

Article

A Hybrid Structure Learning Algorithm for Bayesian Network Using Experts' Knowledge

Hongru Li  and Huiping Guo * 

Information Science and Engineering, Northeastern University, P.O. Box 135, No. 11 St. 3, Wenhua Road, Heping District, Shenyang 110819, China; lihongru@mail.neu.edu.cn

* Correspondence: huipingguo93@gmail.com; Tel.: +86-156-9884-0887

Received: 27 June 2018; Accepted: 18 August 2018; Published: 20 August 2018



Abstract: Bayesian network structure learning from data has been proved to be a NP-hard (Non-deterministic Polynomial-hard) problem. An effective method of improving the accuracy of Bayesian network structure is using experts' knowledge instead of only using data. Some experts' knowledge (named here explicit knowledge) can make the causal relationship between nodes in Bayesian Networks (BN) structure clear, while the others (named here vague knowledge) cannot. In the previous algorithms for BN structure learning, only the explicit knowledge was used, but the vague knowledge, which was ignored, is also valuable and often exists in the real world. Therefore we propose a new method of using more comprehensive experts' knowledge based on hybrid structure learning algorithm, a kind of two-stage algorithm. Two types of experts' knowledge are defined and incorporated into the hybrid algorithm. We formulate rules to generate better initial network structure and improve the scoring function. Furthermore, we take expert level difference and opinion conflict into account. Experimental results show that our proposed method can improve the structure learning performance.

Keywords: Bayesian network; structure learning; explicit knowledge; vague knowledge; hybrid algorithm

1. Introduction

Bayesian networks (BN) is one of the most effective theoretical models for decision making, especially for uncertain knowledge reasoning [1]. In recent years, Bayesian networks have been widely used in a variety of domains such as medical diagnosis [2], device fault diagnosis [3], and system modeling in multiple situations [4–6]. Bayesian network learning is the fundamental topic in Bayesian network research. Generally, Bayesian network learning consists of two parts, structure learning and parameter learning. Of these two parts, structure learning is the core part for Bayesian network learning.

Bayesian network structure learning is to determine every edge in the BN, including judging the existence of edges and the direction of edges. Generally, there are three main algorithms in Bayesian network structure learning: independence-based structure learning algorithms (independence-based algorithms) [7,8], score-based structure learning algorithms (score-based algorithms) [9,10], and hybrid structure learning algorithms (hybrid algorithms) [11–13]. The first group of algorithms use conditional independence (CI) tests to identify conditional independent relationships among variables. The major weakness of these algorithms is high complexity and sensitivity to the CI tests [14]. The second group of algorithms consider structure learning as a structural optimization problem, using a search strategy to select the structure with the highest score of a scoring function which measures the fitting degree of network and data. The major weakness of these algorithms is that they easily to fall into local optima. Both groups of algorithms have drawbacks. Recently, the third group of algorithms,

hybrid algorithms, has become widely used because they combine the advantages of the previous two algorithms. One popular strategy is to use independence-based algorithms (the first stage learning algorithms) to determine the initial network structure and use score-based algorithms (the second stage learning algorithms) to find the highest-scoring network structure.

Bayesian network structure learning from data has been proved to be a NP-hard problem [15], and it faces great challenges when using structural learning algorithms. Firstly, the generation of the initial network structure is very sensitive to CI tests. If the data is insufficient or noisy, the algorithms may work incorrectly. Secondly, the number of the candidate structures grows exponentially with the increase in the number of variables. Finding the highest score network structure through the search strategy is very hard and the search strategy is easy to fall into local optimum. Thirdly, the scoring functions can't work accurately with noisy or insufficient data. Finally, many network structures may have Markov equivalent graph structures that have same joint probability distributions and can't be distinguished by data alone. The effective way to eliminate those limitations is to incorporate experts' knowledge into Bayesian network structure learning.

During the last two decades, a lot of studies have been devoted to incorporating experts' knowledge into Bayesian network structure learning [16–19]. The weakness in the majority of the proposed algorithms is that they assume that the knowledge given by experts can make the causal relationship between nodes in BN structure clear (explicit knowledge), but knowledge provided by experts that cannot make the causal relationship between nodes in BN structure clear (vague knowledge) also exists in the real world. For example, the experts may deem that there is a link between node A and node B. This expert knowledge can't determine the orientation of this pair of nodes in BN. In other words, this expert knowledge is not explicit. We consider it necessary to take this kind of knowledge into consideration in the process of Bayesian network structure learning, rather than just using explicit knowledge alone. Therefore, how to incorporate these two types of experts' knowledge into Bayesian network structure learning is a core problem. We propose a new hybrid algorithm with two types of experts' knowledge to solve the core problem of the use of more extensive experts' knowledge. Using multiple experts, we may also encounter problems of expert level difference and opinion conflict on the same group of node pairs. We incorporate the accuracy of expert into the proposed algorithm and use the principle of the obedience of minority to majority based on credibility to solve those two problems.

In this paper, we propose a new method of using more comprehensive experts' knowledge based on hybrid structure learning algorithm for Bayesian network. We define three kinds of vague knowledge $\{\leftrightarrow, \nrightarrow, -\}$, and three kinds of explicit knowledge $\{\rightarrow, \leftarrow, \leftrightarrow\}$ in this new method. Moreover, we take the accuracy of each expert into account by using six parameters, which we extend the accuracy of an expert in the field of explicit knowledge [17] to the field of both explicit and vague knowledge. The two types of experts' knowledge and the accuracy of each expert are incorporated with training data in hybrid algorithm. In order to improve the utilization of knowledge, this paper applies knowledge to a two-stage hybrid algorithm. In the first stage, firstly, the credibility of experts' knowledge is determined by six accuracy parameters. Then, we fuse the different experts' knowledge of the same node pair based on credibility. Finally, we formulate rules to add and delete edges based on different types of experts' knowledge in the process of generating the initial network structure. In the second stage, we propose an explicit-vague-Bayesian Information Criterion (EVBIC) scoring function which built upon the explicit-accuracy-based score function proposed by [17]. The main advantage of our score function is that it can use vague knowledge. Experimental results reveal that the proposed method can make good use of vague knowledge and explicit knowledge given by experts and can learn a better BN structure than hybrid algorithms.

The remainder of this paper is structured as follows: Section 2 introduces the notations and preliminaries that are used in subsequent sections. Section 3 describes the proposed method. Section 4 details our experimental procedures and presents the experimental evaluation of the algorithm. Finally, Section 5 concludes the work.

2. Preliminaries

In this section, a brief summary of the Bayesian networks and the hybrid structure learning algorithm is introduced. What's more, we present two types of experts' knowledge and they are subdivided into six kinds of experts' knowledge, along with some further notations that will be used throughout the remainder of this paper.

2.1. Bayesian Network

A Bayesian network is a probabilistic graphical model based on probability theory and graph theory. A Bayesian network is a directed acyclic graph that describes the joint probability distribution over a set of random variables, with defining a series of probability independences and conditional independences [20]. A Bayesian network can be represented as a tuple $BN = (G, \rho)$ where $G = (V, E)$. G represents a structure known as a directed acyclic graph (DAG), with a set of nodes $V = \{X_1, X_2, \dots, X_n\}$ and a set of directed edges $E = \{\langle X_1, X_2 \rangle, \langle X_1, X_3 \rangle, \dots, \langle X_i, X_j \rangle\}$. X_i is a random variable. $\langle X_i, X_j \rangle \in E$ represents the directed connect between X_i, X_j , noted as $X_i \rightarrow X_j$, with node X_i known as the parent of node X_j , and node X_j known as the child of node X_i . $\rho = \{\rho_1, \rho_2, \dots, \rho_n\}$ represents a set of conditional probability distributions of nodes. $\rho_i \in \rho$ represents the conditional probability distribution of node X_i . According the chain rule of Bayesian network, the Bayesian network represents the joint probability of all nodes that can be written as the product of the conditional probability distribution of each node:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, X_2, \dots, X_{i-1}) \quad (1)$$

where n represents the number of nodes in Bayesian network, $i = 1, 2, \dots, n$.

Assuming that the conditional probability distribution of the node X_i is only affected by its parent set $\pi(X_i) \subseteq \{X_1, X_2, \dots, X_{i-1}\}$, the Equation (1) can be written as:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | \pi(X_i)) \quad (2)$$

where n represents the number of nodes in Bayesian network, $i = 1, 2, \dots, n$.

2.2. Hybrid Structure Learning Algorithms

As mentioned above, one category of Bayesian network structure learning algorithms is the hybrid structure learning algorithms. They aggregate both independence-based and score-based structure learning algorithms to give full play to the advantages of them and has been widely used at present. One popular strategy is using independence-based algorithms (the first stage learning algorithms) to determine the initial network structure and use it as a seed of score-based algorithms (the second stage learning algorithms) which include search strategies and scoring functions to find the highest score network structure.

