HAL
open science

Learning Bayesian Network Structure from Incomplete Data Without Any Assumption

Céline Fiot, G. A. Putri Saptawati, Anne Laurent, Maguelonne Teisseire

To cite this version:

Céline Fiot, G. A. Putri Saptawati, Anne Laurent, Maguelonne Teisseire. Learning Bayesian Network Structure from Incomplete Data Without Any Assumption. DASFAA: Database Systems for Advanced Applications, Mar 2008, New Delhi, India. pp.408-423, 10.1007/978-3-540-78568-2_30 . lirmm-00273888

HAL Id: lirmm-00273888

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00273888v1

Submitted on 9 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning Bayesian Network Structure from Incomplete Data without any Assumption

Céline Fiot[1], G.A. Putri Saptawati[2],
Anne Laurent[1], and Maguelonne Teisseire[1]

[1] LIRMM - Univ. Montpellier II, CNRS
161 rue Ada, 34392 Montpellier, France
{fiot, laurent, teisseire}@lirmm.fr

[2] Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132, Indonesia
putri@informatika.org

Abstract. Since most real-life data contain missing values, reasoning and learning with incomplete data has become crucial in data mining and machine learning. In particular, bayesian networks are one machine learning technique that allow for reasoning with incomplete data, but training such networks with incomplete data may be a difficult task. Many methods were thus proposed to learn bayesian network structure with incomplete data, based on multiple structure generation and scoring of their adequacy to the dataset. However this kind of approaches may be time-consuming. Therefore we propose an efficient dependency analysis approach that uses a redefinition of probability calculation to take incomplete records into account while learning BN structure, without generating multiple possibilities. Some experiments on well-known benchmarks are described to show the validity of our proposal.

1 Introduction

Graphical models [1] are tools combining two different areas: graph theory and probability theory. They are often used to illustrate and work with conditional independencies and probabilistic relationships among variables in a given problem. Among graphical models, bayesian networks are often involved in tasks requiring to reason under uncertainty. In particular, many approaches allow for reasoning with incomplete data. But the task of training bayesian networks with incomplete datasets is more complex.

In fact, there are two classes of methods for building the graphical structure of bayesian networks from complete datasets. Scoring-based algorithms consist in generating many likely structures and then scoring them using a fitness function measuring how well each possible graph fits the data. On the other hand, constraint-based algorithms – also called dependency analysis approaches – build the graph directly from the data thanks to probability calculations and conditional independency tests.

Some approaches tackle the problem of missing values either by deleting observations with missing values or using ad-hoc techniques to impute missing information. Such procedures may however lead to biased results, and in case of imputing a single value for unassigned attributes, to an overconfidence in the results of the analysis. Some specific algorithms have also been shown to be successful for learning bayesian network structure from complete data, and learning parameters for a fixed network. Other scoring-based algorithm have finally been developed using an estimation of the missing observations on the basis of available information and data distribution. However this may be time-consuming and resource-demanding.

Since very few methods are capable of using incomplete cases as a base to determine the structure of a bayesian network by a constraint-based approach, we propose in this paper an efficient dependency analysis approach to handle incomplete records while learning BN structure. Our proposal consists in a re-definition of probability calculation that allows for information incompleteness without missing value imputation. Then we adapted the efficient Three Phase Dependency Analysis algorithm proposed by [2] to make it use our new probability definitions, while computing conditional independency tests. Some experiments on classical benchmarks are described to show the validity of our proposal for generating bayesian network structure underlying incomplete datasets.

This paper is organized as follows: in Section 2, we introduce the definition and principles used in the context of bayesian network learning; then in Section 3, we detail the basis of our approach and our new definitions for probability calculation. Finally, before concluding in Section 5, some results of experiments are developed by Section 4.

2 Bayesian Network and Incomplete Data

Regarding a data set, a bayesian network gives both a qualitative and quantitative description of the dependencies existing between data attributes. First, these dependencies are visually described by a directed acyclic graph (DAG). In this graph, each vertex, or node, corresponds to an attribute in the database and directed edges between nodes show the dependencies between related attributes. Then, each node is associated with a conditional probability table, which gives, for each value of the node attribute, its probability considering the value of the attribute parent nodes.

2.1 Directed Acyclic Graph (DAG)

A *graph*, or *undirected graph*, can be defined as a set of *nodes*, also called *vertices*, and a set of *edges*, also called *arcs*, each being a pair of nodes. If the two vertices within each edge are ordered, then the edges have a direction assigned to them; this is called a *directed graph*. A *chain* is a series of nodes where each successive node in the chain is connected to the previous node by an edge. A *path* is a chain with the further constraint for directed graphs that each connecting edge in the

chain has a directionality going in the same direction as the chain. A *cycle* is a path that starts and ends at the same node. A *directed acyclic graph*, or *DAG*, is a directed graph that has no cycles.
The terms *parent* and *child* define the relationship between two vertices connected by a directed edge from the parent to the child. Two vertices are said to be *adjacent* when they are connected by an undirected edge.

