On Evidence Absorption for Belief Networks

Linda C. van der Gaag
Utrecht University
Department of Computer Science
P.O. Box 80.089, 3508 TB Utrecht
The Netherlands

Abstract

More and more real-life applications of the belief-network framework are emerging. As applications grow larger, the belief networks involved increase in size accordingly. For large belief networks, probabilistic inference tends to become rather time-consuming. In the worst case this tendency may not be denied as probabilistic inference is known to be NP-hard. However, it is possible to improve on the *average-case* performance of the algorithms involved. For this purpose, the method of *evidence absorption* can be exploited. In this paper, we detail the method of evidence absorption and outline its integration into a well-known algorithm for probabilistic inference. The ability of the method to improve on the average-case computational expense of probabilistic inference is illustrated by means of experiments performed on both randomly generated and real-life belief networks.

Key words: belief networks, probabilistic inference, evidence absorption, (average-case) computational complexity.

1 Introduction

The *belief-network* framework for reasoning with uncertainty in knowledge-based systems has been around for some time now, and more and more practical applications employing the framework are being developed [1, 2, 3]. As applications of the framework grow larger, the belief networks involved increase in size accordingly: belief networks comprising hundreds, or even thousands, of variables are no exception. For belief networks of this size, probabilistic inference shows a tendency to become rather time-consuming, even so to an unacceptable extent. Since probabilistic inference is known to be *NP-hard* [4], this tendency may not be denied in general: the basic algorithms associated with a belief network have an exponential worst-case computational time complexity and it is not expected that a general polynomial-time algorithm will be found. In this paper, we address improving on the *average-case* performance of algorithms for probabilistic inference.

The average-case computational expense of probabilistic inference with a belief network may be improved in many different ways. We propose exploiting for this purpose the method of *evidence absorption*. The method of evidence absorption has been first introduced by R.D. Shachter as part of an algorithm for processing evidence in a belief network [5]. The basic idea of the method is to *dynamically* modify a belief network as evidence becomes available so as to explicitly represent newly created independences. Since all algorithms for probabilistic inference with a belief network exploit the represented independences more or less

directly, the incorporation of evidence absorption into these algorithms is expected to speed up computation while still providing for exact inference. The actual speed-up attained by employing the method in practical applications, however, depends on the inference algorithm used and on the topological properties of the belief network involved.

In this paper, we detail the method of evidence absorption and illustrate its incorporation into Pearl's algorithm for probabilistic inference. The paper is organised as follows. In Section 2, the basic notions involved in the belief network formalism are provided; in addition, we briefly review Pearl's basic algorithm for probabilistic inference and its enhancement with loop cutset conditioning. In Section 3, the method of evidence absorption is detailed. Section 4 addresses incorporation of the method into Pearl's enhanced algorithm. In Section 5, we illustrate the ability of the method of evidence absorption to save on the computational expense of inference by means of experiments performed on randomly generated and real-life belief networks. The paper is rounded off with some conclusions in Section 6.

2 Preliminaries

In this section we review the basic notions involved in the belief-network formalism and briefly outline Pearl's enhanced algorithm for probabilistic inference with a belief network; for further details, the reader is referred to [6].

2.1 The Belief-Network Formalism

A *belief network* is a terse representation of a joint probability distribution on a set of statistical variables. It consists of a qualitative and a quantitative part. The qualitative part of a belief network is a graphical representation of the interdependences between the variables discerned; it takes the form of an acyclic directed graph. In this digraph, each vertex represents a variable that can take one of a set of values. The arcs represent dependences between the variables: informally speaking, we take an arc $V_i \rightarrow V_j$ to represent a direct influential relationship between the variables V_i and V_j, where the direction of the arc designates V_j as the effect of V_i. Absence of an arc between two vertices means that their variables do not influence each other directly, and hence are (conditionally) independent. Associated with the digraph of a belief network is a set of functions representing probabilities from the distribution at hand, with each other constituting the quantitative part of the network.