2.2.1. The First Stage Structure Learning Algorithms

The first stage structure learning algorithms generally use independence-based structure learning algorithms. The independence-based algorithms determine all the independence and dependence relationships among variables via conditional independence (CI) tests and construct networks that characterize these relationships. The independence-based algorithms can be divided into two processes. The first process is to determine whether the edges exist so that we can obtain the undirected network structure. The second process is to direct the edge orientations, especially to direct edges to form head-to-head patterns (triplets of nodes x, y, z such that x and y aren't adjacent and the arcs $x \rightarrow z$ and $y \rightarrow z$ exist). In this paper, we use the Maximum Information Coefficient (MIC) and conditional

independence tests to generate an initial network structure [21]. The initial network structure may still contain some undirected edges which will be used as an initial solution to the second stage structure learning algorithms.

2.2.2. The Second Stage Structure Learning Algorithms

The second stage structure learning algorithms generally use score-based structure learning algorithms. The score-based algorithms have two main components: a search strategy and a scoring function. The number of possible DAGs is super-exponential in the number of random variables, given by the following function [22]:

$$f(n) = f(x) = \begin{cases} 1, & n = 1 \\ \sum_{i=1}^n (-1)^{i+1} \frac{n!}{(n-i)! \cdot i!} 2^{i(n-i)} f(n-i), & n > 1 \end{cases} \quad (3)$$

It is obvious that the search space is huge. Therefore, heuristic search methods have been used to build the network structure, such as Genetic Algorithms, Ant Colony Optimization, Binary version of Particle Swarm Optimization (BPSO) [23]. In this paper, we use the BPSO as our search method. The BPSO utilizes a cooperative swarm of particles, where each particle represents a candidate solution to the problem (the possible DAGs), to explore the space of possible solutions (the searching space) to optimization of interest.

Scoring functions are usually used to measure the quality of the constructed BNs. There are variety of scoring functions such as the Bayesian Dirichlet equivalent uniform (BDeu) scoring function, the Minimum Description Length (MDL) scoring function and the Bayesian Information Criterion (BIC) scoring function, to mention a few. The detail of BIC scoring function is introduced below since it is used in this study. BIC scoring function is in the form of a penalized log-likelihood (LL) function. Given a training dataset D , the log-likelihood function for a structure can be written as:

$$LL(D) = \sum_{i=1}^n \sum_{k=1}^{r_i} \sum_{j=1}^{q_i} m_{ijk} \ln \frac{m_{ijk}}{m_{ij}} \quad (4)$$

where m_{ijk} is the number of samples in the dataset that $X_i = k$ and its parents are in their j -th configuration, and likewise, m_{ij} is the number of samples in the dataset that variable is i -th and its parents are in their j -th configuration. n is the number of random variables in Bayesian network, r_i is the number of different states of i -th random variable, and q_i is the number of possible configurations for parents of i -th random variable. A penalization is used and BIC scoring function can be written as follows:

$$BIC(D) = \sum_{i=1}^n \sum_{k=1}^{r_i} \sum_{j=1}^{q_i} m_{ijk} \ln \frac{m_{ijk}}{m_{ij}} - \frac{1}{2} (\ln N) \sum_{i=1}^n q_i (r_i - 1) \quad (5)$$

2.3. Edge Types, Kinds and Types of the Experts' Knowledge, and Experts' Accuracies

2.3.1. Edge Types

Given the number of random variables in Bayesian network n , there exists $\frac{n(n-1)}{2}$ different node pairs. From Figure 1, since there are four nodes in this simple Bayesian network structure, the number of node pairs is $N = \frac{4 \times (4-1)}{2} = 6$.

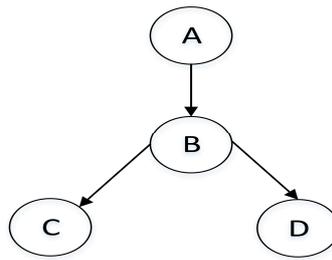


Figure 1. A simple Bayesian network structure.

If the i -th node pair is (X, Y) , the state of the edge between node X and node Y is defined as e_i , where:

- $e_i = \rightarrow$ represents $\langle X, Y \rangle \in E$,
- $e_i = \leftarrow$ represents $\langle Y, X \rangle \in E$,
- $e_i = \leftrightarrow$ represents $\langle \emptyset \rangle \in E$.

(X, Y) represents an unordered pair, and $\langle X, Y \rangle$ represents an ordered pair that node X is the starting point and node Y is the ending point. We can clearly see that there are three types of each edge in the structure: $\{\rightarrow, \leftarrow, \leftrightarrow\}$. As shown in Figure 1, the Bayesian network structure has 6 node pairs that are ordered as (A, B) , (A, C) , (A, D) , (B, C) , (B, D) , (C, D) , and there are $e_1 = \rightarrow$, $e_2 = \leftrightarrow$, $e_3 = \leftrightarrow$, $e_4 = \rightarrow$, $e_5 = \rightarrow$, $e_6 = \leftrightarrow$, which can also represent $\{\langle A, B \rangle, \langle B, C \rangle, \langle B, D \rangle\} \in E$.

2.3.2. Kinds and Types of The Experts' Knowledge, and Experts' Accuracies

(1) The types of the experts' knowledge and representation

In this paper, experts' knowledge can be divided into two types: explicit knowledge and vague knowledge:

Definition 1. *Explicit knowledge is the knowledge that can make the causal relationship between nodes in BN structure clear.*

Definition 2. *Vague knowledge is the knowledge that cannot make the causal relationship between nodes in BN structure clear.*

We define the i -th expert's knowledge of j -th node pair V_j^i taking values in the domain $\{\rightarrow, \leftarrow, \leftrightarrow, -, \rightarrow, \leftarrow\}$, which builds upon the restrictions originally proposed by [24]. The main advantages of our kinds of experts' knowledge are that we simplify kinds of experts' knowledge to be easier to achieve and consider uncertainty of experts' knowledge. Now let us formally define the six kinds of experts' knowledge. Assuming that the j -th node pair is (X, Y) :

Definition 3. $V_j^i = \rightarrow$ is that the i -th expert deems that node X is the parent of node Y in the j -th node pair.

Definition 4. $V_j^i = \leftarrow$ is that the i -th expert deems that node Y is the parent of node X in the j -th node pair.

Definition 5. $V_j^i = \leftrightarrow$ is that the i -th expert deems that node X isn't associated with node Y in the j -th node pair.

Definition 6. $V_j^i = -$ is that the i -th expert deems that node X is associated with node Y in the j -th node pair. However, it is uncertain whether the relationship between node X and node Y is \rightarrow or \leftarrow .

Definition 7. $V_j^i = \rightarrow$ is that the i -th expert deems that node X isn't the parent of node Y in the j -th node pair. This means that it is uncertain whether the relationship between node X and node Y is \leftarrow or \leftrightarrow .

Definition 8. $V_j^i = \leftrightarrow$ is that the i -th expert deems that Y node isn't the parent of node X in the j -th node pair. This means that it is uncertain whether the relationship between node X and node Y is \rightarrow or \leftrightarrow .

According to the above definitions, there exists six kinds of experts' knowledge in BN: $\{\rightarrow, \leftarrow, \leftrightarrow, -, \rightarrow, \leftrightarrow\}$. For convenience, we number the six kinds of experts' knowledge with 1–6.

Obviously, explicit knowledge includes three kinds of experts' knowledge: $\{\rightarrow, \leftarrow, \leftrightarrow\}$, and vague knowledge includes three kinds of experts' knowledge: $\{\leftrightarrow, \rightarrow, -\}$.

In this paper, experts' knowledge can be represented by sets. Explicit knowledge can be represented by a set C which consists of subsets of each expert's explicit knowledge. The set of the i -th expert's explicit knowledge C^i consists of two subsets of node pairs: C_E^i and C_A^i . C_E^i is the set of knowledge which indicate the existence edge between the node pair, including $\{\rightarrow, \leftarrow\}$. C_A^i is the set of knowledge which indicate the absent edge between the node pair, including $\{\leftrightarrow\}$. Each element $\langle x, y \rangle \in C_E^i$ is associated with the corresponding $V_j^i = \rightarrow$, and each element $(x, y) \in C_A^i$ is associated with the corresponding $V_j^i = \leftrightarrow$. C' represents the set of explicit knowledge after fusion and the form is the same as C^i including C'_E and C'_A .

Vague knowledge can also be represented by an another set I which consists of subsets of each expert's vague knowledge. The set of the i -th expert's vague knowledge I^i consists of two subsets of node pairs: I_E^i and I_A^i . V_E^i is the set of knowledge which indicate the existence edge between the node pair, including $\{-\}$. I_A^i is the set of knowledge which indicate the absent edge between the node pair, including $\{\leftrightarrow, \rightarrow\}$. Each element $(x, y) \in I_E^i$ is associated with the corresponding $V_j^i = -$, and each element $\langle x, y \rangle \in I_A^i$ is associated with the corresponding $V_j^i = \rightarrow$. I' represents the set of vague knowledge after fusion and the form is the same as I^i including I'_E and I'_A .