2.2 Bayesian Network

A bayesian network is a specific graphical model that is a concise representation of the joint probability distribution for a large set of attributes in a database. Each attribute can be considered as a random variable associated with several values. Then, for a set of variables $\mathcal{V}$, a bayesian network consists of a *directed acyclic graph* that encodes a set of conditional dependence and independence assertions about variables in $\mathcal{V}$, and a set of local probability distributions associated with each variable. Together, these components define the joint probability distribution for $\mathcal{V}$.

In [3], Pearl defines a bayesian network as a triplet $[\mathcal{V}, G, P(V_i|P_a(V_i))]$, where:

- $\mathcal{V} = \{V_1, ..., V_n\}$ is the *set of random discrete variables*;
- G is a *directed acyclic graph* whose nodes represent variables V_i, and whose arcs encode the *conditional dependencies* between the variables;
- $P(V_i|p_a(V_i))$ describes the conditional probability distribution of each variable V_i considering its immediate parents $pa(V_i)$ in the graph G.

The edges in the bayesian network encode a particular factorization of the joint distribution. In general, the joint probability function for any bayesian network representing the set of nodes $\mathcal{V}$ is given by

$$P(\mathcal{V}) = \prod_{i=1}^{n} P(V_i|pa(V_i))$$

This means that the joint probability of all of the variables is the product of the probabilities of each variable given its parents' values. Then, the graph describes these dependencies. For any given edge between variables V_i and V_j, if there is a causal relationship between variables, the edge will be directional, leading from the cause variable to the effect variable. If there is just a correlation between the two variables, the edge will be undirected [4]. Two variables that are conditionaly independent have no direct impact on each other's values. However, any path through intermediary variables that separates two conditionally independent variables shows how these two conditionally independent variables affect each other.

2.3 Learning Bayesian Networks

As a bayesian network is constituted of one qualitative component, the DAG, and one quantitative component, the conditional probability distribution, learning bayesian network consists in two tasks. First the structure describing the

dependencies is designed, then the conditional probabilities of each node are calculated. Two approaches are generally used to learn the structure. The first one is based on scoring an *a priori* designed structure. The second one uses constraints and conditional independence tests to build the graph.

Scoring-based approaches [5,6,7,8] select the DAG that best fits the data among several ones a priori designed. The objective of learning is then to evaluate each previously designed structure regarding the dataset. These methods require to specify scoring functions that are used to evaluate how well each network matches the training data. In the approaches based on model selection, some criterion is used to measure the degree to which a network structure (equivalence class) fits prior knowledge and data. A search algorithm is then used to find an equivalence class that receives a high score by this criterion.

Constraint-based algorithms, also called dependency analysis algorithms, build the DAG structure by identifying the conditional independence relationships among the variables [9,10,11,12]. These methods are based on the causal sufficiency hypothesis: *for every pair of measured variables in the training data, all their common parents are also measured*. Thus, the graph is built thanks to the set of data, without external knowledge. Vertices of the DAG are built from the variables within the dataset and the edges are built from the observed dependencies between variables within the data.

Once the structure has been designed, each node of the DAG is associated with a table of conditional probabilities that give for each value of each variable the path to follow in the DAG.
Depending on the problem that is defined, either the topology or the probability distributions or both may be pre-defined by hand or may be learned from the data.

In this paper, we will consider that the structure is unknown but that all the variables can be identified (i. e. there is no *hidden variables* [13]). Within a context where some variables are randomly unassigned, we tackle the problem of learning the structure of a bayesian network given data, using an approach based on the information theory.

2.4 Handling Missing Values

In data mining and machine learning, missing value handling is a significant problem as most real-life data contain unassigned variables. One advantage of belief networks is that they allow reasoning with incomplete data [13]. Many inference algorithms can indeed be used to calculate the probability of any variable that has not been measured conditionally to the values of measured variables. But complete data are often required for training such networks.

As described in the previous section, there are two different problems related to the presence of missing values while learning bayesian network. One consists in the evaluation of the probability parameters despite unassigned variables, the second consists in assessing the dependencies and learning the graph structure in spite of incomplete observations.

Some approaches tackle the problem of missing values either by deleting observations with missing values or using ad-hoc techniques to impute missing information. Such procedures may however lead to biased results, and in case of imputing a single value for unassigned attributes, to an overconfidence in the results of the analysis.

