], [18], [21], [24] and compared the performance of these methods on one small rule set. As they noted, the ultimate impact of these differences within a system of many rules would depend on aggregate characteristics of the system or the nature of the situation. This paper offers a new empirical ground for the further understanding of the existing approaches to evidential reasoning in Bayesian networks.

One interesting extension of the present work would be to adjust the weights of implication relationships as new observations are made. An adaptation mechanism has been suggested by Olesen et al. [19] and demonstrated in their aHUGIN system.

APPENDIX
AN EXAMPLE OF POSITIVE IMPLICATION INDUCTION

What follows illustrates how the previously presented algorithm is used to verify the existence of a *positive* implication relation: $\mathtt{A} \Rightarrow \mathtt{B}$ (see Fig. 2).

A.1 Contingency Distributions

In the first step of positive implication induction, a two-dimensional contingency table for variables `A` and `B` is compiled. As computed from an empirical data set, the cells in the contingency table contain the observed joint occurrences for the respective four possible combinations of values. Table 1 shows an example of the contingency table with respective co-occurrences of variables `A` and `B` in a hypothetical data set. In the table, $N_{\cdot\cdot}$ denotes the occurrences of the respective situations. The total numbers of `A`'s and ¬`B`'s can be derived accordingly as follows:

$$N_A = N_{A\wedge B} + N_{A\wedge\neg B} \quad = 21$$
$$N_{\neg B} = N_{A\wedge\neg B} + N_{\neg A\wedge\neg B} = 2$$

TABLE 1
DISTRIBUTION OF OBSERVED OCCURRENCES

	B	¬B
A	20 ($N_{A\wedge B}$)	1 ($N_{A\wedge\neg B}$)
¬A	8 ($N_{\neg A\wedge B}$)	1 ($N_{\neg A\wedge\neg B}$)

A.2 Statistical Tests for Implication Existence

The second step of our induction method consists in an assessment of the numerical constraints imposed by $\mathtt{A} \Rightarrow \mathtt{B}$. More specifically, the assessment is based on the lower tails of binomial distributions $Bin(N_A,\ p_{min})$ and $Bin(N_{\neg B},\ p_{min})$ to test measured conditional probabilities $P(\mathtt{B} \mid \mathtt{A})$ and $P(\neg\mathtt{A} \mid \neg\mathtt{B})$, where

$$N_A = N_{A\wedge B} + N_{A\wedge\neg B}, \qquad N_{\neg B} = N_{A\wedge\neg B} + N_{\neg A\wedge\neg B},$$

and p_{min} is an arbitrary number chosen as the *minimal conditional probability* for an implication relation. For each of the two binomial distributions, we check to see whether Inequality 8 can be satisfied.

Suppose that in this example, $p_{min} = 0.85$; $\alpha_c = 0.20$. Accordingly the binomial distribution for testing $P(\mathtt{B} \mid \mathtt{A})$ can be written as: $Bin(21, 0.85)$. The computation of the lower bound proceeds as follows:

$$\begin{aligned} P(x \le N_{A\wedge\neg B}) &= P(x \le 1) \\ &= P(x = 0) + P(x = 1) \\ &= \binom{21}{0} 0.15^0\, 0.85^0 + \binom{21}{1} 0.15^1\, 0.85^{20} \\ &= 0.155 \end{aligned}$$

hence,

$$P(x \le N_{A\wedge\neg B}) < \alpha_c$$

where symbol

$$\binom{j}{k}$$

represents the number of combinations of k in j. The inference with $\mathtt{A} \Rightarrow \mathtt{B}$ in the *modus ponens* direction is significant with confidence level $(1 - \alpha_c)$.

In a similar way, given $Bin(2,\ 0.85)$, the test for $P(\neg\mathtt{A} \mid \neg\mathtt{B})$ yields:

$$P(x \le N_{A\wedge\neg B}) = \binom{2}{0}\, 0.15^0\, 0.85^0 + \binom{2}{1}\, 0.15^1\, 0.85^1$$

hence,

$$P(x \le N_{A\wedge\neg B}) \not< \alpha_c$$

Since Inequality 8 for the test of $P(\neg\mathtt{A} \mid \neg\mathtt{B})$ is not satisfied, $\mathtt{A} \Rightarrow \mathtt{B}$ cannot be used for *modus tollens* inference. Hence, the positive implication $\mathtt{A} \Rightarrow \mathtt{B}$ is rejected.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] B.G. Buchanan and E.H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, Mass.: Addison-Wesley, 1984.

[2] E. Charniak, "Bayesian Networks Without Tears," *AI Magazine*, pp. 50–63, 1991.

[3] H.L. Chin and G.F. Cooper, "Bayesian Belief Networks Inference Using Simulation," L.N. Kanal, T.S. Levitt, and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence*, pp. 129–147. North-Holland: Elsevier Science, 1989.

[4] G.F. Cooper, "NESTOR: A Computer-Based Medical Diagnostic Aid that Integrates Causal and Probabilistic Knowledge," PhD thesis, Computer Science Dept., Stanford Univ., 1984.

[5] G.F. Cooper and E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Machine Learning*, vol. 9, pp. 309–347, 1992.

[6] A.P. Dempster, "A Generalization of Bayesian Inference," *J. Royal Statistical Soc.*, vol. 30, pp. 205–247, 1968.