(2) Experts' accuracies

Considering the actual situation that the experts are heterogeneous. In this paper, the accuracy of an expert is represented by six parameters, which we extend the accuracy of an expert in the field of explicit knowledge [17] to the accuracy of an expert in the field of vague knowledge:

- γ_1 : The probability of the correct orientation in the actual network in the field of explicit knowledge;
- γ_2 : The probability of the reverse orientation in the actual network in the field of explicit knowledge;
- γ_3 : The probability of correctly detecting the absent edges in the actual network in the field of explicit knowledge;
- β_1 : The probability of the correct orientation in the actual network in the field of vague knowledge;
- β_2 : The probability of the reverse orientation in the actual network in the field of vague knowledge;
- β_3 : The probability of correctly detecting the absent edges in the actual network in the field of vague knowledge.

From the above, there are two types of accuracy of an expert: the accuracy of an expert in the field of explicit knowledge and the accuracy of an expert in the field of vague knowledge. In this paper, we use a superscript such as $\gamma_1^i, \gamma_2^i, \gamma_3^i, \beta_1^i, \beta_2^i, \beta_3^i$ to represent six accuracy parameters of the i -th expert.

3. Using Different Types of Experts' Knowledge within the Hybrid Structure Learning Algorithm

In order to improve the learning performance and utilization of knowledge, this paper applies knowledge to two stage algorithm of hybrid algorithms respectively.

3.1. Using Different Types of Experts' Knowledge with the First Stage Structure Learning Algorithm

In order to guide the search of the second stage structure learning algorithms better, we need to determine a better initial network structure in the first stage structure learning algorithms. In this

paper, we formulate rules for adding and deleting edges of different types of experts' knowledge to modify the initial network structure in order to improve the accuracy of the initial network structure and improve the overall performance of the algorithm. Firstly, the credibility of experts' knowledge is determined according to the accuracy parameters of experts. In addition, the different experts' knowledge is fused based on the credibility of experts' knowledge. The purpose of these two steps is to fuse different experts' knowledge on the same set of nodes based on credibility. Finally, by formulating rules for adding and deleting edges of different types of experts' knowledge, the fusion of experts' knowledge is modified the initial network structure based on credibility. It can be divided into the following three steps.

(1) To determine the credibility of experts' knowledge

The credibility of experts' knowledge is the probability that experts' knowledge is true. Different kinds of experts' knowledge have different credibility. The credibility θ of experts' knowledge is determined according to the accuracy parameters of experts.

θ_l^i represents the credibility that the experts' knowledge is the l -th kind and the expert is the i -th. Take the i -th expert as an example to explain the credibility of different kinds of expert knowledge $\theta_l^i, l \in \{1, 2, 3, 4, 5, 6\}$.

The i -th expert' explicit knowledge which indicates the existence edge between the node pair includes $\{\rightarrow, \leftarrow\}$. The two kinds of knowledge don't essentially differ and the credibility is the same, and credibility of two kinds of knowledge can be written as $\theta_1^i = \theta_2^i = \gamma_1^i$.

The i -th expert' explicit knowledge which indicates the absent edge between the node pair includes $\{\leftrightarrow\}$, and credibility of this kind of knowledge can be written as $\theta_3^i = \gamma_3^i$.

The i -th expert' vague knowledge which indicates the existence edge between the node pair includes $\{-\}$, and credibility of this kind of knowledge can be written as $\theta_4^i = \beta_1^i + \beta_2^i$.

The i -th expert' vague knowledge which indicates the absent edge between the node pair includes $\{\rightarrow, \leftarrow\}$. The two kinds of knowledge don't essentially differ and the credibility is the same, and credibility of two kinds of knowledge can be written as $\theta_5^i = \theta_6^i = 1 - \beta_2^i$.

(2) The different experts' knowledge fusion based on the credibility of experts' knowledge

As mentioned earlier, the i -th expert's knowledge of j -th node pair V_j^i takes values in the domain $\{\rightarrow, \leftarrow, \leftrightarrow, \leftarrow, \rightarrow, -\}$. For a specific node pair, the experts' knowledge may not be inconsistent. We take the following four steps to fuse different experts' knowledge for the j -th node pair:

- (a) Divide the experts for the j -th node pair into six sets including $G_1^j, G_2^j, G_3^j, G_4^j, G_5^j$ and G_6^j (Define G_l^j as a set of experts for the j -th node pair with the same kind l th of knowledge).
- (b) Sum and normalize the credibility of the six kinds of knowledge for the j -th node pair:

$$P_l^j = \frac{\sum_{i \in G_l^j} \theta_l^i}{\sum_{l \in \{1,2,3,4,5,6\}} \sum_{i \in G_l^j} \theta_l^i} \tag{6}$$

- (c) Divide the close interval $[0, 1]$ into six subintervals including $[0, P_1^j), [P_1, P_1^j + P_2^j), [P_1^j + P_2^j, P_1^j + P_2^j + P_3^j), [P_1^j + P_2^j + P_3^j, P_1^j + P_2^j + P_3^j + P_4^j), [P_1^j + P_2^j + P_3^j + P_4^j, P_1^j + P_2^j + P_3^j + P_4^j + P_5^j)$ and $[P_1^j + P_2^j + P_3^j + P_4^j + P_5^j, 1]$. For a $rand(1)$ which is a random number $[0, 1]$, if $rand(1) \in I_k$ where k is the k -th subinterval as previously described, choose the l -th kind of experts' knowledge where $l = k$ as the result of knowledge fusion for j -th node pair. These six kinds of knowledge are not mutually exclusive, so we cannot simply choose the knowledge with higher P_l^j . We divide the close interval $[0, 1]$ into six subintervals whose length are $P_1^j, P_2^j, P_3^j, P_4^j, P_5^j$ and P_6^j , which is the same with roulette. And we

use a random number located in [0,1] which trends to choose the knowledge with higher P_i^j . Put the result into the set of C' or I' .

- (d) Calculate the credibility θ_k^j of the result of knowledge fusion for the j -th node pair:

$$\theta_k^j = \frac{\sum_{i \in G_k^j} \theta_k^i}{\text{card}(G_k^j)} \tag{7}$$

where $\text{card}(G_k^j)$ is a function that returns the number of elements in the set G_k^j . Put the θ_k^j into the set of credibility B . The experts' knowledge set C' and I' and the corresponding credibility set B after fusion will be used as inputs to the following rule.

- (3) Modify the process of generating the initial network structure by rules for adding and deleting edges of different types of experts' knowledge

In this paper, we use the rule-based method to formulate the rules for adding and deleting edges of different types of experts' knowledge. Based on the rules, we modify the undirected network structure that is determined by the first process in the first stage structure learning and the partial directed network structure that is determined by the second process in the first stage structure learning. The rules for adding and deleting edges of different types of experts' knowledge are described below.

Rules for adding and deleting edges of vague knowledge include the rule for adding the edge in the undirected network structure and the rule for deleting the edge in the partial directed network structure.

Rule 1. (The rule for adding the edge in the undirected network structure)

$H' = (V, E'_H)$ represents the undirected network structure after adding the edge, and $H = (V, E_H)$ represents the undirected network structure before adding the edge. For $(X, Y) \in I'_E$, including $\{-\}$, the credibility of knowledge is θ_4^j , if $\text{rand}(1) \in (0, \theta_4^j)$, $E'_H = E_H \cup (X, Y)$.

Rule 2. (The rule for deleting the edge in the partial network structure)

$H' = (V, E'_H)$ represents the partial network structure after deleting the edge, and $H = (V, E_H)$ represents the partial network structure before deleting the edge. For $X, Y \in I'_A$, including $\{\leftrightarrow, \nrightarrow\}$, the credibility of knowledge is θ_1^j , with $l = 5$ or 6 , if $\text{rand}(1) \in (0, \theta_1^j)$, $E'_H = E_H \setminus \langle X, Y \rangle$.

Rules for adding and deleting edges of explicit knowledge include the rule for deleting the edge in the undirected network structure and the rule for adding the edge in the partial directed network structure.

Rule 3. (The rule for deleting the edge in the undirected network structure)

$H' = (V, E'_H)$ represents the undirected network structure after deleting the edge, and $H = (V, E_H)$ represents the undirected network structure before deleting the edge. For $(X, Y) \in C'_A$, including $\{\leftrightarrow\}$, the credibility of knowledge is θ_3^j , if $\text{rand}(1) \in (0, \theta_3^j)$, $E'_H = E_H \setminus (X, Y)$.

Rule 4. (The rule for adding the edge in the partial network structure)

$H' = (V, E'_H)$ represents the partial network structure after adding the edge, and $H = (V, E_H)$ represents the partial network structure before adding the edge. For $X, Y \in C'_E$, including $\{\rightarrow, \leftarrow\}$, the credibility of knowledge is θ_1^j , with $l = 1$ or 2 , if $\text{rand}(1) \in (0, \theta_1^j)$, $E'_H = E_H \cup \langle X, Y \rangle$.

The schematic diagram of using different types of experts' knowledge with the first stage structure learning algorithm is shown in Figure 2.