Therefore specific algorithms have been developed and several methods have been shown to be successful for learning both network structure and parameters from complete data, and learning parameters for a fixed network [14]. But very few methods are capable of using incomplete cases as a base to determine the structure of a bayesian network by a constraint-based approach.

Most of techniques that determine the bayesian network structure from incomplete data are indeed based on model scoring and selection: these proposals use an a priori known structure and compare it to the observed distribution. Well-known methods typically involve the use of the EM algorithm [15] or Markov Chain Monte-Carlo methods, such as Gibbs sampling [16]. The basic strategy underlying these methods is based on the *Missing Information Principle* [17]: fill in the missing observations on the basis of the available information.

Thus [18,19,20] propose different algorithms based on extensions of the expectation-maximization algorithm for model selection problems. [21] and [22] describe approaches based on stochastic search and evolutionary algorithm that approximates a maximum likelihood approach to score the network by evolving samples of incomplete data. Unfortunately, these processes are usually highly resource demanding, their convergence rates may be slow, and their execution time heavily depends on the number of missing values.

Therefore [23] uses an entropy maximization procedure to incorporate information regarding the nature of the missing data mechanism and thus considerably saving in computation time when compared to Gibbs sampling.

Other scoring-based approaches use estimation of missing data for both parameter and structure learning. [24,25] introduces a deterministic method to estimate the conditional probabilities defining the dependencies in a bayesian network which does not rely on the Missing Information Principle and proposes the *Bound and Collapse* algorithm for parameter estimation and model selection from incomplete data. However this algorithm also relies on an assumed pattern of missing data that may be either provided by an external source of information or may be estimated from the available information under the assumption that data are missing at random. This approach is extended in [26] to learn the graphical structure of a bayesian network from a possibly incomplete database, using estimation of missing data.

More recently, [27] describes an imputation-based approach for model learning from incomplete data, where possible completions of the data are scored together with the observed part of the data. [28,29] describes an algorithm based on extended evolutionary programming method. It uses fitness function based on expectation, which converts incomplete data to complete data. [30] introduces

an approach for assessing the predictive distribution of missing values that is then combined to any learning algorithm.

2.5 Objectives

The main disadvantage of scoring-based approaches is that they rely on determining among numerous structures the one that best fit the data. So this kind of approaches requires *a priori* expert knowledge to design few structures to be tested or to generate every possible structures from the data. This processes are thus highly resource demanding and in the case of incomplete data, the runtime may become very high.

Therefore in this paper we propose a constraint-based approach to learn bayesian network from a randomly incomplete dataset without assessing or deleting missing values nor generating several possible structures. We use observed data without requiring any external information or estimation of missing value distribution.

Our method is based on a redefinition of the probability functions taking into account that some variable values are unknown. We developed our algorithm for learning bayesian network structure by adapting the efficient Three Phase Dependency Analysis (TPDA) algorithm proposed in [2] to make it use our own probability definitions. Finally, we ran several experiments to show the feasibility, validity and robustness of our approach.

In the following section, we introduce the principles we based our approach on. Then, we detail our new definitions and prove that they hold all the conditions required to define a probability measure. Last, we describe the overall learning algorithm and run a brief example. In Section 4, we present the results of our experiments.

3 TPDA for Incomplete Databases

Our approach is based on the same principles as the ones used by the RAR algorithm [31] for association rule mining [32] in incomplete databases. The data formalism of association rules is indeed quite similar to the one of bayesian network: the dataset is a relational table consisting in records in which values are associated with attributes, that correspond to random variables in the context of bayesian networks.

The main idea of our approach is based on the RAR method. It consists in disabling incomplete elements, within our context, incomplete records. As the RAR algorithm for association rules mining, we will only regard complete records to compute the conditional probabilities. In other words, when an incomplete record is scanned, only filled-in attributes will be considered for probability calculation. Thus each conditional probability will be computed on a partial database, but the whole dataset will be used to find the whole set of dependencies.

3.1 Overall Principle

The RAR algorithm (Robust Association Rules), proposed by [31], allows the user to consider incomplete data while association rule mining within incomplete relational databases, thanks to partial and temporary omission of such incomplete records. The main idea consists in taking only filled-in attributes in incomplete records into account. The whole database is not used to discover each rule but the whole set of rules.

This technique is based on the valid database concept, which is a complete dataset for a given itemset, i.e. a set of attributes or variables. The remaining part of the database is temporary ignored. In order to consider this dataset partitioning, definitions of support (percentage of records in database that include the rule items) and confidence (probability for a record to contain the right part of the rule knowing it contains the left part) were reformulated.

Learning the structure of a bayesian network by a constraint-based approach such as TPDA algorithm requires to compute conditional probabilities and probabilistic conditional independence tests. We here apply the formalism of the RAR algorithm to define a new probability measure. This measure will then be used by our implementation of the TPDA algorithm to run the conditional independence tests and thus to build the DAG structure.

