ELSEVIER International Journal of Approximate Reasoning 24 (2000) 39–57

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

# Priors on network structures. Biasing the search for Bayesian networks

Robert Castelo *, Arno Siebes

*CWI, P.O. Box 94079, 1090 GB, Amsterdam, Netherlands*

Received 1 November 1998; accepted 1 June 1999

**Abstract**

In this paper we show how a user can influence recovery of Bayesian networks from a database by specifying prior knowledge. The main novelty of our approach is that the user only has to provide partial prior knowledge, which is then completed to a full prior over all possible network structures. This partial prior knowledge is expressed among variables in an intuitive pairwise way, which embodies the uncertainty of the user about his/her own prior knowledge. Thus, the uncertainty of the model is updated in the normal Bayesian way. © 2000 Elsevier Science B.V. All rights reserved.

## 1. Introduction

Bayesian nets provide much insight in the conditional (in)dependencies among the attributes in a database. As such, Bayesian network recovery is an important tool for data miners. However, a straightforward recovery of these networks has two major drawbacks from the viewpoint of the user, who deals with real world data:

- In the first place, minor errors in the data may have large effects. For example, leading to counter-intuitive arrows.
- Secondly, our database might not be a fair random sample. This situation

* Corresponding author. Corresponding address: CWI, Information Systems (INSI), Kruislaan 413, 1098 SJ, Amsterdam, The Netherlands. Tel.: +31-20-5924123; fax: +31-20-5924199.

*E-mail addresses:* robert@cwi.nl (R. Castelo); arno@cwi.nl (A. Siebes).



PII: S0888-613X(99)00041-9

arises commonly with data gathered by inquiries that, for some reason, are not performed over a random sample of the population.

Both problems can be (partially) alleviated by allowing the user to specify a priori knowledge. In fact, there always exists some domain (expert) knowledge about a problem. This a priori knowledge should then be combined with the evidence in the database during the recovery. The resulting network is then consistent with both the user's a priori knowledge and the database. This last fact makes that, from the viewpoint of the user, the results are better and more acceptable.

So, the problem studied in this paper is how to let the user specify his a priori knowledge and how to use this knowledge to bias the search of the recovery algorithm.

Among the different approaches to learn Bayesian networks from data, we have carried out our work within the Bayesian framework. Therefore, whenever we use the term *probability*, we refer to a Bayesian (subjective) probability. In order to denote this fact, we will express the (Bayesian) probability of an event $e$ with $p(e \mid \xi)$, where $\xi$ indicates the background knowledge that is relevant to the assessment of this probability [3].

The standard Bayesian approach

$$posterior(model \mid data) \propto prior(model)\ likelihood(model, data)$$

translates to the posterior of a Bayesian network structure $B_s$ given a database $D$:

$$p(B_s \mid D, \xi) \propto p(B_s, D \mid \xi),$$
$$p(B_s \mid D, \xi) \propto p(B_s \mid \xi) p(D \mid B_s, \xi).$$

Let $\Theta$ be the set of parameters related to the Bayesian network structure $B_s$. Then

$$p(D \mid B_s, \xi) = \int_{\Theta} p(D \mid B_s, \Theta, \xi) f(\Theta \mid B_s, \xi)\, \mathrm{d}\Theta,$$

where the term $p(B_s \mid \xi)$ corresponds to the prior of the Bayesian network structure $B_s$. The reader may find a detailed description of the method in [1,4,5,8].

There are three earlier approaches in the Bayesian framework to the problem of how to let the user specify his a priori knowledge and how to use it to bias the search. The first, which we will nick-name the *partial theory* approach, is by Buntine [1]. The second, which we will nick-name the *penalizing* approach, is by Geiger et al. [5]. The third approach, which we will nick-name the *imaginary data* approach, is by Gavrin et al. [6].

In the partial theory approach, an initial partial theory provided by the expert is transformed into a prior probability over the space of theories. This partial theory consists of:

- A total ordering $\prec$ on variables, such that a parent set of a given variable must be a subset of the variables less than the given one (i.e., $y \in \pi_x \Rightarrow y \prec x$).
- A specification of beliefs for every possible arc, that a variable is parent of another one, measured in units of subjective probability.

The assumption of independence between parent sets is made, and thus a full prior conditioned on the total ordering of variables is given by

$$p(B_s \mid \prec, \xi) = \prod_{i=1}^{n} p(\pi_i \mid \prec, \xi),$$

where

$$p(\pi_i \mid \prec, \xi) = \left( \prod_{y \in \pi_i} p(y \to x_i \mid \prec, \xi) \right) \cdot \left( \prod_{y \notin \pi_i} (1 - p(y \to x_i \mid \prec, \xi)) \right).$$

Madigan and Raftery [8] also propose to elicit prior probabilities for the presence of every possible link and assuming that the links are mutually independent. However, they do not attach an order among variables as part of the prior information.

In the penalizing approach, the user builds a prior network from which it is possible (see [5]) to assess the joint probability distribution of the domain $U$ for the next case to be seen $p(U \mid B_{sc}, \xi)$ (where $B_{sc}$ is the complete network). From this joint probability distribution they then construct informative priors for the prior distribution of the parameters, yielding the Bayesian Dirichlet equivalent (BDe) metric.