Before defining the concept of a belief network more formally, we provide some additional terminology and introduce our notational convention. In the sequel, we will restrict the discussion to *binary* variables, taking one of the values *true* and *false*; the generalisation to variables with more than two discrete values, however, is straightforward. We will use the following notation: v_i denotes the proposition that the variable V_i takes the truth value *true*; $V_i = \mathit{false}$ will be denoted by $\neg v_i$. For a given set of variables V, the conjunction $C_V = \bigwedge_{V_i \in V} V_i$ of all variables from V is called the *configuration template* of V; a conjunction c_V of value assignments to the variables from V is called a *configuration* of V. In the sequel, we will use $\{C_V\}$ to denote the set of all configurations of V. Furthermore, we will write C_{V_i} and c_{V_i} instead of $C_{\{V_i\}}$ and $c_{\{V_i\}}$, respectively, for singleton sets $\{V_i\}$. The independence relation embedded in a joint probability distribution Pr will be denoted as $I_{\Pr}$; an *independence statement* $I_{\Pr}(X, Y, Z)$ signifies that in the distribution Pr the sets of variables X and Z are conditionally independent given the set of variables Y.

We now define the concept of a belief network more formally.

Definition 2.1 *A* belief network *is a tuple* $B = (G, ,)$ *such that*

- $G = (V(G), A(G))$ *is an acyclic digraph with vertices* $V(G) = \{V_1, \ldots, V_n\}$, $n \geq 1$, *and*
- $, = \{\gamma_{V_i} \mid V_i \in V(G)\}$ *is a set of real-valued functions* $\gamma_{V_i} : \{C_{V_i}\} \times \{C_{\pi_G(V_i)}\} \rightarrow [0, 1]$, *called* probability assessment functions, *such that for each configuration* $c_{\pi_G(V_i)}$ *of the set* $\pi_G(V_i)$ *of (immediate) predecessors of vertex* V_i *we have that* $\gamma_{V_i}(\neg v_i \mid c_{\pi_G(V_i)}) = 1 - \gamma_{V_i}(v_i \mid c_{\pi_G(V_i)})$, $i = 1, \ldots, n$.

Note that in the previous definition V_i is viewed as a vertex from the digraph and as a statistical variable, alternatively.

To link the qualitative and quantitative parts of a belief network, a probabilistic meaning is assigned to the topology of the digraph of the network [6].

Definition 2.2 *Let* $G = (V(G), A(G))$ *be an acyclic digraph and let* s *be a chain in* G. *Then, we say that* s *is* blocked *by a set of vertices* $W \subseteq V(G)$ *if* s *contains three consecutive vertices* X_1, X_2, X_3 *for which one of the following conditions holds:*

- $X_1 \leftarrow X_2$ *and* $X_2 \rightarrow X_3$ *are on the chain* s *and* $X_2 \in W$;
- $X_1 \rightarrow X_2$ *and* $X_2 \rightarrow X_3$ *are on the chain* s *and* $X_2 \in W$;
- $X_1 \rightarrow X_2$ *and* $X_2 \leftarrow X_3$ *are on the chain* s, *and* $\sigma^*_G(X_2) \cap W = \varnothing$, *where* $\sigma^*_G(X_2)$ *denotes the set of vertices composed of* X_2 *and all its descendants in* G.

Building on the notion of blocking we define the d-separation criterion.

Definition 2.3 *Let* $G = (V(G), A(G))$ *be an acyclic digraph and let* $X, Y, Z \subseteq V(G)$ *be sets of vertices from* G. *The set* Y *is said to* d-separate *the sets* X *and* Z, *denoted as* $\langle X | Y | Z \rangle^d_G$, *if for each* $V_i \in X$ *and* $V_j \in Z$ *every chain from* V_i *to* V_j *in* G *is blocked by* Y.

The d-separation criterion provides for reading independence statements from a digraph, as stated in the following definition.

Definition 2.4 *Let* $G = (V(G), A(G))$ *be an acyclic digraph. Let* Pr *be a joint probability distribution on* $V(G)$ *and let* $I_{\Pr}$ *be the independence relation of* Pr. *Then, the digraph* G *is called an* I-map *for* Pr *if for all mutually disjoint sets* $X, Y, Z \subseteq V(G)$ *we have: if* $\langle X | Y | Z \rangle^d_G$ *then* $I_{\Pr}(X, Y, Z)$.

The following theorem now states that the probability assessment functions of a belief network provide all information necessary for uniquely defining a joint probability distribution on the variables discerned that respects the independence relation portrayed by the graphical part of the network; henceforth, we will call this distribution the *joint probability distribution defined by the network.*

Theorem 2.5 *Let* $B = (G, ,)$ *be a belief network as defined in Definition 2.1. Then,*

$$\Pr(C_{V(G)}) \quad = \quad \prod_{V_i \in V(G)} \gamma_{V_i}(V_i \mid C_{\pi_G(V_i)})$$

defines a joint probability distribution Pr *on the set of variables* $V(G)$ *such that* G *is an I-map for* Pr.