So, let us consider a set of random variables V each associated with one of their values v, the set $\mathcal{R}$ of records r in the database DB can be divided into three disjoint subsets (Figure 1). The set of records filled in with the corresponding value v_i for each variable V_i of V is denoted by $\mathcal{R}_V$. The set of records filled in with at least one value different from the set v is denoted by $\mathcal{R}_{\overline{V}}$. And the set of records for which at least one value v is unfilled, i. e. is missing and we do not know if $r(V_i)=v_i$ or not, is denoted by $\mathcal{R}^*_V$.

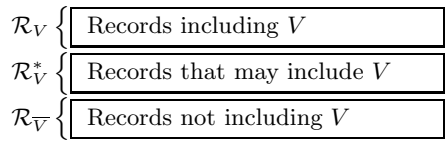

Fig. 1. Partition of the database depending on V inclusion.

	X_1	X_2	X_3	X_4	X_5
R_1	?	y	y	n	n
R_2	y	n	n	?	y
R_3	y	y	n	y	n
R_4	y	n	n	y	n
R_5	n	?	y	y	y

Fig. 2. An incomplete dataset

For each set of variables V, only the subsets $\mathcal{R}_{\overline{V}} \cup \mathcal{R}_V$ will be kept to determine the conditional probabilities on V. This subset represents the *valid database* for V. Incomplete records are *disabled* for V.

Definition 1. *A* valid database *is a database only containing complete records for a given set of random variables, i.e. each value of each record in the data corresponds to an identified values v of $\mathcal{D}om(V)$.*

Definition 2. *A record is* disabled *for an instanciation of a set of variables V if it is incomplete for V (i.e. we cannot decide whether it includes V or not). The set of records disabled for a set V is denoted by $Dis(V)$.*

For instance, considering the dataset described by Fig. 2, the valid database for X_1 is composed of records R_2 to R_5 and $Dis(X_1) = \{R_1\}$. The valid database for $V = \{X_1, X_4\}$ is $\{R_3, R_4, R_5\}$, and $Dis(V) = \{R_1, R_2\}$.

Building a valid database leans on temporary disabling records that contain missing values for variables in the set of random variables. This implies a redefinition of the probability calculation to consider the database partial deactivation.

3.2 Redefining Calculation of Probabilities

The probability definition is modified in order to consider the valid database concept, and thus that only one part of the dataset is used for each probability calculation.

Definition 3. *The* probability *of an event v_i is the appearance rate of this event among the records that can include it. It is defined as the ratio of the number of records r such that $r(V_i) = v_i$ by the number of records that are filled in for V_i (complete records for V_i). It is given by:*

$$P(V_i = v_i) = \frac{card\left(\{r \in \mathcal{R} | r(V_i) = v_i\}\right)}{card(\mathcal{R}) - card(Dis(V_i))} \tag{1}$$

Considering a set of random variables $V = \{V_1, ..., V_n\} \subseteq \mathcal{V}$ and a joint probability function P defined on $\mathcal{V}$, the probability of a joint event $P(v_1, ..., v_n)$ is computed considering the set of records that are complete for all the variables $V_1, ...V_n$. Then the previous formula can be expressed as follows:

$$P_{V_1,...,V_n}(v_1, ..., v_n) = \frac{card\left(\bigcap_{i \in [1,n]} \{r \in \mathcal{R} | r(V_i) = v_i\}\right)}{card(\mathcal{R}) - card(Dis(V_1, ..., V_n))} \tag{2}$$

For instance on Fig. 2, to compute $P(X_1 = y)$, we find $Dis(X_1) = \{R_1\}$, then $P(X_1 = y) = card(\{R_2, R_3, R_4\})/5 - card(Dis(X_1)) = 3/(5-1) = 0.75$. If we compute $P(X_1 = y, X_2 = n)$, then $Dis(X_1, X_2) = \{R_1, R_5\}$ and $P(X_1 = y, X_2 = n) = card(\{R_2, R_4\})/(5 - card(Dis(X_1, X_2))) = 2/(5-2) = 0.67$.

Proposition 1. *Given a random variable V_i of values v_i in domain $\mathcal{D}(V_i)$, the redefinition of the probability P calculation defines a joint probability function over the variable set $\mathcal{V}$.*

Proof. A probability function P must satisfy the following properties:

1. for every event $A \in \mathbf{A}$, $0 \leq P(A) \leq 1$;
2. for the impossible event $\varnothing$ and the certain event Ω, $P(\varnothing) = 0$ and $P(\Omega) = 1$;
3. if the events $A_i \in \mathbf{A}$ are finite or countably many mutually exclusive events ($A_i A_k = \varnothing$ for $i \neq k$), then $P(\bigcup_{i=1}^{n} A_i = \Sigma_{i=1}^{n} P(A_i)$

We will denote $r(V_i)$ the value of attribute/variable V_i in the record r and $card(S)$ will denote the cardinality of a subset S of records.