In principle, the prior distribution of network structures is independent of this prior network, but they propose an approach where structures that closely resemble the prior network will tend to have higher prior probabilities, and these higher probabilities will be achieved by penalizing those networks that differ from the prior network.

Let $P$ be the prior network. The number of nodes in the symmetric difference of $\pi_i(B_s)$ and $\pi_i(P)$ is

$$\delta_i = |(\pi_i(B_s) \cup \pi_i(P)) \setminus (\pi_i(B_s) \cap \pi_i(P))|.$$

So, the amount of arcs $\delta$ in what the prior network and any network $B_s$ differ is

$$\delta = \sum_{i=1}^{n} \delta_i.$$

As we pointed out before, the idea is to penalize $B_s$ by a constant factor $0 < \kappa \leqslant 1$ for each such arc:

$$p(B_s \mid \xi) = c\kappa^{\delta},$$

where $c$ is a normalization constant.

In the imaginary data approach the user is asked to complete a certain amount of imaginary cases (each containing a random value in a variable chosen at random). This amount may depend on the problem domain. With this set of imaginary data the uniform prior probability over the sample space of Bayesian networks is updated. This updated distribution is then used as the prior distribution in the rest of the process.

The approach taken in this paper is that we assume far less prior knowledge from the user. Given two attributes $A$ and $B$ in the database, the user may specify his confidence in the possible connections between $A$ and $B$ in the network. We do not expect the user to have an opinion about all possible links. This partial prior knowledge of the user is then completed into a prior probability distribution on the space of possible networks.

In Section 2, the user's specification of his/her (incomplete) prior knowledge and its completion into a prior is discussed. In Section 3, we show how this prior information is actually used to bias the search for the discovered network. In Section 4, we give some experimental results that illustrate how the user's prior knowledge biases the search. In the final section we compare our approach with the three approaches discussed above and we formulate some problems for further research.

## 2. The prior

### 2.1. The user specification

Bayesian networks are graphically defined as acyclic digraphs (DAGs), and our main goal is to let the user define his/her preferences for some of this graphical objects as a probability distribution over the set of acyclic digraphs. A naive approach would be to obligate the user to give some prior probability to every DAG such that the priors for all possible networks sum to 1. This is impractical for the reason that no expert cannot be precise assessing some prior probability between 0 and 1 for an object formed by $n$ nodes and (up to) $n(n-1)/2$ arcs. We consider that assessing some degree of belief over the (in)dependency between two variables is natural for the user. We assume that the knowledge of the user is coherent, i.e., there are no contradictions in his/her beliefs. This assumption means that the user's beliefs over the three possible states of a link must yield a probability distribution. This is formally defined as follows.

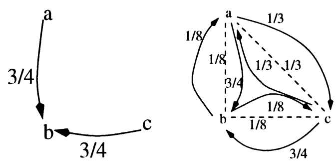
Let $a$ and $b$ be two nodes (variables) in a Bayesian network, the user may assess as prior knowledge in the link formed by these two nodes, a probability distribution over the three possible states of the link (arc in one direction $a \rightarrow b$, arc in opposite direction $a \leftarrow b$, no arc $a \cdots b$), which holds

$$p(a \rightarrow b \mid \xi) + p(a \leftarrow b \mid \xi) + p(a \cdots b \mid \xi) = 1.$$

In a Bayesian network with $n$ nodes there are $C_{n,2}$ [1] different links, and for every link, we consider a probability distribution over three states. For the links for which the user does not specify a prior, we assume an uniform prior:

$$p(a \rightarrow b \mid \xi) = p(a \leftarrow b \mid \xi) = p(a \cdots b \mid \xi) = \frac{1}{3}.$$

So, given a partial prior by the user, we may complete the prior for a link given this uniform distribution and the assumption that the user's beliefs are coherent. For example, we may see below on the left, a partial prior for a Bayesian network of three nodes, which may be specified by the user. On the right we may see its completion.




## 2.2. From an informal prior to a formal prior

We have shown the way we want the user to specify his/her partial knowledge and how to complete it to obtain a full prior. But still this prior is a collection of prior beliefs over a set of links. We need a full prior for a Bayesian network. Therefore, we are going now to specify, how to combine these priors of the links to obtain a full prior for a Bayesian network.

The amount of different objects we want to deal with (i.e., the amount of acyclic digraphs) is exponential in the number of nodes [9], so the situation asks for an incremental way of computing the full prior. This is, a way in which, given priors for a set of components (links) we obtain a prior for an acyclic digraph (a Bayesian network).

For us, the decomposition of a certain type of graphical object is useful as far as it allows us to enumerate all possible objects of the sample space (like all possible acyclic digraphs in our case). Such a decomposition keeps us aware of the set of objects that contain a given set of components, and then we are able to estimate the amount of confidence in the entire space of objects consistent with the given beliefs of the components.

Our main problem, we are going to discuss now, is that the natural decomposition of acyclic digraphs does not help us to build the full prior. Acyclic

[1] $C_{n,2} = \binom{n}{2}$.

digraphs are characterized by the so-called *out-points* [9]. Every node in a digraph has a (possibly empty) set of incoming arcs, and a (possibly empty) set of outgoing arcs. The cardinality of the former is the *in-degree* of the node, and the cardinality of the latter is the *out-degree* of the node. An *out-point* is a node in a digraph with *in-degree* 0. In other fields like operations research, this type of node is known as *source*, and its counter-part (the *in-point, out-degree* 0) as *sink*. Every acyclic digraph has at least one *out-point*, because has no directed cycles. We can decompose any acyclic digraph of $n$ nodes in sub-DAGs of $k$ out-points and $n - k$ non-outpoints for $1 \leqslant k \leqslant n$.