[7] M.C. Desmarais, L. Giroux, and S. Larochelle, "Fondements Methodologiques et Empiriques Dun Systeme Consultant Actif pour l'Edition de Texte: Le Projet Edcoach, *Technologies de l'information et Societe*, vol. 4, no. 1, pp. 61–74, 1992.

[8] M.C. Desmarais, L. Giroux, S. Larochelle, and S. Leclerc, "Assessing the Structure of Knowledge in a Procedural Domain," *Proc. Cognitive Science Soc.*, pp. 475–481, Montreal, Aug. 14–17, 1988.

[9] M.C. Desmarais and J. Liu, "Knowledge Assessment Based on the Dempster-Shafer Belief Propagation Theory," Technical Report CRIM–92/09–06, Centre de Recherche Informatique de Montreal, 1992.

[10] M.C. Desmarais and J. Liu, "Exploring the Applications of User-Expertise Assessment for Intelligent Interfaces," *Proc. Int'l CHI '93: Bridges Between Worlds*, Amsterdam, 1993.

[11] M.C. Desmarais, A. Maluf, and J. Liu, "User-Expertise Modeling with Empirically Derived Probabilistic Implication Networks," *Int'l J. User Modeling and User-Adapted Interaction*, vol. 5, nos. 3-4, pp. 283–315, 1996.

[12] J. Gebhardt and R. Kruse, "Reasoning and Learning in Probabilistic and Possibilistic Networks: An Overview," N. Lavrac and S. Wrobel, eds., *Machine Learning: ECML-95, Proc. Eighth European Conf. Machine Learning*, pp. 3–16. Springer-Verlag, 1995.

[13] D. Geiger, "An Entropy-Based Learning Algorithm of Bayesian Conditional Trees," D. Dubois, M.P. Wellman, B.D'Ambrosio, and P. Smets, eds., *Uncertainty in Artificial Intelligence*, pp. 92–97. San Mateo, Calif.: Morgan Kaufmann, 1992.

[14] D. Geiger, A. Paz, and J. Pearl, "Learning Simple Causal Structures," *Int'l J. Intelligent Systems*, vol. 8, pp. 231–247, 1993.

[15] D. Hecherman, *Probabilistic Similarity Networks*, ACM doctoral dissertation award series. Cambridge, Mass.: MIT Press, 1991.

[16] M. Henrion, "Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling," J.F. Lemmer and L.N. Kanal, eds., *Uncertainty in Artificial Intelligence*, pp. 149–163. North-Holland: Elsevier Science, 1988.

[17] D.K. Hildebrand, J.D. Laing, and H. Rosenthal, *Prediction Analysis of Cross Classifications*. New York: John Wiley & Sons, 1977.

[18] H.E. Kyburg, "Bayesian and Non-Bayesian Evidential Updating," *Artificial Intelligence*, vol. 31, pp. 271–293, 1987.

[19] K.G. Olesen, S.L. Lauritzen, and F.V. Jensen, "aHUGiN: A System Creating Adaptive Causal Probabilistic Networks," D. Dubois, M.P. Wellman, B. D'Ambrosio, and P. Smets, eds., *Uncertainty in Artificial Intelligence*, pp. 223–229. San Mateo, Calif.: Morgan Kaufmann, 1992.

[20] J. Pearl, "A Constraint-Propagation Approach to Probabilistic Reasoning," L.N. Kanal and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence*, pp. 357–369. North-Holland: Elsevier Science, 1986.

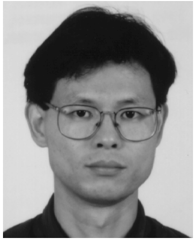
[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, Calif.: Morgan Kaufmann, 1988.

[22] I. Pitas, E. Milios, and A.N. Venetsanopoulos, "A Minimum Entropy Approach to Rule Learning from Examples." *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, no. 4, pp. 621–635, 1992.

[23] R.D. Shachter and M.A. Peot, "Simulation Approaches to General Probabilistic Inference on Belief Networks," M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence*, pp. 221–231. North-Holland: Elsevier Science, 1990.

[24] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, N.J.: Princeton Univ. Press, 1976.

[25] B.P. Wise and M. Henrion, "A Framework for Comparing Uncertain Inference Systems to Probability," L.N. Kanal and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence*, pp. 69–83. North-Holland: Elsevier Science, 1986.



Jiming Liu received an MA degree from Concordia University in 1987, and an MEng degree and PhD degree in electrical engineering from McGill University in 1990 and 1994, respectively. He is now an associate professor of computing studies at Hong Kong Baptist University. Prior to joining HKBU in 1993, he worked for several years as a software engineer, research associate, senior research agent, researcher, and consultant at research and development firms and government laboratories in Canada, including the Canadian Workplace Automation Research Center (government of Canada), Kentek Inc., Virtual Prototypes Inc., and the Computer Research Institute of Montreal (CRIM). The primary themes of Dr. Liu's current research work have been the studies of computational approaches to uncertainty management and reasoning, and autonomous agents. He is a member of the IEEE, the IEEE Computer Society, the IEEE Robotics and Automation Society, and the ACM.

Michel C. Desmarais received his PhD degree in psychology from the University of Montreal. He is currently a researcher with the Computer Research Institute of Montreal and the scientific director of the Computer Assisted Training Group there. He has worked in the areas of cognitive psychology, artificial intelligence, and user interfaces. His most recent efforts have been devoted to applying knowledge assessment technology to training and to performance support systems applications.