1. Considering the database partitioning, $\{r \in \mathcal{R} | r(V_i) = v\} \subseteq \mathcal{R} \backslash Dis(V_i)$, which implies that $card(\{r \in \mathcal{R} | r(V_i) = v\}) \leq card(\mathcal{R}) - card(Dis(V_i))$. Then, as a cardinality is necessarily greater than or equal to zero and assuming that at least one record in the database is complete for V_i, we obtain

$$0 \leq \frac{card(\{r \in \mathcal{R} | r(V_i) = v\})}{card(\mathcal{R}) - card(Dis(V_i))} \leq 1 \;\Rightarrow\; \forall v, 0 \leq P_{V_i}(v) \leq 1$$

2. A record necessarily contains a value, missing or not, for a variable V_i then $card(\{r \in \mathcal{R} | r(V_i) = \varnothing\}) = 0$ and $P_{V_i}(\varnothing) = 0$. Then we show that $P_{V_i}(\cup_{v \in \mathcal{D}(V_i)} V_i = v) = 1$:

$$\cup_{v \in \mathcal{D}(V_i)} \{r \in \mathcal{R} | r(V_i) = v\} = \mathcal{R} \backslash Dis(V_i) \Rightarrow \frac{card(\cup_{v \in \mathcal{D}(V_i)} \{r \in \mathcal{R} | r(V_i) = v\})}{card(\mathcal{R}) - card(Dis(V_i))} = 1$$
$$\Rightarrow P_{V_i}(\cup_{v \in \mathcal{D}(V_i)} V_i = v) = 1$$

3. $\forall A_j \in V_i | \forall j \neq k, A_j \cap A_k = \varnothing, P_{V_i}(\cup_j A_j) = \sum_j P_{V_i}(A_j)$.

   Within our context, such an event A_j corresponds to a set of values v for a random variable V_i. In other words, it can be expressed by the formula

$$\forall A_j, \exists D_j \subseteq \mathcal{D}(V_i) | A_j = \cup_{v \in D_j} \{r \in \mathcal{R} | r(V_i) = v\}$$

   We use this formulation to prove that the last condition is satisfied.

$$P_{V_i}(\cup_j A_j) = P_{V_i}(\cup_j \cup_{v \in D_j} V_i = v) = \frac{card(\cup_j \cup_{v \in D_j} \{r \in \mathcal{R} | r(V_i) = v\})}{card(\mathcal{R}) - card(Dis(V_i))}$$

   As the events A_j are disjoint,
$$= \sum_j \frac{card(\cup_{v \in D_j} \{r \in \mathcal{R} | r(V_i) = v\})}{card(\mathcal{R}) - card(Dis(V_i))}$$
$$= \sum_j P_{V_i}(A_j)$$

   So the last condition is satisfied, for all random variables V_i, the new measure P_i defines a probability measure on each variable.

□

Proposition 2. *Given a set of random variables V, every function defined by $P_{W \subseteq V}(\bigcap_{W_i \in W} W_i = w_i)$, computed by the formula 2, is a joint probability function.*

Proof. We have to prove that for all set W of random variables such that $W \subseteq V$, the function P_W defined by

$$P_W(\cap_{W_i \in W} W_i = w_i) = \frac{card(\cap_{W_i \in W}\{r \in \mathcal{R} | r(W_i) = w_i\})}{card(\mathcal{R}) - card(Dis(W))}$$

is a joint probability function. $Dis(W)$ denotes the set of records disabled for W, i.e. the set of records for which at least one variable in W is unassigned.

First we show that $P_W(\cap_{W_i \in W} W_i = w_i)$ is in $[0, 1]$. As $\{r \in \mathcal{R} | \cap_{W_i \in W} r(W_i) = w_i\} \subseteq \mathcal{R} \backslash Dis(W)$, we can simply show that $P_W(\cap_{W_i \in W} W_i = w_i) \leq 1$, using the same proof as previously. Moreover, as it is defined by set cardinalities, it is necessary greater than 0.