2.2 Pearl's Enhanced Algorithm for Probabilistic Inference

Since a belief network defines a joint probability distribution, it can be used for *probabilistic inference*. An algorithm for probabilistic inference with a belief network provides for computing probabilities of interest and for processing evidence, that is, for entering evidence into the network and subsequently computing the revised probability distribution given the evidence. Several such algorithms have been developed [5, 6, 7]. Here, we only briefly review the basic idea of the algorithm designed by J. Pearl [6].

In outlining Pearl's algorithm for probabilistic inference, we take an object-centered point of view. The digraph of a belief network is taken as a *computational architecture*: the vertices of the digraph are autonomous objects having a local processor and a local memory in which the associated probability assessment function is stored; the arcs of the digraph are bi-directional communication channels. Through these communication channels the vertices send each other *parameters* providing information about the represented joint probability distribution and the evidence entered so far. Each vertex is equipped with a set of computation rules for computing the probabilities of its values and the parameters to send to its neighbours, from the information it receives from these neighbours and its own local probability assessment function. Initially, the network is in an *equilibrium state*: repeated computation of the parameters does not result in a change in any of them. When a piece of evidence is entered into the network, however, this equilibrium is *perturbed*. The vertex for which the evidence has been entered modifies the parameters to send to its neighbours to reflect the new information. These modifications activate updating parameters throughout the entire network: after receiving modified parameters, each vertex in turn computes new parameters to send to its neighbours. If the digraph of the network is *singly connected*, then a piece of evidence is diffused through the network in a single pass: the network will reach a new equilibrium state once every vertex has been visited, that correctly reflects the updated joint probability distribution given the evidence.

Unfortunately, Pearl's algorithm applies to belief networks involving a singly connected digraph only. Straightforward application of the algorithm to an acyclic digraph comprising one or more loops leads to insuperable problems [8]: vertices may indefinitely send updated messages to their neighbours causing the network never to reach a new equilibrium, or, if the network *does* reach an equilibrium, it is not guaranteed to correctly reflect the updated joint probability distribution. Pearl has proposed several methods for probabilistic inference with a belief network comprising a *multiply connected digraph* [6]. Of these, the method of *loop cutset conditioning* may be looked upon as a supplement to the basic algorithm. The idea underlying this method is that of *reasoning by assumption*. For a multiply connected digraph, vertices are selected that, upon instantiation, with each other effectively 'cut' or block all loops and cause the digraph to behave as if it were singly connected; the selected vertices are said to constitute the *loop cutset* of the digraph. Each configuration of the loop cutset now is looked upon as an assumption on which reasoning is performed. For each vertex, the probabilities of its values are computed by conditioning successively on all possible configurations of the loop cutset and subsequently weighting the results obtained. In the sequel, we will use the phrase *Pearl's enhanced algorithm* to denote Pearl's basic algorithm supplemented with the method of loop cutset conditioning for general probabilistic inference.

The details of the various computations involved in Pearl's basic algorithm and in loop cutset conditioning are not relevant to the present paper. It suffices to note that the computational expense of probabilistic inference using Pearl's enhanced algorithm is largely determined

by the topology of the digraph of the belief network at hand. Informally speaking, Pearl's enhanced algorithm performs the better from a computational point of view as the digraph is sparser.

3 Evidence Absorption

A belief network generally is constructed to reflect as many of the independences between the variables discerned as possible. There are several reasons for seeking to represent these independences to accuracy. One of these reasons is a computational one. The more independences are represented explicitly, the sparser the digraph of the network will be and, as we have mentioned before, the sparser the digraph, the lesser the computational expense of probabilistic inference with the network. Now observe that during reasoning with a belief network, evidence is entered and processed. Each new piece of evidence provides additional information about the represented joint probability distribution for a given context. More in specific, new dependences and independences may have come to hold in this context. It is possible to modify the topology of the digraph of the network dynamically so as to reflect these newly created dependences and independences explicitly. In fact, Shachter's algorithm for probabilistic inference is built on this very idea [5]. As we will argue in the sequel, however, it is worthwhile to modify the topology of the digraph to reflect the new *in*dependences only. The method of *evidence absorption* is designed for this purpose.