In our current situation, this decomposition is not useful since does not match the linkwise form of our prior components, which is also more intuitive for the user than some *out-point*-based formalization. We claim below that this linkwise form stems from the way we made our independence assumption among beliefs. To provide this intuitive way of decomposing an object let us assume for a moment that, instead of acyclic digraphs, we are working with *oriented graphs*.

An *oriented graph* [7] is a directed graph with no loops and no cycles of size two. So, it admits cycles of size greater than two. We can decompose this type of graphical object in links (pairs of nodes) such that for a given connection in this link, one-third of the whole space of objects will contain this concrete connection (arc in a certain direction, arc in the opposite direction, no arc).

To formalize the way we are going to combine the link priors, we should assume first that the beliefs of the user over different links are independent. In other words, what the user thinks about the pair of nodes, e.g., $a - b$ is not related to what the user thinks about $a - c$ or $c - d$, and so on. By this assumption, we define the combination of beliefs of different links as the product of their numerical values, which are probabilities. For example, these are the full priors for three different networks given the partial prior we showed above:

$$p(a \rightarrow b \leftarrow c \mid \xi) = \frac{3}{4} \times \frac{3}{4} \times \frac{1}{3} = \frac{3}{16},$$

$$p(a \rightarrow b \rightarrow c \mid \xi) = \frac{3}{4} \times \frac{1}{8} \times \frac{1}{3} = \frac{1}{32},$$

$$p(a \cdots b \rightarrow c \mid \xi) = \frac{1}{8} \times \frac{1}{8} \times \frac{1}{3} = \frac{1}{192}.$$

Gavrin et al. [6] pointed out that the assumption of independence among links is possibly unreasonable. We agree, but the benefit of making such assumption is that we require from the user the least amount of work to elicit a prior distribution.

The effect of using oriented graphs instead of acyclic digraphs as decomposable objects is that we are considering a sample space bigger than the one defined by acyclic digraphs. Due to those digraphs which contain one or more directed cycles. So, some amount of strength of our belief is distributed over a set of objects that will be never considered in the search we want to bias, the search for Bayesian networks. Therefore, we do not have a prior distribution over the set of possible Bayesian networks.

The solution we give to this problem is to compute the amount of weight we miss, and then we distribute it uniformly or proportionally over the set of acyclic digraphs. Let $\mathscr{A}_n$ be the set of acyclic digraphs of $n$ nodes. Let $\mathscr{O}_n$ be the set of oriented graphs of $n$ nodes. Let $\mathscr{C}_n = \mathscr{O}_n - \mathscr{A}_n$ be the set of digraphs that contain one or more directed cycle. Let $S_n = f(\mathscr{C}_n)$ be the sum of the prior values of the objects contained in $\mathscr{C}_n$. The function $f$ computes this sum given the set of digraphs with cycles, but for the moment we will not specify $f$. Using $S_n$ we can construct a prior distribution over the set of possible Bayesian networks of $n$ nodes in two ways:

- *Uniformly* let $A_n$ be the cardinality of $\mathscr{A}_n$. The amount of strength we sum to every acyclic digraph in $\mathscr{A}_n$ is

$$c = \frac{S_n}{A_n}.$$

- *Proportionally* we multiply every acyclic digraph by the value

$$c = \frac{1}{1 - S_n}.$$

In this way, the full prior for a Bayesian network $B = (B_s, B_p)$, where $B_s = (E, V)$ and $E$ is the set of edges and $V$ is the set of vertex such that $B_s$ is an acyclic digraph, may take one of these forms:

$$p(B_s \mid \xi) = c + \prod_{\substack{v_i, v_j \in V \\ i \neq j}} \rightleftharpoons v_j \mid \xi,$$

$$p(B_s \mid \xi) = c \prod_{\substack{v_i, v_j \in V \\ i \neq j}} p(v_i \rightleftharpoons v_j \mid \xi),$$

where $p(v_i \rightleftharpoons v_j)$ stands for the prior probability of certain connection ($v_i \rightarrow v_j$ or $v_i \leftarrow v_j$ or $v_i \cdots v_j$) specified in $E$ about the link $(v_i, v_j)$.