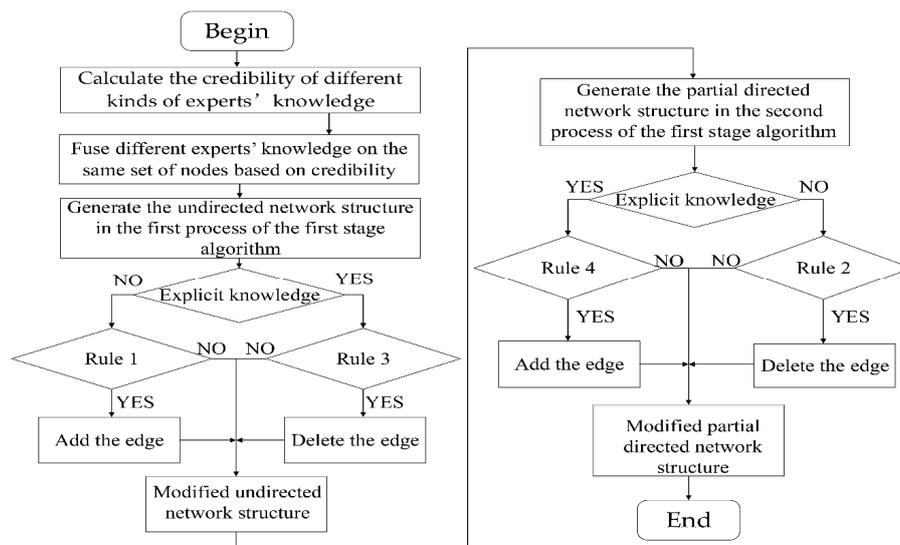


Figure 2. The schematic diagram of using different types of experts’ knowledge with the first stage structure learning algorithm.

3.2. Using Different Types of Experts’ Knowledge with the Second Stage Structure Learning Algorithm

The second stage structure learning algorithms use a search strategy to select the structure with the highest score of a scoring function. The key to select the “optimal” structure is the scoring function.

Explicit-accuracy-based scoring function is proposed by [17]. First, [17] sets up three various problem models. Then, [17] derives some independence statements from three models using the principle of d-separation. Last, scoring function is derived with simplification of independence statements derived above. And this scoring function is given as:

$$Score_{explicit} = \log P(D|G) + \sum_{j=1}^R \sum_{i=1}^N \log P(V_i^j | e_i, \gamma^j) \tag{8}$$

This scoring function is composed of two parts: $\log P(D|G)$ and $\sum_{j=1}^R \sum_{i=1}^N \log P(V_i^j | e_i, \gamma^j)$. The first part is the marginal likelihood part of BDeu scoring function [18]. $P(V_i^j | e_i, \gamma^j)$ in the second part indicates the probability of V_i^j given e_i and γ^j and is computed using the decision tree shown in Figure 3.

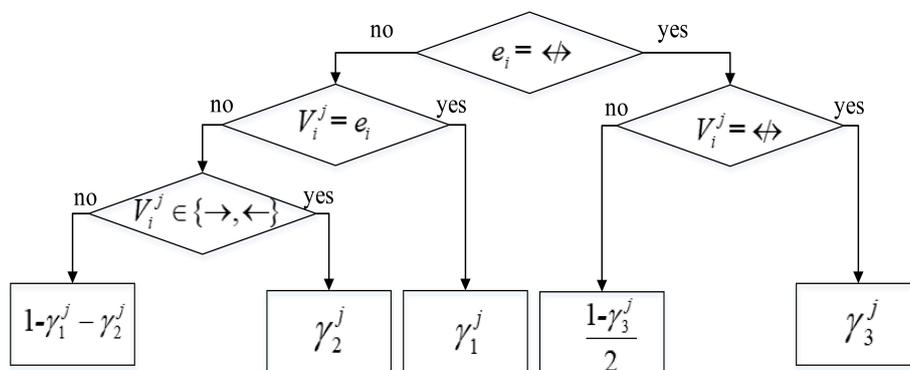


Figure 3. Decision tree for computing $P(V_i^j | e_i, \gamma^j)$.

We can comprehend this scoring function in this way. The first part of Explicit-accuracy-based scoring function means the scoring function which measures the goodness of network to data. The second part of Explicit-accuracy-based scoring function means a penalization which measures the difference between experts' knowledge with network. Experts' knowledge here only refers to the explicit knowledge. Based on this, in this paper, the Explicit-Vague-BIC (EVBIC) scoring function is proposed and given as:

$$Score_{EVBIC} = BIC(D) + \sum_{j=1}^R \sum_{i=1}^{N_explicit} \log P(V_{explicit_i^j} | e_i, \gamma^j) + k \sum_{j=1}^R \sum_{i=1}^{N_vague} \log P(V_{vague_i^j} | e_i, \beta^j), \quad (9)$$

where $N_explicit$ is the number of node pairs with explicit knowledge, and N_vague is the number of node pairs with vague knowledge. $V_{explicit_i^j}$ is the three kinds of explicit knowledge and $V_{vague_i^j}$ is the three kinds of vague knowledge. The EVBIC scoring function is composed of three parts. In the first part, we use the BIC scoring function as the scoring function which measures the goodness of network to data. The second and third part represent penalizations which measure the difference between experts' knowledge with network. The second part represents the penalization which measures the difference between explicit knowledge and network. The third part represent the penalization which measures the difference between vague knowledge and network. In the same case, the vague knowledge is more ambiguous and weaker for building network structure than explicit knowledge. Therefore, we consider using a coefficient $k \in (0, 1)$ to weigh the contribution of explicit knowledge and vague knowledge. Considering each vague knowledge corresponds to two kinds of relationship between nodes in BN structure, in this paper, we take $k = 0.5$.

The term $P(V_{explicit_i^j} | e_i, \gamma^j)$ in the second part in Equation (9) is calculated as the decision tree shown in Figure 3. The calculation of the term $P(V_{vague_i^j} | e_i, \beta^j)$ in the third part in Equation (9) is same as Figure 3, as shown in Figure 4.

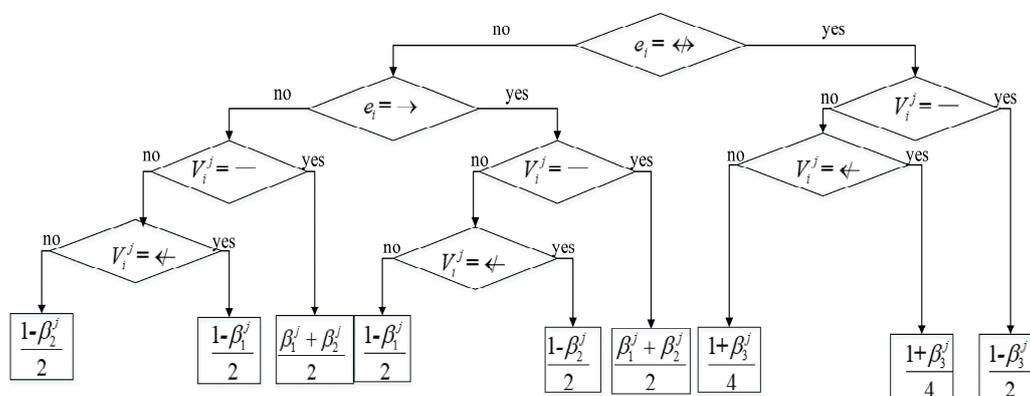


Figure 4. Decision tree for calculating $P(V_{vague_i^j} | e_i, \beta^j)$.

4. Experiments

4.1. Experimental Setup

All the simulation work is implemented and executed in MATLAB 2016b, using the Bayesian network toolbox FullBNT-1.0.7, the maximum information coefficient toolbox minepy-1.2.1 and the graph toolbox matgraph-2.0. The PC has a CPU of Intel 3.40 GHz, an 8 GB memory and Windows 10 operating system.

We use the well-known benchmarks of BN: Alarm and Asia as the experimental network. The Alarm network is a medical diagnostic system for patient monitoring, with 37 nodes and 46 arcs. Because of the complexity of the network structure, it is considered as a standard network to measure

the quality of BN structure learning. The Asia network is a small Bayesian network for virtual medical cases, with 8 nodes and 8 arcs. Figures 5 and 6 shows the structure of the Alarm network and Asia network. The experiment network mainly uses Alarm network, the default network.