Informally speaking, the method of evidence absorption amounts to modifying a belief network after a piece of evidence has been entered for some variable so as to reflect the newly created independences. The topology of the digraph of the network is modified by deleting all arcs emanating from the vertex for which the evidence has been entered; in addition, the probability assessment functions for the (former) successors of this vertex are adjusted. The modified network is defined more formally in the following definition.

Definition 3.1 *Let $B = (G, ,)$ be a belief network where $G = (V(G), A(G))$ is an acyclic digraph and $, = \{\gamma_{V_i} \mid V_i \in V(G)\}$ is a set of associated probability assessment functions. Let V_i be a vertex in G for which the evidence $V_i = true$ is entered. We define the tuple $B^{v_i} = (G^{v_i}, , {}^{v_i})$ by*

- *$G^{v_i} = (V(G^{v_i}), A(G^{v_i}))$ is the acyclic digraph with $V(G^{v_i}) = V(G)$ and $A(G^{v_i}) = A(G) \setminus \{(V_i, V_j) \mid V_j \in \sigma_G(V_i)\}$ where $\sigma_G(V_i)$ is the set of all (immediate) successors of the vertex V_i in G, and*
- *$, {}^{v_i} = \{\gamma^{v_i}_{V_j} \mid V_j \in V(G)\}$ is the set of real-valued functions $\gamma^{v_i}_{V_j} : \{C_{V_j}\} \times \{C_{\pi_{G^{v_i}}(V_j)}\} \rightarrow [0, 1]$ with*
  - *$\gamma^{v_i}_{V_j}(V_j \mid C_{\pi_{G^{v_i}}(V_j)}) = \gamma_{V_j}(V_j \mid C_{\pi_G(V_j)\setminus\{V_i\}} \wedge v_i)$, for all vertices $V_j \in \sigma_G(V_i)$, and*
  - *$\gamma^{v_i}_{V_k}(V_k \mid C_{\pi_{G^{v_i}}(V_k)}) = \gamma_{V_k}(V_k \mid C_{\pi_G(V_k)})$, for all vertices $V_k \in V(G) \setminus \sigma_G(V_i)$.*

The tuple $B^{\neg v_i} = (G^{\neg v_i}, , {}^{\neg v_i})$ is defined analogously by substituting $\neg v_i$ for v_i in the above.

It will be evident that the modified network resulting after evidence absorption once more is a belief network.

The method of evidence absorption is illustrated by means of an example.

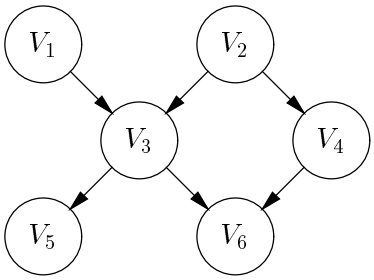

Figure 1: The Digraph G of the Example Belief Network B.

Example 3.2 Consider the belief network $B = (G, ,)$ where G is the multiply connected digraph shown in Figure 1 and , consists of the six probability assessment functions $\gamma_{V_1}, \ldots, \gamma_{V_6}$:

$\gamma_{V_1}(V_1)$
$\gamma_{V_2}(V_2)$
$\gamma_{V_3}(V_3 \mid V_1 \wedge V_2)$
$\gamma_{V_4}(V_4 \mid V_2)$
$\gamma_{V_5}(V_5 \mid V_3)$
$\gamma_{V_6}(V_6 \mid V_3 \wedge V_4)$

Now suppose that the evidence $V_3 = \mathit{true}$ is obtained for the variable V_3. The belief network B then is modified to $B^{v_3} = (G^{v_3}, , {}^{v_3})$. The digraph G^{v_3} is obtained from G by deleting all arcs emanating from vertex V_3, and is shown in Figure 2; the evidence for the variable V_3 is represented by drawing vertex V_3 with shading.

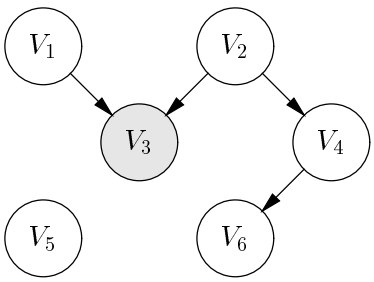

Figure 2: The Digraph G^{v_3} of the Belief Network B^{v_3}.