## 3. Using the prior

### 3.1. Constants do not matter

The aim of building a prior out of the background knowledge of some user, is to bias the search for a Bayesian network towards a model that contains the preferences expressed in this prior. Whenever there is no much evidence in the data against the user's beliefs, in that case the search will not be biased. Since the central role of the prior relies in the search process, it is easy to realize that the previous formulation of our prior is significantly simplified as follows. Let $B_s^1$ and $B_s^2$ be two Bayesian network structures involved in our search for a Bayesian network, with priors $p(B_s^1 \mid \xi)$ and $p(B_s^2 \mid \xi)$. Let $B_s^1 = (E_1, V)$ and $B_s^2 = (E_2, V)$, where $E_1, E_2$ are the sets of edges, and $V$ the set of vertex. As we already know, the Bayesian posterior that guides the search is proportional to the prior, so the larger prior, the better posterior. For some two Bayesian networks $B_s^1$ and $B_s^2$, in some point of the search they are compared, and let's say that $B_s^2$ has a better prior than $B_s^1$, thus

$$p(B_s^1 \mid \xi) < p(B_s^2 \mid \xi).$$

Let's expand the inequality with one of our formulas for the prior

$$c + \prod_{\substack{v_i, v_j \in V \\ i \neq j}} 1(v_i \rightleftharpoons v_j \mid \xi) < c + \prod_{\substack{v_i, v_j \in V \\ i \neq j}} p_2(v_i \rightleftharpoons v_j \mid \xi),$$

it is clear that the constants cancel themselves, and they do not modify the comparison among the priors. Therefore, we can use the improper prior

$$p(B_s \mid \xi) = \prod_{v_i, v_j \in V^{i \neq j}} p(v_i \rightleftharpoons v_j \mid \xi).$$

Clearly, this also holds in the case we expand the inequality with

$$p(B_s \mid \xi) = c \prod_{v_i, v_j \in V^{i \neq j}} p(v_i \rightleftharpoons v_j \mid \xi).$$

### *3.2. The new local measure*

Robinson [9] showed that number of acyclic digraphs grows exponentially in the number of nodes. Since the Bayesian network structures are acyclic digraphs (DAGs), it is infeasible to enumerate all of them and identify the structure with the highest posterior. Chickering [3] proves in his Ph.D. thesis that to learn Bayesian networks from data using the Bayesian posterior (concretely the BDe posterior [5]) is NP-complete. The way that the Bayesian posterior is developed and the assumptions made to find the final closed formula, afford to suit a range of search operators and search strategies. This makes it possible to learn Bayesian networks from data.

Acyclic digraphs may be splitted in sub-DAGs (one for every node), where each sub-DAG contains one *sink* node and its parent set of nodes. This decomposition is made within the development of the Bayesian posterior at the moment we factorize the probability of a case in a database through the *chain* *rule* and the assumption of completeness in the database. Chickering [3] calls the scoring functions that hold this property *decomposable scoring functions*. We recall below his definition.

**Definition 3.1** (*Decomposable scoring function*) Given a network structure, a measure on that structure is decomposable if it can be written as a product of measures, each of which is a function only of one node and its parents.

In this way, we treat separately every component, modifying and qualifying it, to combine later all the components in one Bayesian network. Thus, it is important that any further development in the learning process, as a prior to bias the *search*, is given in such a way that makes possible to compute it locally for every component and to combine it later with the rest of components. We will show that this is possible with the prior we give.

We can group links depending on whether they represent arcs for a concrete network with a *sink* node. Let $k$ be the amount of links where the user assessed some subjective probability. Let $S_0$ be the set of links with subjective probability derived from the prior given by the user, that for the network $B_s$, represent no arc. Let $\pi_i^p$ be the set of parent nodes of the node $x_i$ in the sub-DAG formed by the set of links specified as prior knowledge by the user. We can express $p(B_s \mid \xi)$ as follows:

$$p(B_s \mid \xi) = \left(\frac{1}{3}\right)^{C_{n,2}-k} \prod_{i=1}^{n} \left[ \prod_{j=1}^{|\pi_i^p|} p(\pi_{ij}^p \to x_i) \right] \prod_{(x,y)\in S_0} p(x \cdots y).$$

In this situation local changes are possible by changing single terms in the two main products. We may see that the second main product is not a function of one node and its parents. This term must be computed globally for every network, thus the expression, as a whole, is not fully decomposable. However, its complexity is $O(|S_0|)$ because it depends on how much information is provided by the user. Therefore, in practice $|S_0|$ is substantially smaller than $C_{n,2}$ and then the overhead in the computation caused by this global term is negligible.

### 3.3. The algorithm

In first place we will give a simple algorithm to complete a prior given by the user. Let $S_u$ be a set of vectors $(x, y, t, p)$ given by the user, where $x, y$ are two variables such that $x \neq y$, $t$ is an element of $\{\leftarrow, \rightarrow, \cdots\}$ specifying the type of prior information, and $p$ is the subjective probability that exists a connection of type $t$ between $x$ and $y$. We will denote by $S_c$ the set of vectors that complete the prior, the set that contains all the prior information is denoted by $S_p$. We will build $S_c$ as follows:

**let** $S_c = \emptyset$
**for** $v = (x, y, t, p) \in S_u$ **do**
  **let** $v' = (x, y, t', p')$
  **let** $v'' = (x, y, t'', p'')$
  **let** $t \cup t' \cup t'' = \{\leftarrow, \rightarrow, \cdots\}$
  **if** $v' \in S_u$ **and** $v'' \notin S_u$ **and** $v'' \notin S_c$ **then**
    $p'' = 1.0 - p - p'$
    $S_c \leftarrow S_c \cup v''$
  **else if** $v'' \in S_u$ **and** $v' \notin S_u$ **and** $v' \notin S_c$ **then**
    $p' = 1.0 - p - p''$
    $S_c \leftarrow S_c \cup v'$
  **else if** $v', v'' \notin S_u$ **and** $v', v'' \notin S_c$ **then**
    $p' = p'' = (1.0 - p)/2.0$
    $S_c \leftarrow S_c \cup v' \cup v''$
  **endif**
  $S_c \leftarrow S_c \cup v$
**endfor**