Then we prove that $\sum_{w_i \in D(W_i)} M_W(\cap_{W_i \in W} W_i = w_i) = 1$.

$$\begin{aligned}
& \cup_{w_i \in \mathcal{D}(W_i)}\{r \in \mathcal{R} | \cap_{W_i \in W} r(W_i) = w_i\} && = \mathcal{R} \backslash Dis(W) \\
\Rightarrow\ & card(\cup_{w_i \in \mathcal{D}(W_i)}\{r \in \mathcal{R} | \cap_{W_i \in W} r(W_i) = w_i\} && = card(\mathcal{R}) - card(Dis(W)) \\
& \text{events } \cap_{W_i \in W} W_i = w_i \text{ being mutually exclusive,} \\
\Rightarrow\ & \sum_{w_i \in \mathcal{D}(W_i)} card(\{r \in \mathcal{R} | \cap_{W_i \in W} r(W_i) = w_i\}) && = card(\mathcal{R}) - card(Dis(W)) \\
\Rightarrow\ & \sum_{w_i \in \mathcal{D}(W_i)} \frac{card(\{r \in \mathcal{R} | \cap_{W_i \in W} r(W_i) = w_i\})}{card(\mathcal{R}) - card(Dis(W))} = \sum_{w_i \in \mathcal{D}(W_i)} M_W(\cap_{W_i \in W} W_i = w_i) = 1
\end{aligned}$$

□

3.3 Learning Algorithm

The proposition 2 allow us to apply all the formalisms defined for bayesian network learning methods with complete data. Our approach is based on the generic principle of constraint-based learning methods. More precisely, we implemented our algorithm from the Three Phase Dependency Analysis algorithm developed by [2], using the probability formulae introduced in the previous section for computing the conditional independency tests. The overall algorithm lays on three elementary steps:

1. conditional independencies are uncovered from the data using statistical tests,
2. then, these independencies are used to build a partially directed acyclic graph (PDAG) in two steps,
   - edges X–Y of an undirected fully connected graph are deleted for each pair of independent variables (X, Y),
   - the undirected graph then obtained is partially directed using the discovered conditional independencies;
3. last the PDAG is completed applying the following rules:

- if there is an edge such that $X \rightarrow Y$ and Z is adjacent to Y but not to X, then if there is an undirected edge between Y and Z, this edge is directed from Y to Z ($Y \rightarrow Z$),
- if there exists a directed path from X to Y and an undirected edge between X and Y, then this edge should be oriented from X to Y ($X \rightarrow Y$) to avoid building cycle.

4 Experiments

Our experiments were done to compare network structures generated by TPDA with complete data with the one obtained running TPDA adapted for handling missing values (TPDAID), using our own definitions for probability calculations, detailed by section 3. The goal was to show the validity of our redefinition of probabilities within the context of training bayesian network with incomplete data. We also aimed at measuring the robustness of our proposal to various incompleteness rate of the datasets.

4.1 Datasets

The results detailed here were obtained on several standard benchmarks often used by the bayesian network community. The characteristics of these datasets created from the various belief networks are described by Table 1.
From these complete datasets we generated incomplete ones. Missing values were randomly inserted in the database, replacing some attribute values. For each complete database, we thus created six incomplete datasets respectively containing 5%, 10%, 20%, 30%, 40% and 50% of missing values.

Table 1. Characteristics of the datasets

Dataset	# of attributes	# of records
Fire Network [33]	6	10,000
Asia / Chest Clinic Network [34]	8	5,000
Alarm Network [35]	37	10,000

4.2 Results

For each complete dataset and then incomplete datasets, we ran TPDA or TPDAID and so for each dataset we generated the bayesian network structure. Then we compared the graphs resulting from training with incomplete data to those resulting from training with complete data. To do so we analyzed the number of missing or additional edges and the number wrong directions.

Figure 4(a) shows the comparison between the graphs obtained for the *Fire* dataset according to the incompleteness rate. The original bayesian network contains five edges, it is described by Figure 3(a).

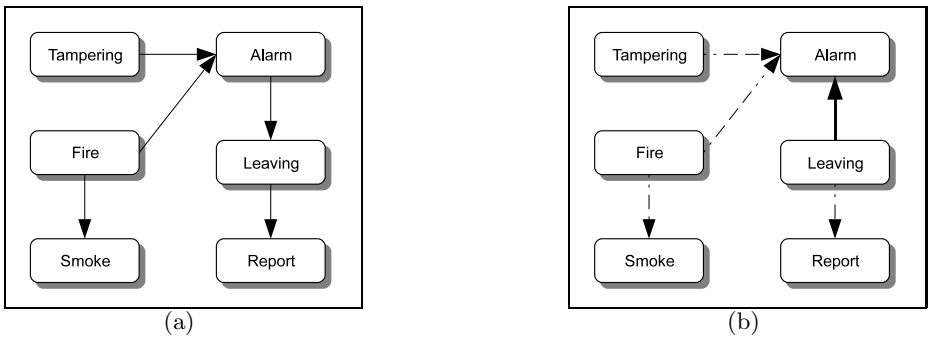

Fig. 3. (a): The *Fire* bayesian network structure (taken from [33]); (b): The *Fire* bayesian network structure obtained from 20% of missing values dataset.