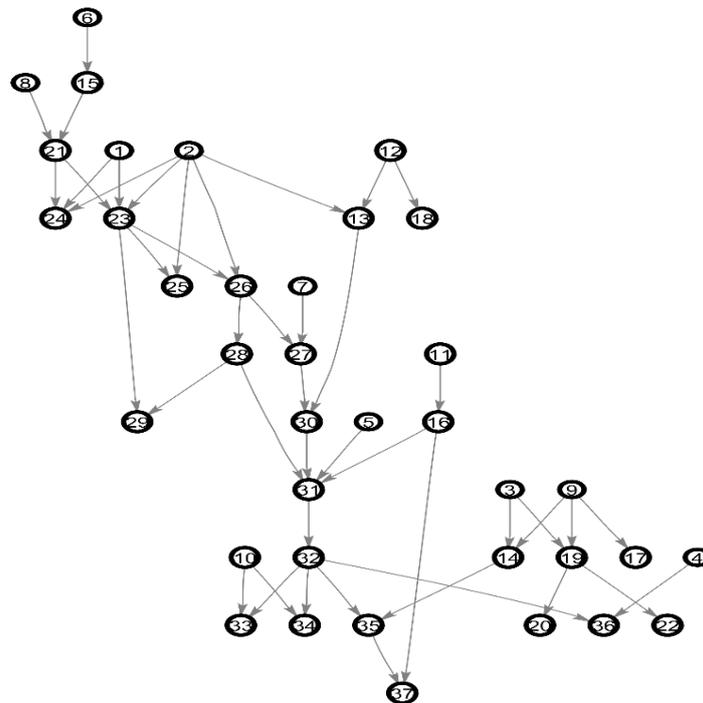


Figure 5. The Alarm network.

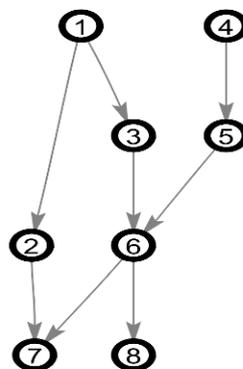


Figure 6. The Asia network.

We measure the operators to reconstruct the original network from the learned network, including the number of deleted arcs (D), the number of added arcs (A) and the number of inverted arcs (I). The total number of the three operators is called Structural Hamming Distance (SHD). Measuring the same edges between the learned network and the original network is called Correct Edges (C). The smaller A, D, I and SHD are, the better the structure learning is. The bigger C is, the better the structure learning is. The measures of performance are: (1) BIC scoring function, which also guides the score-based structure learning algorithms. The higher the BIC score is, the better the structure learning is. (2) The measures of the structural difference between the learned and original networks such as A, D, I, SHD and C. In the experiments, we consider $E = 10$ and corresponding ten groups of experts' accuracies. Table 1 lists the details of ten group parameters of experts' accuracies.

Table 1. The accuracy parameters assigned to experts.

E	γ_i^j, β_i^j	γ_1^j, β_1^j	γ_2^j, β_2^j	γ_3^j, β_3^j
1		0.65	0.15	0.7
2		0.65	0.16	0.75
3		0.67	0.15	0.78
4		0.7	0.17	0.77
5		0.75	0.14	0.8
6		0.78	0.11	0.81
7		0.8	0.12	0.82
8		0.72	0.2	0.79
9		0.79	0.18	0.8
10		0.8	0.13	0.8

In this paper, we use a parameter $v \in [0, 1]$ as the percentage of experts' knowledge. In this paper, the value of the parameter v is selected from $\{0.4, 0.5, 0.6\}$. We use the hybrid structure learning algorithm of MIC-BPSO with BIC scoring function to learn the network structure by the data and three different cases of experts' knowledge. Our experiments are divided into three cases for the Results and Discussion. For the first case, we use three different cases of experts' knowledge in the first stage of the hybrid algorithm in order to verify the validity of using the different types of experts' knowledge in the first stage algorithm. For the second case, we use three different cases of experts' knowledge in the second stage of the hybrid algorithm in order to verify the validity of using the different types of experts' knowledge in the second stage algorithm. For the third case, we use three different cases of experts' knowledge in two stage of the hybrid algorithm in order to verify the validity of using the different types of experts' knowledge in the hybrid algorithm.

In the experiments, we have three cases of experts' knowledge:

- (1) Explicit knowledge: this case of experts' knowledge only uses explicit knowledge in BN.
- (2) Vague knowledge: this case of experts' knowledge only uses vague knowledge in BN.
- (3) EV knowledge: this case of experts' knowledge uses the mixed knowledge including explicit knowledge and vague knowledge in BN.

In our experiments, we use 10 different datasets with 2000 samples and 10 different datasets with 5000 samples in the Alarm network and we use 10 different datasets with 500 samples and 10 different datasets with 2000 samples in the Asia network. For each tuple {case of experts' knowledge, v }, we generate 10 different experts' knowledge sets. In each experiment, we use one dataset and one experts' knowledge set to learn the BN structure.

4.2. Results and Discussion

4.2.1. Results and Discussion Using the Different Types of Experts' Knowledge in the First Stage Algorithm

We calculate the A, D, I and SHD of the learned network. MR is the initial Mean Result, representing 10 runs of the average results. BR is the initial Best Result, representing 10 runs of the optimal value. The term data means using the hybrid algorithm to learn structure without experts' knowledge and is equivalent to $v = 0$. Table 2 shows the results of this experiment.

The smaller A, D, I and SHD are, the better the structure learning is. The SHD shows the total difference between the learned and original networks and A, D, I represent the details of different kinds of difference between the learned and original networks. From Table 2, we can easily find that the results of learning with each case of experts' knowledge are better than the results of learning without experts' knowledge, which proves that the proposed rules using different types of experts' knowledge in the first algorithm is effective. Compared with three cases of experts' knowledge, the experimental

results indicate that learning with explicit knowledge and vague knowledge is better than learning with explicit knowledge alone.

Table 2. A, D, I and SHD values of learned networks in the first Results and Discussion case.

The Case of Knowledge	Measures of Performance	2000				5000				
		0.4	0.5	0.6	Data	0.4	0.5	0.6	Data	
Vague knowledge	MR	SHD	7.5	6.9	7	8.6	5.4	5.3	5.2	6.7
		D	2.9	2.6	2.4	3.1	1.9	1.7	2.1	2.1
		A	1	0.9	0.7	1.2	1	1	1.1	1.6
		R	3.6	3.4	3.9	4.3	2.5	2.6	2.2	3
	BR	SHD	5	5	5	7	2	2	3	6
Explicit knowledge	MR	SHD	7.3	6.4	5.6	8.6	5.1	4.6	4.5	6.7
		D	2.9	2.6	2.6	3.1	1.9	1.6	1.8	2.1
		A	1	0.9	0.6	1.2	1.1	0.9	0.8	1.6
		R	3.4	2.9	2.4	4.3	2.1	2.1	1.9	3
	BR	SHD	5	4	4	7	2	2	2	6
EV knowledge	MR	SHD	6.7	6	5.4	8.6	4.8	4.6	4.1	6.7
		D	2.8	2.7	2.6	3.1	1.7	1.4	1.2	2.1
		A	0.8	0.6	0.6	1.2	1	0.7	0.7	1.6
		R	3.1	2.7	2.2	4.3	2.1	2.5	2.2	3
	BR	SHD	5	3	3	7	3	2	2	6

Because the proposed method in the first stage structure learning algorithm takes no account of BIC scoring function, we can compare the learned network with experts' knowledge with the learned network with data by calculating BIC scoring function. Tables 3 and 4 show the results of this experiment.

The higher BIC score is, the better the structure learned is. From Tables 3 and 4, we can see that the BIC scores of learning with each case of experts' knowledge are higher than the BIC scores of learning without experts' knowledge. It is shown that the proposed rules using different types of experts' knowledge in the first algorithm is effective. Comparing with three cases of experts' knowledge, we can easily find that learning with explicit knowledge and vague knowledge is better than learning with explicit knowledge alone.

Table 3. BIC score of learned networks with 2000 samples in the first Results and Discussion case.

The Case of Knowledge	Measures of Performance	2000				
		0.4	0.5	0.6	Data	
Vague knowledge	MR	BIC score	-20,206.9	-20,202.5	-20,108.5	-20,288.4
	BR	BIC score	-19,900.5	-19,870.6	-19,472	-19,929
Explicit knowledge	MR	BIC score	-20,013.2	-19,904.8	-19,829.2	-20,288.4
	BR	BIC score	-18,925.7	-18,903.8	-18,212.9	-19,929
EV knowledge	MR	BIC score	-19,813.5	-19,776.23	-19,738.14	-20,288.4
	BR	BIC score	-18,212.94	-18,212.94	-17,893.82	-19,929

Table 4. BIC score of learned networks with 5000 samples in the first Results and Discussion case.

The Case of Knowledge	Measures of Performance	5000				
		0.4	0.5	0.6	Data	
Vague knowledge	MR	BIC score	-48,571.2	-48,544	-48,467.7	-48,689.4
	BR	BIC score	-48,044.4	-47,932.4	-47,581.1	-48,200.4
Explicit knowledge	MR	BIC score	-48,390	-48,095.6	-48,083.5	-48,689.4
	BR	BIC score	-47,886.6	-46,305	-47,000.3	-48,200.4
EV knowledge	MR	BIC score	-47,923.74	-47,823.73	-47,813.43	-48,689.4
	BR	BIC score	-46,682.75	-46,305	-45,768.92	-48,200.4

4.2.2. Results and Discussion Using the Different Types of Experts' Knowledge in the Second Stage Algorithm

We calculate the A, D, I and SHD of learned networks. Table 5 shows the results of this experiment.