For computational reasons we will work with the logarithmic form of the prior

$$\log\, p(B_\mathrm{s} \mid \xi) = (C_{n,2} - k)\, \log\left(\frac{1}{3}\right) + \sum_{i=1}^{n}\left[\sum_{j=1}^{|\pi_i^\mathrm{p}|} \log\, p(\pi_{ij}^\mathrm{p} \rightarrow x_i)\right] + \sum_{(x,y)\in S_0} \log\, p(x \cdots y).$$

Then, for a given node $x_i$ we will compute the corresponding part of the prior $p(B_\mathrm{s})$ using the following function:

**function** *computeLocalPrior*$(x_i, \pi_i, S_p)$ **do**
  **let** $\pi_i^\mathrm{p} \leftarrow \{y : y \in \pi_i \wedge (y, x_i, \rightarrow, p) \in S_p\}$
  $prior \leftarrow 0$
  **for** $x_j \in \pi_i^\mathrm{p}$ **do**
    **let** $v = (x_j, x_i, \rightarrow, p) : v \in S_p$
    $prior \leftarrow prior + \log p$
  **endfor**
  **return** *prior*
**endfunction**

When the values of the priors of the components are combined (by summing them), we should compute the term corresponding to those links specified in the user's prior, where there is no arc in the network that is currently qualified, i,e.,

$$\sum_{(x,y)\in S_0} \log\, p(x\cdots y).$$

## 4. Experimental results

In this section our aim is twofold: to make clear how the prior works and to show an example that reproduces a situation we may find dealing with real world data. Both experiments have been realized using synthetic data thus we can evaluate the correctness of the results.

The Bayesian posterior used in this experimentation is the BDe with uninformative priors for the parameters, also known as BDeu [1,5]. This posterior assumes complete ignorance about the parameters of the Bayesian network, and the prior network involved in the posterior (do not confuse with our prior about the structure) is the empty network. The equivalent sample size that assesses the confidence of the user in this previous settings is also completely uninformative. The BDeu posterior assigns equivalent values to equivalent networks. A comprehensive and self-contained explanation of this settings is beyond the scope of this article. We recommend the reader to consult [1,5].

To show how the prior works, we will consider a small sample space of Bayesian networks (three nodes). We will bias the probability distribution of this sample space using our prior. This means that we will be changing the local maxima that a search process would achieve.

Let us consider we have two databases, db1 and db2, with 10 000 cases each. These databases reflect the independencies shown in Fig. 1.

They are claiming two different independence assertions: db1 infers $I(a, \emptyset, c)$, and db2 infers $I(a, b, c)$. We have mixed them in proportions from 0% to 100%, and in the Fig. 2 we may see the different probability distributions $p(B_s \mid D, \xi)$ over the set of possible DAGs. These two pictures show how the shape of the distribution changes through the different proportions of evidence towards the two original models from which we sample the data. The vertical axis indicates the value of probability, and the horizontal indicates the Bayesian network. The one generating db1 is on the second position in the horizontal axis, and the

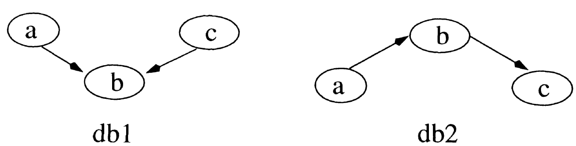



Fig. 1. Bayesian network structures corresponding to two different databases.

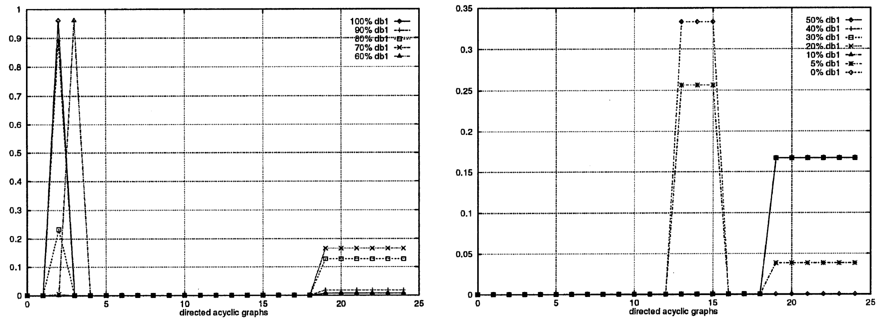
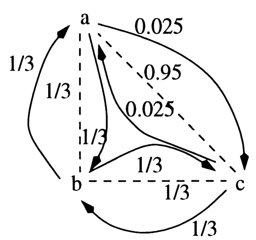



Fig. 2. Distributions of $p(B_s \mid D, \xi)$ for proportions of db1 from 0% to 100%.

ones (we use a score equivalent Bayesian measure) generating db2 are on the 13th, 14th and 15th positions in the horizontal axis. The last six positions of the horizontal axis correspond to the six complete Bayesian networks of three variables.