For 5% and 10% of missing values, the graphs resulting of TPDAID are exactly the same as the complete database graph. Then the number of additional edges increases to 1 for 30 to 50% of missing values. On the *Fire* dataset it seems that the incompleteness rate influences more the number of wrong directions. Indeed, the graphs contain at least one wrong direction from 20% of missing values, as shown by Figure 3(b); the dashed arrows are the same as in the original network, the boldfaced one is the one modified.

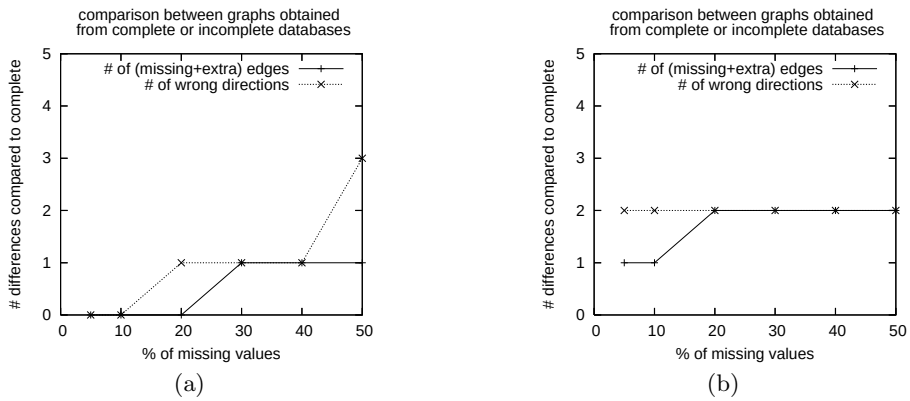

Fig. 4. (a): Results for the *Fire* dataset; (b): Results for the *Asia* dataset.

Figure 4(b) shows the comparison between graphs obtained for the *Asia* dataset according to the incompleteness rate. We can observe that results are not as good as with the *Fire* dataset for low incompleteness rates. However the

quality of the graph remains stable as the percentage of missing values increases.

Last, applying TPDAID on the *Alarm* incomplete datasets gives us interesting results for incompleteness rates below 30% of missing values. Indeed with these proportions of missing values – from 5 to 20% – the resulting structure is still close to the one obtained from TPDA on the complete dataset. From 30% of missing values the graph contains half of badly-oriented or additional edges, however half of the graph remains correctly built.

4.3 Synthesis and Future Work

Through our experiments, we observed that the data distribution and the incompleteness rate have an influence on how well our approach returns interesting results. Analysing the different results, we consider that our approach is robust until around 25% to 30% of missing values in the training dataset. However, there are important differences if we compare the results obtained on datasets with many or few variables and the number of records also influences the results.

Therefore, we are now working on improving the quality of the graph trained on incomplete datasets using our approach and reducing the influence of the data distribution. We plan to define a parameter, based on statistical properties, to bound the minimum number of complete records that should be used for computing each conditional independence test. Thus each probability should be computed on significant-enough valid databases.

Besides the second step for learning BN, i. e. learning the probability table for each node of the structure, should be tested. These experiments will aim at assessing how well our redefinition of probabilities is adapted to learn the parameters from an incomplete dataset. If these results are conclusive we will be able to propose a global dependency analysis algorithm for efficiently learning bayesian network structure and parameters from incomplete databases.

5 Conclusion

In this paper, we introduced a new method for learning bayesian networks from incomplete data. On the contrary to existing algorithms that are based on model scoring and selection, or on assessing or imputing missing values, our approach is based on dependency analysis and a redefinition of the probability calculations.

Thus while the other approaches are resource-demanding or time-consuming because of multiple iterations, our algorithm generates the graph computing conditional independence tests using a reformulation of probabilities. This redefinition is based on the principle that incomplete records contain some certain information that is exactly regarded, assigned attributes, and an uncertain part of information that should be ignored, missing values. This hypothesis enables us

to compute probabilities without multiple iteration for estimating missing values nor a biasing a priori imputation and without requiring external knowledge.

As the preliminary experimental results show, this new approach leads to quite good results for databases containing up to 30% of missing values. However, it can be improved by taking data distribution and statistical results into account to refine the probability calculation. We thus plan to develop a global algorithm that will both learn structure and parameters for bayesian network from incomplete datasets, based on our redefinition of probabilities that handle uncertainty contained in incomplete data.