From Table 5, we can easily find that the results of learning with each case of experts' knowledge are better than the results of learning without experts' knowledge. It is shown that the proposed the EVBIC scoring function using different types of experts' knowledge in the second algorithm is effective. Comparing with three cases of experts' knowledge, we can easily find that learning with explicit knowledge and vague knowledge is better than learning with explicit knowledge alone.

Table 5. A, D, I and SHD values of learned networks in the second Results and Discussion case.

The Case of Knowledge	Measures of Performance	2000				5000				
		0.4	0.5	0.6	Data	0.4	0.5	0.6	Data	
Vague knowledge	MR	SHD	7.7	7.6	7.1	8.6	6.1	5.8	5.5	6.7
		D	2.8	2.9	3	3.1	2.2	2.1	2	2.1
		A	1	1	1	1.2	1.5	1.5	1.4	1.6
		R	3.9	3.7	3.1	4.3	2.4	2.2	2.1	3
	BR	SHD	7	7	6	7	5	5	5	6
Explicit knowledge	MR	SHD	7.4	6.8	6.4	8.6	5.8	5.4	5	6.7
		D	3.1	2.8	2.7	3.1	2.2	2.2	2.1	2.1
		A	0.9	0.9	0.9	1.2	1.3	1.2	1.2	1.6
		R	3.4	3.1	2.8	4.3	2.3	2	1.7	3
	BR	SHD	5	4	3	7	3	4	3	6
EV knowledge	MR	SHD	6.7	6	5.9	8.6	5.4	5.2	4.7	6.7
		D	2.7	2.9	2.8	3.1	2.1	2.4	2.1	2.1
		A	0.8	0.9	0.9	1.2	1.2	1	1	1.6
		R	3.2	2.2	2.2	4.3	2.1	1.8	1.6	3
	BR	SHD	3	4	3	7	4	3	2	6

We calculate the correct edges of learned networks. Figure 7 shows the results of this experiment.

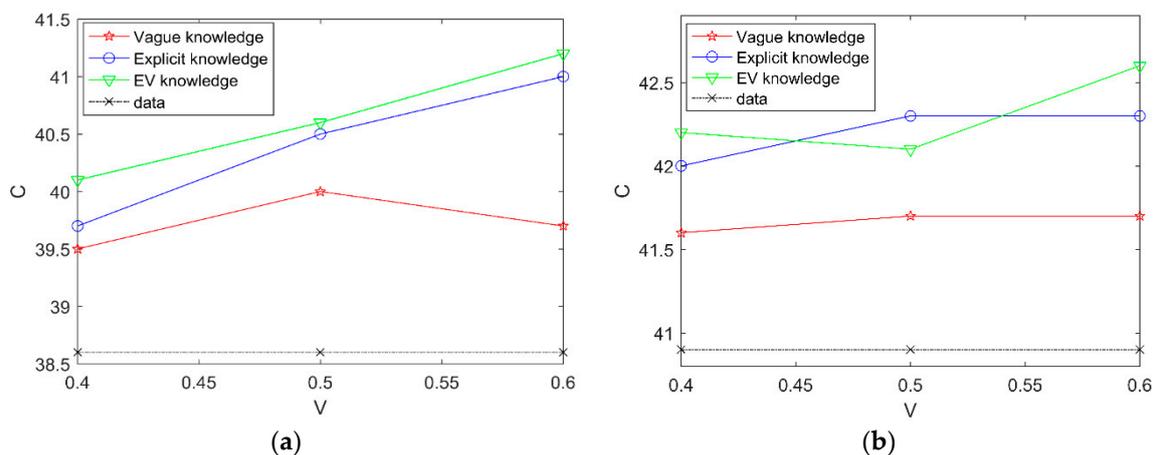


Figure 7. The correct edges of learned networks with different datasets in the second Results and Discussion case; (a) the correct edges of learned networks with 2000 samples in the second Results and Discussion case; (b) the correct edges of learned networks with 5000 samples in the second Results and Discussion case.

In Figure 7, the horizontal coordinate represents the percentage of experts' knowledge and the vertical coordinate represents average SHD. Compared to learning with three cases of experts' knowledge and learning with data alone in different datasets, the experimental results indicate that the proposed the EVBIC scoring function using different types of experts' knowledge in the second

algorithm is effective and learning with explicit knowledge and vague knowledge is better than learning with explicit knowledge alone.

4.2.3. Results and Discussion Using the Different Types of Experts' Knowledge in the Hybrid Algorithm

In order to ensure the robustness of the results, we calculate the A, D, I and SHD of learned networks for the Asia network and the Alarm network. Tables 6 and 7 show the results of this experiment.

Table 6. A, D, I and SHD values of learned network for the Alarm network in the third Results and Discussion case.

The Case of Knowledge	Measures of Performance	2000				5000				
		0.4	0.5	0.6	Data	0.4	0.5	0.6	Data	
Vague knowledge	MR	SHD	7.3	6.8	6.1	8.6	4.8	4.6	3.5	6.7
		D	3	2.7	2.1	3.1	1.7	1.7	1.4	2.1
		A	1	0.8	0.5	1.2	1	0.9	0.8	1.6
	BR	R	3.3	3.3	3.5	4.3	2.1	2	1.3	3
	BR	SHD	6	3	3	7	1	2	1	6
Explicit knowledge	MR	SHD	6.2	5.2	4.7	8.6	4.5	3.9	3.4	6.7
		D	2.6	2.4	2.4	3.1	2	1.7	1.5	2.1
		A	0.8	0.5	0.4	1.2	0.8	0.9	0.9	1.6
	BR	R	2.8	2.3	1.9	4.3	1.7	1.3	1	3
	BR	SHD	3	3	3	7	1	1	1	6
EV knowledge	MR	SHD	6	5	4	8.6	4.1	3.4	3.3	6.7
		D	2.6	2.4	2.1	3.1	1.9	1.7	1.5	2.1
		A	0.7	0.4	0.1	1.2	1	0.8	0.9	1.6
	BR	R	2.7	2.2	1.8	4.3	1.2	0.9	0.9	3
	BR	SHD	3	3	3	7	1	1	0	6

Table 7. A, D, I and SHD values of learned network for the Asia network in the third Results and Discussion case.

The Case of Knowledge	Measures of Performance	2000				5000				
		0.4	0.5	0.6	Data	0.4	0.5	0.6	Data	
Vague knowledge	MR	SHD	2.1	1.9	1.6	3.2	0.9	0.6	0.4	1.7
		D	1.2	1.1	0.9	1.6	0.6	0.5	0.2	0.5
		A	0.8	0.5	0.3	1.2	0.3	0.1	0.1	0.7
	BR	R	0.1	0.3	0.4	0.4	0.1	0	0.1	0.5
	BR	SHD	1	1	0	2	0	0	0	0
Explicit knowledge	MR	SHD	1.6	1.3	1	3.2	0.7	0.4	0.2	1.7
		D	0.9	1.0	0.6	1.6	0.4	0.2	0.1	0.5
		A	0.5	0.2	0.2	1.2	0.1	0.1	0	0.7
	BR	R	0.2	0.1	0.2	0.4	0.2	0.1	0.1	0.5
	BR	SHD	1	0	0	2	0	0	0	0
EV knowledge	MR	SHD	1.2	0.9	0.6	3.2	0.4	0.2	0	1.7
		D	0.8	0.5	0.4	1.6	0.3	0.1	0	0.5
		A	0.3	0.2	0.1	1.2	0.1	0	0	0.7
	BR	R	0.1	0.2	0.1	0.4	0	0.1	0	0.5
	BR	SHD	1	0	0	2	0	0	0	0

The smaller A, D, I and SHD are, the better the structure learning is. The SHD shows the total difference between the learned and original networks and A, D, I represent the details of different kinds of difference between the learned and original networks. From Table 6, we can easily find that the results of learning with each case of experts' knowledge in the Alarm network are better than the results of learning without experts' knowledge in the Alarm network. It is shown that the hybrid algorithm using different types of experts' knowledge are effective. Comparing with three cases of experts' knowledge, we can easily see that learning with explicit knowledge and vague knowledge is better than learning with explicit knowledge alone. From Table 7, according to the Asia network,

we can also find that the hybrid algorithm using different types of experts' knowledge are effective. It is thus proved that the result is robust.

From Table 7, according to the Asia network, we can also find that the hybrid algorithm using different types of experts' knowledge are effective. It is thus proved that the result is robust.

In order to ensure the robustness of the results, we use the Asia network and Alarm network as our standard networks. Compared to learning with three cases of experts' knowledge and learning with data alone by calculating the single SHD, the experimental results can directly the effect of structure learning. Because the results of using the samples = 5000 is the same as the samples = 2000 for the Alarm network and the results of using the samples = 500 is the same as the samples = 2000 for the Asia network, we use samples = 5000 for the Alarm network and samples = 500 for the Asia network as examples. Figures 8 and 9 show the results of this experiments.