Let us take the database with a proportion of 70% of db1 and 30% of db2. This mixture of evidence benefits the six equivalent models that have all three variables mutually dependent (the complete network). We know that 70% of the database contains evidence that $a$ and $c$ are marginally independent while $b$ is conditionally dependent on $a$ and $c$. Therefore, by using prior information in the structure we want to see whether the existing evidence plus our prior knowledge allow us to bias the original distribution. We can achieve that by using the following prior:




In this prior we incorporate our notion of marginal independence between $a$ and $c$ by providing prior probability in the lack of an arc in either direction in the link formed by $a$ and $c$.

The distribution is biased in such a way that we could achieve a different local maxima in the search process, as we can see in Fig. 3.

Now, we want to show the prior working under more realistic circumstances. We have implemented this prior within an algorithm that uses the Bayesian posterior we described at the beginning of this section. Further, the learning algorithm uses as search strategy, a beam search with a beam of width 3, that in this case guarantees us to find always the highest posterior. The

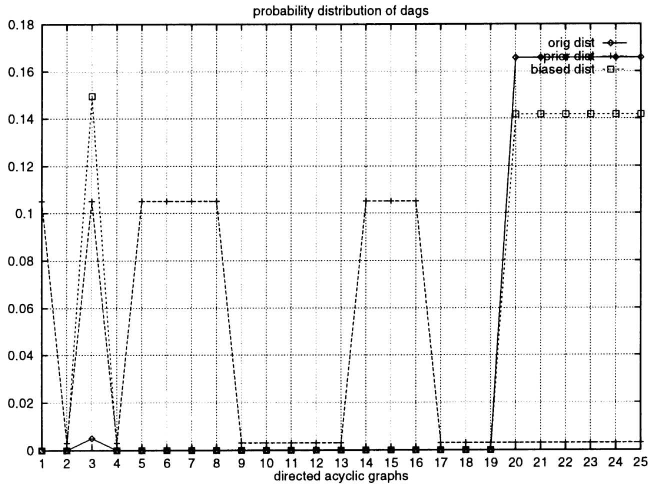



Fig. 3. Biased distribution by the effect of prior knowledge.

neighbour operator used by the beam search generates at every step of the search all possible networks with one arc more, one arc less and one arc reversed. For a more detailed description of the implementation of the learning algorithm the reader may consult [2].

Let us consider the Bayesian network of Fig. 4, as a possible model for a synthetic insurance domain. In this Bayesian network all arcs which direction is compelled are marked with C, and those that are reversible are marked with R, which in this case is just one.

From this network we sample a database of 100 000 records, by computing the entire probability distribution of tuples given the bayesian network of Fig. 4, and then each case is sampled by generating a random number between 0 and 1.

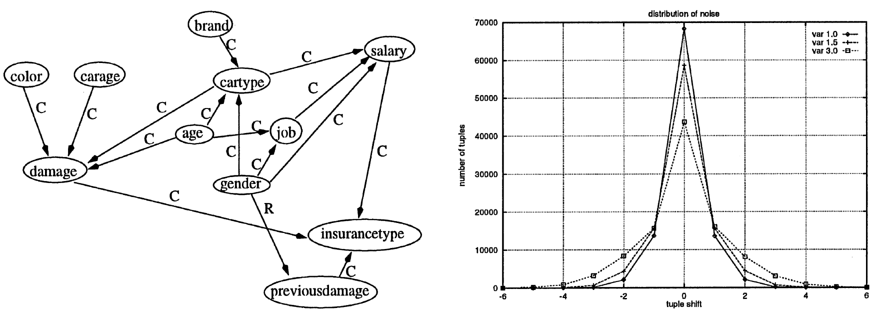



Fig. 4. Bayesian network for an insurance domain on the left, and distributions of Gaussian noise over a sample of 100 000 records for three different variances on the right.

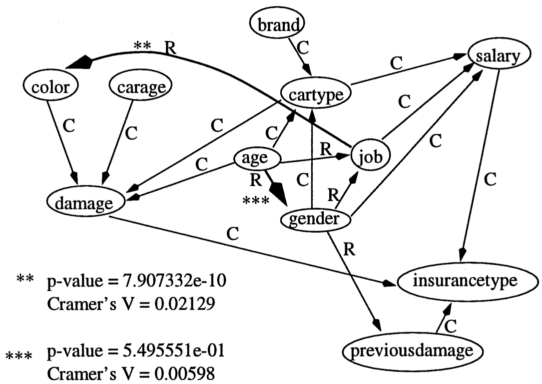



Fig. 5. The database contains 30% of tuples touched by noise.

In the generation of this sample, we introduce Gaussian noise with three different variances. Thus, we obtain three different databases with three different levels of noise. The effect of the noise is to disturb the selection of the proper tuple at each sampling of the probability distribution built from the Bayesian network. More concretely, given a random number between 0 and 1, we pick up the first tuple for which the accumulated probability is smaller than this random number. Given a normal value from the Gaussian distribution (with mean 0), this normal value may shift the selection of the tuple according to the random number. In Fig. 4 we may see how this noise is distributed. The horizontal axis gives the length of the shift caused by the noise, the vertical axis gives the amount of tuples that has been shifted. Those tuples with a null tuple shift are not disturbed by the noise. Thus, we may see that for a variance of 1.0, the 30% of the database is touched by noise. It means that in those tuples at least one value is different from what it should had been.