The set , ^{v_3} consists of the six functions $\gamma^{v_3}_{V_1}, \ldots, \gamma^{v_3}_{V_6}$ that are obtained from the probability assessment functions of the original belief network B:

$$\begin{aligned}
\gamma^{v_3}_{V_1}(V_1) &= \gamma_{V_1}(V_1) \\
\gamma^{v_3}_{V_2}(V_2) &= \gamma_{V_2}(V_2) \\
\gamma^{v_3}_{V_3}(V_3 \mid V_1 \wedge V_2) &= \gamma_{V_3}(V_3 \mid V_1 \wedge V_2) \\
\gamma^{v_3}_{V_4}(V_4 \mid V_2) &= \gamma_{V_4}(V_4 \mid V_2) \\
\gamma^{v_3}_{V_5}(V_5) &= \gamma_{V_5}(V_5 \mid v_3) \\
\gamma^{v_3}_{V_6}(V_6 \mid V_4) &= \gamma_{V_6}(V_6 \mid v_3 \wedge V_4)
\end{aligned}$$

□

The following proposition states that, after evidence absorption, the modified belief network and the original belief network model the same *updated* joint probability distribution given the evidence.

Proposition 3.3 *Let $B = (G, ,)$ be a belief network and let* Pr *be the joint probability distribution defined by B. Let V_i be a vertex in G for which the evidence $V_i = true$ is observed and let* $\Pr^{v_i}$ *denote the updated joint probability distribution given $V_i = true$. Now, let the network $B^{v_i} = (G^{v_i}, , {}^{v_i})$ be defined as in Definition 3.1 and let $\mathbb{P}$ be the joint probability distribution defined by B^{v_i}. Furthermore, let $\mathbb{P}^{v_i}$ denote the updated joint probability distribution given $V_i = true$. Then,* $\Pr^{v_i} = \mathbb{P}^{v_i}$.

Proof. We consider the belief network $B = (G, ,)$ and its joint probability distribution Pr, and the modified network $B^{v_i} = (G^{v_i}, , {}^{v_i})$ and its joint probability distribution $\mathbb{P}$. To prove that $\Pr^{v_i} = \mathbb{P}^{v_i}$, we show that

$$\Pr(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n) = \mathbb{P}(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n)$$

The main result then follows from the property of marginalisation and the definition of conditional probability.

From Theorem 2.5, we have that the joint probability distribution Pr defined by the belief network B can be expressed as

$$\Pr(V_1 \wedge \cdots \wedge V_n) \quad = \quad \prod_{V_j \in V(G)} \gamma_{V_j}(V_j \mid C_{\pi_G(V_j)})$$

From this expression, we derive an expression for the marginal distribution $\Pr(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n)$ by filling in the value v_i for the variable V_i.

The joint probability distribution $\mathbb{P}$ defined by the belief network B^{v_i} can be expressed as

$$\mathbb{P}(V_1 \wedge \cdots \wedge V_n) \quad = \quad \prod_{V_j \in V(G)} \gamma^{v_i}_{V_j}(V_j \mid C_{\pi_{G^{v_i}}(V_j)})$$

From this expression, we derive an expression for the marginal distribution $\mathbb{P}(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n)$ by substituting the value v_i for the variable V_i.

To show that $\Pr(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n) = \mathbb{P}(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n)$, it suffices to show that the various terms in the expressions for the marginal distributions stated above match. We distinguish between several different cases:

- for the assessment functions γ_{V_i} and $\gamma^{v_i}_{V_i}$ for the variable V_i, we have that

  $$\gamma_{V_i}(v_i \mid C_{\pi_G(V_i)}) = \gamma^{v_i}_{V_i}(v_i \mid C_{\pi_{G^{v_i}}(V_i)})$$

  by definition;

- for the assessment functions γ_{V_j} and $\gamma^{v_i}_{V_j}$ for a variable V_j with $V_j \in \sigma_G(V_i)$, we have that

  $$\gamma_{V_j}(V_j \mid C_{\pi_G(V_j) \setminus \{V_i\}} \wedge v_i) = \gamma^{v_i}_{V_j}(V_j \mid C_{\pi_{G^{v_i}}(V_j)})$$

  by definition;

- no other assessment function involves the variable V_i; for the functions γ_{V_k} and $\gamma^{v_i}_{V_k}$ for a variable V_k with $V_k \in V(G) \setminus (\sigma_G(V_i) \cup \{V_i\})$, we therefore have that

  $$\gamma_{V_k}(V_k \mid C_{\pi_G(V_k)}) = \gamma^{v_i}_{V_k}(V_k \mid C_{\pi_{G^{v_i}}(V_k)})$$

  by definition.