References

1. Whittaker, J.: Graphical models in applied multivariate statistics. John Wiley & Sons, Inc. (1990)
2. Cheng, J., Bell, D., Liu, W.: Learning belief networks from data: an information theory based approach. In: the 6th ACM International Conference on Information and Knowledge Management. (1997) 207–216
3. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann (1988)
4. Cowell, R.G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J.: Probabilistic networks and expert systems. Statistics for engineering and information science. Springer-Verlag (1999)
5. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning **9**(4) (1992) 309–347
6. Spiegelhalter, D.J., Dawid, A.P., Lauritzen, S.L., Cowell, R.G.: Bayesian analysis in expert systems. Statistical Science **8** (1993) 219–282
7. Lam, W., Bacchus, F.: Learning bayesian belief networks: An approach based on the mdl principle. Computational Intelligence **10** (1994) 269–293
8. Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. Machine Learning **20**(3) (1995) 197–243
9. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory **14** (1968) 462–467
10. Pearl, J., Verma, T.S.: A theory of inferred causation. In: Principles of Knowledge Representation and Reasoning (KR'91). (1991) 441–452
11. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Lecture Notes in Statistics. Springer (1993)
12. Spirtes, P., Meek, C.: Learning bayesian networks with discrete variables from data. In: 1st International Conference on Knowledge Discovery and Data Mining (KDD'95). (1995)
13. Heckerman, D.: A tutorial on learning with bayesian networks. In: the NATO Advanced Study Institute on Learning in graphical models. (1998) 301–354
14. Lauritzen, S.L.: The em algorithm for graphical association models with missing data. Computational Statistics and Data Analysis **19** (1995) 191–201
15. Dempster, A.P., Laid, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society **39**(1) (1977) 1–38
16. Chickering, D.M., Heckerman, D.: Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. Machine Learning **29**(2-3) (1997) 181–212

17. Little, R.J.A., Rubin, D.B.: Statistical analysis with missing data. John Wiley & Sons, Inc. (1987)
18. Friedman, N.: Learning belief networks in the presence of missing values and hidden variables. In: 14th International Conference on Machine Learning. (1997) 125–133
19. Friedman, N.: The bayesian structural em algorithm. In: 14th Conference on Uncertainty in Artificial Intelligence. (1998) 129–138
20. Leray, P., François, O.: Bayesian network structural learning and incomplete data. In: International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05). (2005) 33–40
21. Myers, J.W., Laskey, K.B., Levitt, T.S.: Learning bayesian networks from incomplete data with stochastic search algorithms. In: 15th Conference on Uncertainty in Artificial Intelligence (UAI'99). (1999)
22. Myers, J.W., Laskey, K.B., Dejong, K.: Learning bayesian networks from incomplete data using evolutionary algorithms. In: Genetic and Evolutionary Computation Conference (GECCO'99). (1999)
23. Cowell, R.G.: Parameter estimation from incomplete data for bayesian networks. In: International Workshop on Artificial Intelligence and Statistics. (1999) 193–196
24. Ramoni, M.F., Sebastiani, P.: The use of exogenous knowledge to learn bayesian networks from incomplete databases. In: Second International Symposium on Advances in Intelligent Data Analysis and Reasoning about Data (IDA'97). Volume 1280 of Lecture Notes in Computer Science., Springer-Verlag (1997)
25. Ramoni, M.F., Sebastiani, P.: Parameter estimation in bayesian networks from incomplete databases. Intelligent Data Analysis **2**(1) (1998) 139–160
26. Ramoni, M.F., Sebastiani, P.: Learning bayesian networks from incomplete databases. In: 13th Conference on Uncertainty in Artificial Intelligence (UAI'97). (1997) 401–408
27. Riggelsen, C., Feelders, A.J.: Learning bayesian network models from incomplete data using importance sampling. In: 10th International Workshop on Artificial Intelligence and Statistics. (2005) 301–308
28. Li, X., He, X., Yuan, S.: Learning bayesian networks structures from incomplete data: An efficient approach based on extended evolutionary programming. In: 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, (PAKDD'05). (2005) 474–479
29. Li, X., He, X., Yuan, S.: A new method of learning bayesian networks structures from incomplete data. In: 15th International Conference on Artificial Neural Networks, (ICANN'05). (2005) 261–266
30. Riggelsen, C.: Learning bayesian networks from incomplete data: An efficient method for generating approximate predictive distributions. In: 6th SIAM International Conference on Data Mining (SDM'06). (2006)
31. Ragel, A., Cremilleux, B.: Treatment of missing values for association rules. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. (1998) 258–270
32. Agrawal, R., Imielinski, T., Swami, A.N.: Mining Association Rules between Sets of Items in Large Databases. In: the ACM SIGMOD International Conference on Management of Data. (1993) 207–216
33. Poole, D., Mackworth, A., Goebel, R.: Computational Intelligence. Oxford University Press (1998)
34. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. (1990) 415–448
35. Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F.