If we recover the network with the highest posterior from the database with noise ruled by a variance of 1.0, we obtain the Bayesian network of Fig. 5. This figure also shows the $P$-value of a $\chi^2$ test for independency and Cramer's $V$-value, for the two extra arc that have appeared. According to the $P$-value, the relation between *color* and *job* appears to be significant but the degree of association given by Cramer's $V$-value is not strong. In the other case the relation between *age* and *gender* does not seem to be even significant. Another difference is that the two arcs between *job*, and *age* and *gender* are now reversible because the latter extra arc leads them to be covered. [2]

Our goal is first to see whether we can bias the search and obtain the original model using prior knowledge. Second, to find out how strong our prior

[2] An arc is covered when the parent set of the sink coincides with the parent set of the source.

knowledge must be to achieve the first purpose. Thus, we can get a feeling of how the evidence about the original model is deteriorated by such amount of noise in the database. It is also important to realize in which minimal configuration of prior knowledge we can obtain the original model.

By trying different values combined in different ways over the arcs that have been modified we have found out the following. To remove the extra arc between *color* and *job* is necessary to assess $p(color \cdots job \mid \xi) = 0.95$, while to remove the extra arc between *age* and *gender*, it does not suffice to assess only $p(age \cdots gender \mid \xi) = 0.95$. In fact, for the current model underlying this data, that is not the way of removing it. To remove this extra arc, and to fix as compelled the arcs between *job* and *age* and *gender* as they originally were, one should assess $p(gender \rightarrow job \mid \xi) = 0.4$.

The degree of association between *gender* and *job* is the second strongest one and the search process adds an arc between these two variables at the second step of the beam search. By setting this prior we are expressing our preference over a model where a compelled arc should appear pointing to *job*, that is to say, the arc should not be covered, and this implies that the arc between *age* and *job* should not be covered either, thus both of them pointing to *job*. Because of the strong association between *gender* and *job* that leads the process to link them in the second step of the search, a correction towards the right model in this step leads the whole search to achieve the model we expected. If the amount of evidence would account differently for this link, we would have to set our prior knowledge in a different way. The assertions of (conditional) independence contained in the recovered model (the I-map) are then the sum of the evidence of the database plus our prior knowledge about the model.

When one considers databases of the size we have been using now, the evidence about a certain fact may be very large. We have seen that the *P*-value for a significant relation was practically 0. The size of a database may help to smooth the effect of noisy tuples. So, we are now going to show what happens if we introduce the same proportion of noise, but in a sample ten times smaller: a database of 10 000 records. Under this condition we obtain the Bayesian network of the Fig. 6.

In this case, we are not able to bias the search towards the original model, even if we believe, let's say in 9/10, that the missed arc between *previousdamage* and *insurancetype* exists. Let $B_s$ be the original Bayesian network structure and $B'_s$ the Bayesian network structure of Fig. 6. The log-likelihoods of these two networks are:

$$\log p(B_s, D \mid \xi) = -115518.77,$$

$$\log p(B'_s, D \mid \xi) = -115414.45,$$

$$\mathbf{diff} = -104.32.$$

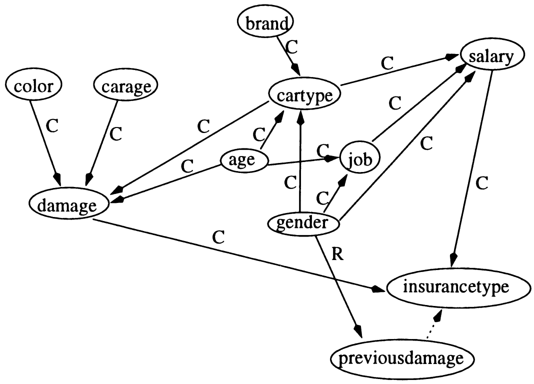



Fig. 6. Sample of 10 000 records, where 30% are touched by noise.

So, we would have to believe in 0.999..99 about 100 nines to bias the search. Of course, such belief would not make sense, and the only conclusion we may draw is that sometimes, and in this case, we cannot win. There is enough evidence in the database against a direct relation between the two variables mentioned before.

Finally, we will treat the case of having a database that is a bad random sample of a certain underlying model. We have simulated this by generating a sample of 10 000 records using the built-in random generator of the standard C library [3] (the *rand*() function) to sample from the model. In this case the Bayesian network with the highest posterior is showed in Fig. 7.

Similar to the first case, the two extra arcs are covering the arc between *damage* and *age* and this latter one becomes reversible. We can recover the original model setting probabilities on the modified arcs towards their proper form, but by looking for the minimal amount of prior knowledge we need, we can find out which portion of evidence has been deteriorated. In this case, this portion affects the compelled nature of the link between *damage* and *age*.

By setting a prior probability in this link of $p(age \rightarrow damage \mid \xi) = 0.4$ we recover the correct model. Provide that this value is only slightly over the ignorance threshold 1/3, it means that the original evidence is not too deteriorated, and that it is more sensible to consider the original model as the one that generated the data, which in fact is true.

The relation that now is present in the model, between *cartype* and *damage* is significant ($P$-value $= 1.687106\,\mathrm{e} - 10$), but with a weak degree of association (Cramer's $V = 0.06387$).

[3] Which is known to be pretty bad.