The red circles mark the best result. The smaller SHD is, the better the structure learning is. From Figures 8 and 9, we can easily see that the best results appear in learning BN structure with knowledge and the learned BN structure is more similar to the original network structure through using knowledge than through using data alone. From Figure 8, according to the Alarm network, the results indicate that our proposed method using the experts' knowledge in the hybrid algorithm is effective. From Figure 9, according to the Asia network, we can also get same result. It is proved that the results are robust.

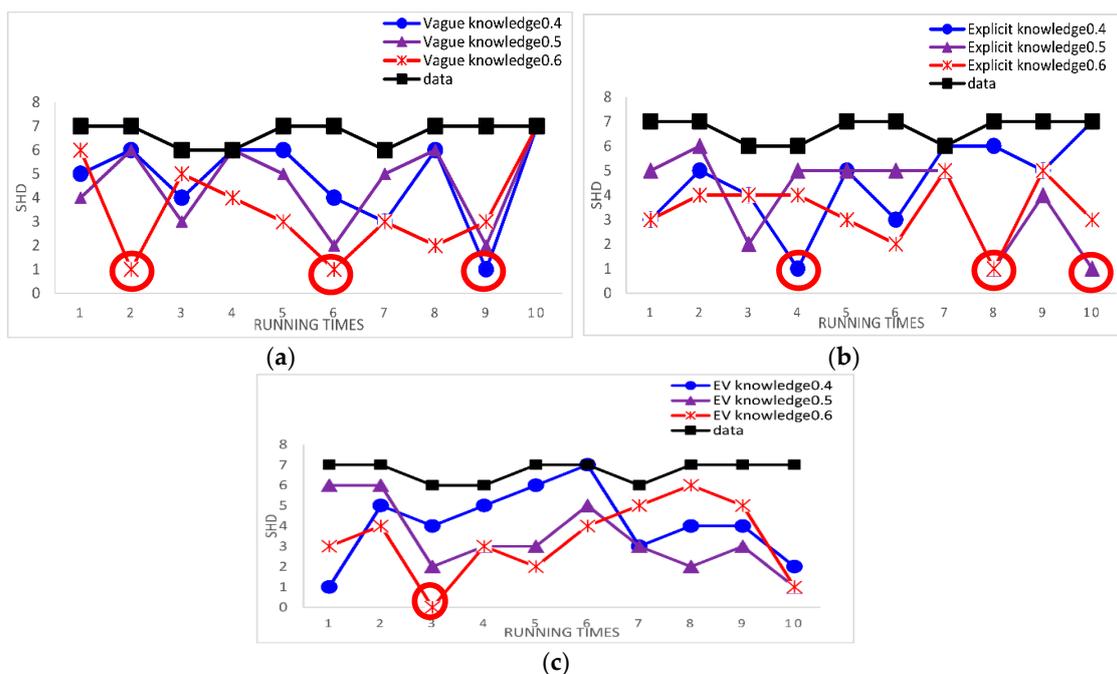


Figure 8. The single SHD of learned networks using different case of experts' knowledge and using data alone for the Alarm network in the third Results and Discussion case. (a) The single SHD of learned networks using vague knowledge and using data alone in 5000 samples; (b) the single SHD of learned networks using explicit knowledge and using data alone in 5000 samples; (c) the single SHD of learned networks using EV knowledge and using data alone in 5000 samples.

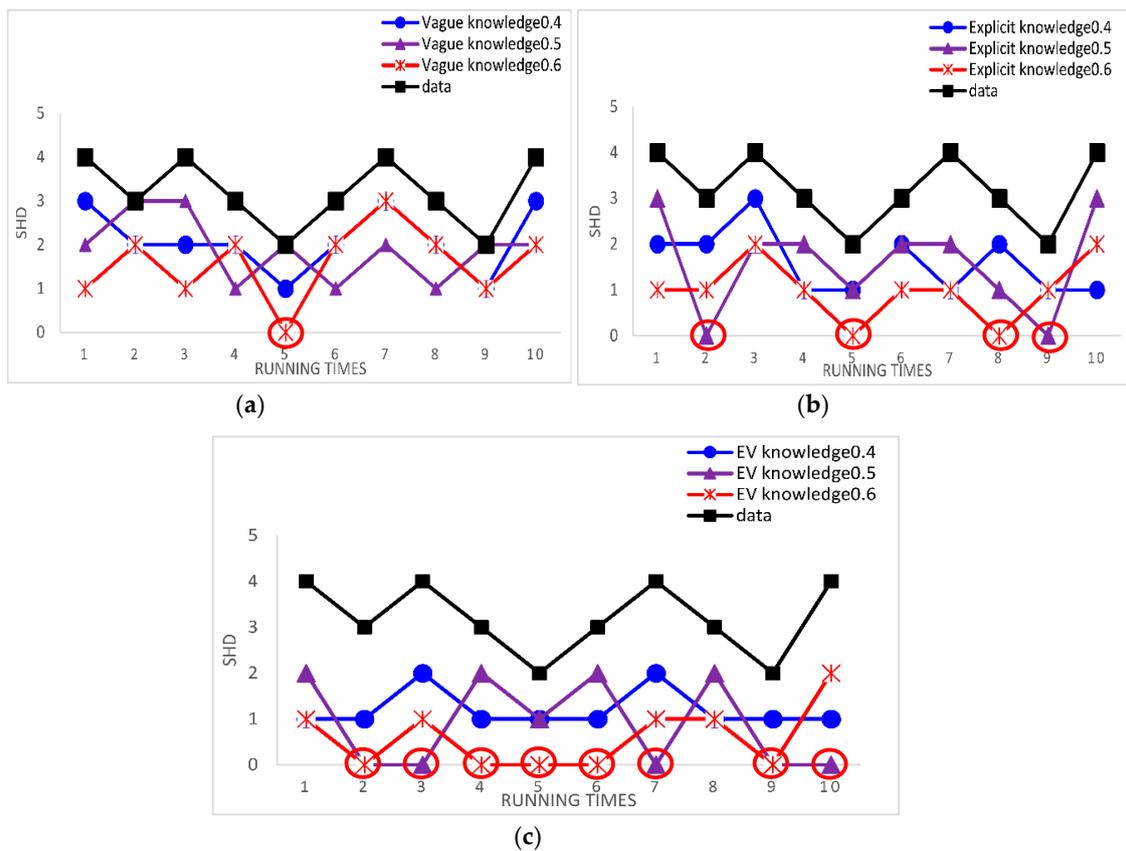


Figure 9. The single SHD of learned networks using different case of experts’ knowledge and using data alone for the Asia network in the third Results and Discussion case. (a) The single SHD of learned networks using Vague knowledge and using data alone in 500 samples; (b) the single SHD of learned networks using Explicit knowledge and using data alone in 500 samples; (c) the single SHD of learned networks using EV knowledge and using data alone in 500 samples.

Compared learning with experts’ knowledge in different stage of the hybrid algorithm and learning with data alone by calculating average SHD, the experimental results are shown in Figure 10, where we can see that learning with experts’ knowledge can get a better structure than learning with data alone, which proved that our proposed method using the experts’ knowledge in the hybrid algorithm is effective. Moreover, learning with experts’ knowledge in two stage of the hybrid algorithm can get a better structure than learning with experts’ knowledge in each stage of the hybrid algorithm. It is indicated that using experts’ knowledge in each stage of the hybrid algorithm is not redundant. It is shown that our proposed method using experts’ knowledge is effective.

To expose the effectiveness of proposed method, we compare our method with the method in [17] by calculating average SHD in the Alarm network and Asia network. the experimental results are shown Figures 11 and 12.

The smaller SHD values are, the better the structure learning is. From Figures 11 and 12, we can easily find that the learned network using our proposed method is better than the learned network using the method in [17] in the Asia network and Alarm network. The results show that our proposed method in this paper has better learning performance and better learning effect.

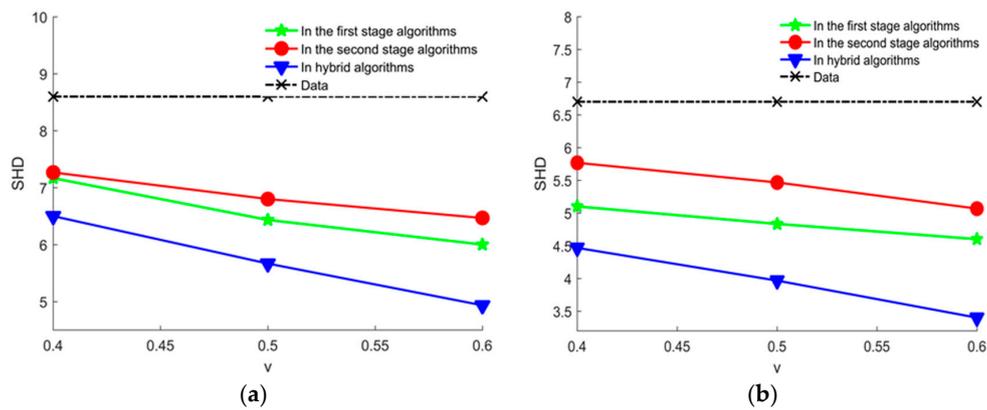


Figure 10. The average SHD values of learned networks using experts’ knowledge in different stage of the hybrid algorithm and learning with data alone in different datasets. (a) Average SHD values of learned networks using experts’ knowledge in different stage of the hybrid algorithm and learning with data alone in 2000 samples; (b) the average SHD values of learned networks using experts’ knowledge in different stage of the hybrid algorithm and learning with data alone in 5000 samples.