We conclude that

$$\Pr(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n) = \mathbb{P}(V_1 \wedge \cdots \wedge V_{i-1} \wedge v_i \wedge V_{i+1} \wedge \cdots \wedge V_n)$$

□

Note that a similar property holds with respect to evidence $V_i = \mathit{false}$.

From the previous proposition and its proof, it is easily seen that application of the method of evidence absorption cannot introduce into the modified network any independences conditional on the evidence that were not already reflected by the original network. In the following lemma, we state more formally that the two networks represent the same independences given the evidence; a separate proof for this property is provided in [9].

Lemma 3.4 *Let $B = (G, ,)$ be a belief network with $G = (V(G), A(G))$. Let V_i be a vertex in G for which the evidence $V_i = \mathit{true}$ is observed and let the network $B^{v_i} = (G^{v_i}, , {}^{v_i})$ be defined as in Definition 3.1. Then, $\langle X | Y | Z \rangle_G^d$ if and only if $\langle X | Y | Z \rangle_{G^{v_i}}^d$, for all sets $X, Y, Z \subseteq V(G)$ such that $V_i \in Y$.*

So far we have considered applying the method of evidence absorption for one piece of evidence only. It will be evident, however, that Proposition 3.3 and Lemma 3.4 are easily generalised to hold for multiple pieces of evidence.

As we have mentioned before, the method of evidence absorption has been first introduced by R.D. Shachter as part of an algorithm for processing evidence in a belief network. The basic idea of this algorithm is to *eliminate* a vertex from a belief network as soon as it is instantiated, modifying the network to reflect the *updated* probability distribution given the evidence for the vertex. The algorithm is composed of two phases. When a piece of evidence is entered for a specific variable, the method of evidence absorption is applied. Subsequently, the evidence is spread throughout the network by a method called *evidence propagation* which basically consists of repeated application of an arc-modifying operation called *arc reversal*. In these two phases, the topology of the digraph of the network is modified dynamically to reflect the newly created independences *and* dependences. In doing so, new arcs may be inserted into the digraph to portray the newly created dependences among the remaining variables and for these arcs accompanying conditional probabilities are calculated.

Shachter's algorithm for processing evidence has some drawbacks, as has been noted before by J. Pearl [6 (pp. 144 – 145)]. Related to the computational effort involved, we note that eliminating an instantiated vertex from a belief network is computationally expensive: the algorithm has an exponential worst-case time complexity. In addition, the computational expense of further probabilistic inference with the modified belief network after elimination may increase as a result of the insertion of new arcs into the digraph of the network. These drawbacks cannot be alleviated if the aim is to eliminate an instantiated vertex from the network. Upon close examination of Shachter's algorithm for processing evidence, it becomes clear, however, that these drawbacks arise to a large extent from the arc-reversal operation employed during evidence propagation: it is this method that accounts for the high computational expense. As opposed to evidence propagation, evidence absorption can be performed efficiently as all computations involved are local to a vertex and its successors — in fact, evidence absorption will generally take constant time.

4 Incorporating Evidence Absorption into Pearl's Algorithm

The method of evidence absorption has been designed to dynamically modify a belief network as evidence becomes available to explicitly represent the new independences holding in view of the evidence. All existing algorithms for exact probabilistic inference exploit such independences more or less directly. Pearl's (enhanced) algorithm, for example, tends to perform the better from a computational point of view as the digraph of a belief network is the sparser, that is, as the digraph portrays the more independences. Since the method of evidence absorption aims at explicitly representing new independences only, it tends to delete arcs from the digraph of a belief network and never inserts any new ones. The incorporation of this method into the existing algorithms, therefore, is expected to improve on the average-case computational expense of probabilistic inference.