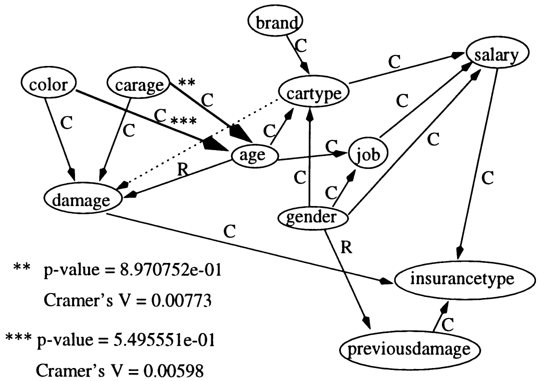



Fig. 7. Bad random sample.

## 5. Discussion

In the formalization of our approach to incorporate prior knowledge, we work within the framework of Bayesian statistics. This keeps the induction process sound. The beliefs of the user are requested to be coherent, this means that the user should think in terms of which are his/her preferences among three existing possibilities of connection between two variables. In the discussion of the construction of a full prior we have seen that independence assumptions over prior knowledge are coupled with the nature of the models we try to induce. Our approximation of the full prior by using *oriented graphs* looks good given the results in the experimentation with synthetic data. Of course it would be desirable to find a better coupling between Bayesian networks (acyclic digraphs) and independence assumptions over prior knowledge.

If we compare our work with the existing approaches, the most important difference is that we do not expect the user to have prior knowledge about the whole network structure. Partial prior information can be taken into account as well. Compared with the partial theory approach, where a total ordering on the variables is required, an important difference is that the user's prior belief can be negated by the facts in the database. That is, the user may think that $A$ is a parent of $B$ with a 99% probability, but if the database overwhelmingly supports that $B$ is a parent of $A$, then in the final network, $B$ will be a parent of $A$, while in the partial approach the order overrides any evidence.

Compared with the penalizing approach, an important difference is that we achieve our aims not by penalizing networks that differ much from the user's prior belief, but by Bayesian updating of the user's prior belief with the facts from the database. Penalizing overrides the user's uncertainty about how variables are linked. Finally, compared with the imaginary data approach, our solution requires the least amount of work of the user.

The fact that the database can override the prior belief of the user could be also seen as a weakness of the approach taken in this paper. In the previous section we have seen that in a rather small database the user already needs a very high confidence in his knowledge to ''win'' from the data. Although this is a straightforward effect of Bayesian updating it may appear counter-intuitive to the user. Currently, we are working on an approach in which the user may specify his prior beliefs by a (partial) database. This will allow the user to state that he can think of 100 000 cases in which $A$ is the parent of $B$. We hope that the user will feel more confident in supplying such a number of cases rather than a prior probability of 99.999%. Given this (partial) database, the prior probability can be computed in a way very similar to that in the current paper. We are taking into account as well how the notion of equivalent sample size, used by Geiger et al. [5] for their prior network, is related to this idea, and also how is related to the imaginary data approach from Gavrin et al. [6].

Another extension we are working on is to allow the user to specify his prior knowledge in chunks larger than single links, thus trying to relax the assumption of independence among links. It is very well possible that the user beliefs that $A$ is a parent of $B$, if $C$ is also a parent of $B$ but that he has another opinion if $C$ turns out to be a child of $B$. In principle, this problem is not much different from the one studied in this paper, the major difference lies in the completion of the prior probability.

Concerning the use of this prior, much more experimentation must be done, mainly in front of real world problems. This is the only way to know the added value of prior knowledge in data analysis and interpretation of results. Garvin et al. [6] provided an experiment where they show how prior knowledge can improve predictive performance.

## Acknowledgements

This paper owes much to discussions with Ulises Cortes and Ramón Sangüesa while they were visiting CWI. Many thanks to them.

## References

[1] W.L. Buntine, Theory refinement on Bayesian networks, in: P. Smets, B.D. D'Ambrosio, P.P. Bonissone (Eds.), Proceedings of Uncertainty in Artificial Intelligence, vol. 7, Morgan Kaufmann, Los Altos, CA, 1991, pp. 52–60.

[2] R. Castelo, A.P.J.M. Siebes, Bayesian networks in a data mining tool, in: Proc. Jornades d'Inteligencia Artificial: Noves Tendencies, Bulletin of the ACIA, vol. 12, University of Lleida, Spain, October 1997, pp. 70–78.

[3] D.M. Chickering, Learning Bayesian networks from data, Ph.D. thesis, Department of Computer Science, University of California, Los Angeles, CA, 1996.

[4] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (1992) 309–347.
[5] D. Geiger, D. Heckerman, D.M. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, Machine Learning 20 (1995) 197–243.
[6] J. Gavrin, D. Madigan, A.E. Raftery, Eliciting prior information to enhance the predictive performance of Bayesian graphical models, Communications in Statistics – Theory and Methods 24 (1995) 2271–2292.
[7] F. Harary, E.M. Palmer, Graphical Enumeration, Academic Press, New York, 1973.
[8] D. Madigan, A.E. Raftery, Model selection and accounting for model uncertainty in graphical models using Occam's window, Journal of the American Statistical Association 89 (428) (1994) 1535–1546.
[9] R.W. Robinson, Counting labeled acyclic digraphs, in: F. Harary (Ed.), New Directions in the Theory of Graphs, Academic Press, New York, 1973, pp. 239–273.