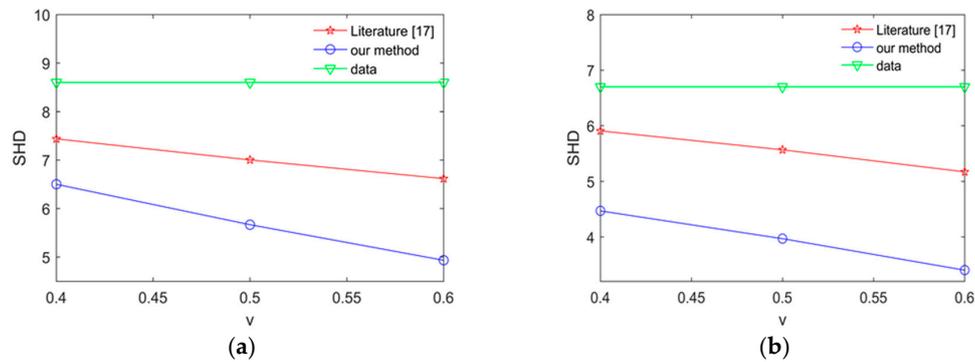


Figure 11. The average SHD values for the Alarm network using different methods. (a) The average SHD values for the Alarm network using different method in 2000 samples; (b) the average SHD values for the Alarm network using different method in 5000 samples.

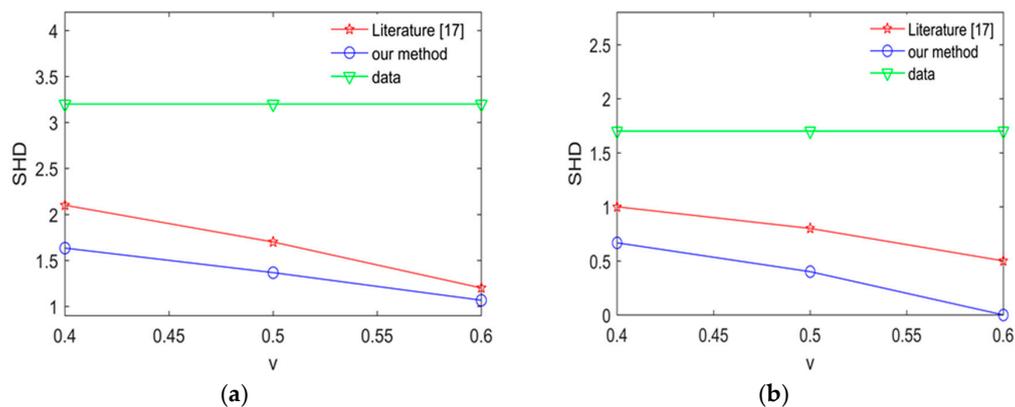


Figure 12. The average SHD values for the Asia network using different method. (a) The average SHD values for the Asia network using different method in 2000 samples; (b) the average SHD values for Asia network using different method in 5000 samples.

5. Conclusions

We introduce a new method of using explicit knowledge and vague knowledge based on a hybrid structure learning algorithm for Bayesian network. The simulation experiment results show that our method achieves a higher BIC score and a smaller structure difference than previous algorithms by taking two types of experts' knowledge into account. It is acknowledged that the introduction of explicit knowledge can improve the accuracy of Bayesian network structure based on data learning algorithm. Furthermore, we point out that the experts can give not only explicit knowledge but also vague knowledge in the real world and it is effective to improve the accuracy of Bayesian network structure by using vague knowledge. The main novelty of our proposed algorithm is that we consider these two types of experts' knowledge in the process of learning Bayesian network structure but not explicit knowledge alone. Our proposed algorithm can also solve the problems of expert level difference and opinion conflict on the same group of node pairs.

Author Contributions: Writing-Original Draft Preparation, H.G.; Writing-Review & Editing, H.L.

Acknowledgments: This study was supported by the National Natural Science Foundation of China (61533007), the National Key R&D Program of China (2017YFB0304205).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jensen, F.V. *An Introduction to Bayesian Networks*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 1–5.
2. Nikovski, D. Constructing Bayesian networks for medical diagnosis from vague and partially correct statistics. *IEEE Trans. Knowl. Data Eng.* **2000**, *12*, 509–516. [[CrossRef](#)]
3. Wolbrecht, E.; D'Ambrosio, B.; Paasch, R.; Kirby, D. Monitoring and diagnosis of a multistage manufacturing process using Bayesian networks. *Ai Edam* **1998**, *14*, 53–67. [[CrossRef](#)]
4. Hosseini, S.; Barker, K. A Bayesian network model for resilience-based supplier selection. *Int. J. Prod. Econ.* **2016**, *180*, 68–87. [[CrossRef](#)]
5. Hosseini, S.; Khaled, A.A.; Sarder, M.D. A general framework for assessing system resilience using Bayesian networks: A case study of sulfuric acid manufacturer. *J. Manuf. Syst.* **2016**, *41*, 211–227. [[CrossRef](#)]
6. Hosseini, S.; Barker, K. Modeling infrastructure resilience using Bayesian networks: A case study of inland waterway ports. *Comput. Ind. Eng.* **2016**, *93*, 252–266. [[CrossRef](#)]
7. Bui, A.T.; Jun, C.H. Learning Bayesian network structure using Markov blanket decomposition. *Pattern Recognit. Lett.* **2012**, *33*, 2134–2140. [[CrossRef](#)]
8. Dünder, E.; Cengiz, M.A.; Koç, H. Investigation of the impacts of constraint-based algorithms to the quality of Bayesian network structure in hybrid algorithms for medical studies. *J. Adv. Sci. Res.* **2014**, *5*, 8–12.
9. Ziegler, V. Approximation algorithms for restricted Bayesian network structures. *Inf. Process. Lett.* **2008**, *108*, 60–63. [[CrossRef](#)]
10. Campos, L.M.D.; Fernández-Luna, J.M.; Gámez, J.A.; Puerta, J.M. Ant colony optimization for learning Bayesian networks. *Int. J. Approx. Reason.* **2002**, *31*, 291–311. [[CrossRef](#)]
11. Tsamardinos, I.; Brown, L.E.; Aliferis, C.F. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* **2006**, *65*, 31–78. [[CrossRef](#)]
12. Perrier, E. Finding optimal Bayesian network given a super-structure. *J. Mach. Learn. Res.* **2012**, *9*, 2251–2286.
13. Acid, S.; Campos, L.M.D. A hybrid methodology for learning belief networks: Benedict. *Int. J. Approx. Reason.* **2001**, *27*, 235–262. [[CrossRef](#)]
14. Li, S.; Zhang, J.; Sun, B.; Lei, J. An incremental structure learning approach for Bayesian network. In Proceedings of the 26th Chinese Control and Decision Conference, Changsha, China, 31 May–2 June 2014; pp. 4817–4822.
15. Chickering, D.M.; Dan, G.; Heckerman, D. Learning Bayesian networks is NP-hard. *Networks* **1994**, *112*, 121–130.
16. Xu, J.-G.; Zhao, Y.; Chen, J.; Han, C. A structure learning algorithm for Bayesian network using prior knowledge. *J. Comput. Sci. Technol.* **2015**, *30*, 713–724. [[CrossRef](#)]

17. Amirkhani, H.; Rahmati, M.; Lucas, P.; Hommersom, A. Exploiting experts' knowledge for structure learning of Bayesian networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 2154–2170. [[CrossRef](#)] [[PubMed](#)]
18. Heckerman, D.; Dan, G.; Chickering, D.M. Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.* **1995**, *20*, 197–243. [[CrossRef](#)]
19. Chen, J.; Jia, H.; Huang, Y.; Liu, D. Learning the structure of dynamic Bayesian network with domain knowledge. In Proceedings of the International Conference on Machine Learning and Cybernetics, Xi'an, China, 15–17 July 2012; pp. 372–375.
20. Murphy, K.P. An Introduction to Graphical Models. Available online: http://www2.denizyuret.com/ref/murphy/intro_gm.pdf (accessed on 19 August 2018).
21. Zhang, Y.; Zhang, W.; Xie, Y. Improved heuristic equivalent search algorithm based on maximal information coefficient for bayesian network structure learning. *Neurocomputing* **2013**, *117*, 186–195. [[CrossRef](#)]
22. Robinson, R.W. *Counting Unlabeled Acyclic Digraphs*; Springer: Berlin/Heidelberg, Germany, 1977; pp. 28–43.
23. Mirjalili, S.; Lewis, A. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [[CrossRef](#)]
24. Campos, L.M.D.; Castellano, J.G. Bayesian Network Learning Algorithms Using Structural Restrictions. *Int. J. Approx. Reason.* **2007**, *45*, 233–254. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).