The method of evidence absorption is easily incorporated into Pearl's basic algorithm for probabilistic inference with a belief network involving a singly connected digraph. The basic idea is as follows. When a piece of evidence is entered into the belief network for some vertex, the method of evidence absorption is applied *before* propagating the evidence. Then, Pearl's algorithm is called upon to perform the actual propagation. In contrast with Shachter's algorithm for probabilistic inference, the instantiated vertex is *not* eliminated from the network: as the method of evidence absorption models new independences only, the instantiated vertex has to remain in the digraph of the network to properly reflect the newly created dependences. Note that the ability of the method of evidence absorption to improve on the average-case computational expense of probabilistic inference with a belief network comprising a singly connected digraph derives from its effect on the topology of this digraph: if applying evidence absorption lets the digraph of the network fall apart into (equally large) components, then any further probabilistic inference can be restricted to one component only. Also note that the speed-up of inference obtained easily outweighs the computational effort of evidence absorption.

The incorporation of the method of evidence absorption into Pearl's enhanced algorithm for probabilistic inference with a belief network involving a multiply connected digraph in essence is the same as its incorporation into Pearl's basic algorithm. In view of loop cutset conditioning, however, the concept of evidence absorption can even be exploited to a further extent. We recall from Section 2 that for a multiply connected digraph a loop cutset is selected that, upon instantiation, effectively 'cuts' all loops and causes the digraph to behave as if it were singly connected. Now observe that a piece of evidence may equally provide for 'cutting' one or more loops of the digraph at hand. So, when evidence is entered, it may render one or more vertices of the loop cutset obsolete. The method of evidence absorption therefore provides for dynamically reducing an initial loop cutset as evidence is entered into a belief network involving a multiply connect digraph; for further details of dynamic loop cutset reduction, we refer to a forthcoming paper [10].

5 The Experiments and Their Results

In the previous sections, we have detailed the method of evidence absorption and its incorporation into Pearl's enhanced algorithm for probabilistic inference. The most interesting question to address now is what impact applying the method of evidence absorption has on the topology of the digraph of a belief network as successive evidence is entered, since this

impact can be related directly to the computational expense involved in further probabilistic inference.

From a theoretical point of view, the best case and the worst case are easily identified. The *worst* case would be a digraph for which evidence is entered only for vertices without any arcs emanating from them. In this case, applying the method of evidence absorption is pointless: there are no arcs deleted from the digraph and further computations are just as expensive as when evidence absorption had not been applied. It is worth noting, however, that the method of evidence absorption would not weigh heavily on the computational effort spent on probabilistic inference: the only additional work required would be a simple check on a vertex' successor set. In the *best* case, the method of evidence absorption causes the digraph of the network to fall apart into components of size one only for a single piece of evidence: this would be a digraph having the shape of a tree of depth one for which evidence is entered for the root vertex.

The above observations are general and not very illuminating. To gain more insight into the impact of the method of evidence absorption, we have conducted several experiments on different classes of randomly generated belief networks. In addition, we have analysed the impact of evidence absorption on some real-life networks.

5.1 Experiments on Randomly Generated Belief Networks

The aim of our experiments with the method of evidence absorption on randomly generated belief networks is to gain insight into the impact of the method on the average-case computational expense of probabilistic inference. Since this impact derives from the way the method modifies the graphical part of a belief network and not from the modification of the associated conditional probabilities, we have designed our experiments to apply to the graphical part of a network only.

The Set-Up of the Experiments
In each experiment, we have generated a set of one hundred (connected) acyclic digraphs by means of a *graph generator*; for further details of the graph generator used, we refer the reader to [11]. Each digraph is randomly generated to comprise n vertices, $n \geq 1$, and m arcs, $n - 1 \leq m \leq \frac{1}{2}n \cdot (n - 1)$. To study the impact of *repeated* application of the method of evidence absorption, in each experiment we have entered k pieces of evidence into the digraphs generated; we have modelled entering a piece of evidence by selecting a vertex from the set of vertices of the digraph at hand and applying the modifying operation of the method of evidence absorption to the digraph's topology. Vertices modelling pieces of evidence are selected by means of an *evidence generator*. This generator selects vertices from the digraph at hand either randomly or with one of two different biases. These biases concern the location in the digraph of the vertices for which evidence is entered and have been introduced into the evidence generator because it is expected that the location in the digraph of the vertices for which evidence is entered plays a major role in the impact of the method of evidence absorption on a digraph's topology. We would like to note that for diagnostic applications the vertices for which evidence is entered tend to be located in the lower part of the digraph whereas for prognostic applications these vertices are more likely to be located in the upper part of the network. In this paper, however, we will not address the impact of these biases. For further details of the evidence generator and for an overview of all experiments performed and their results, we refer the reader once more to [11].

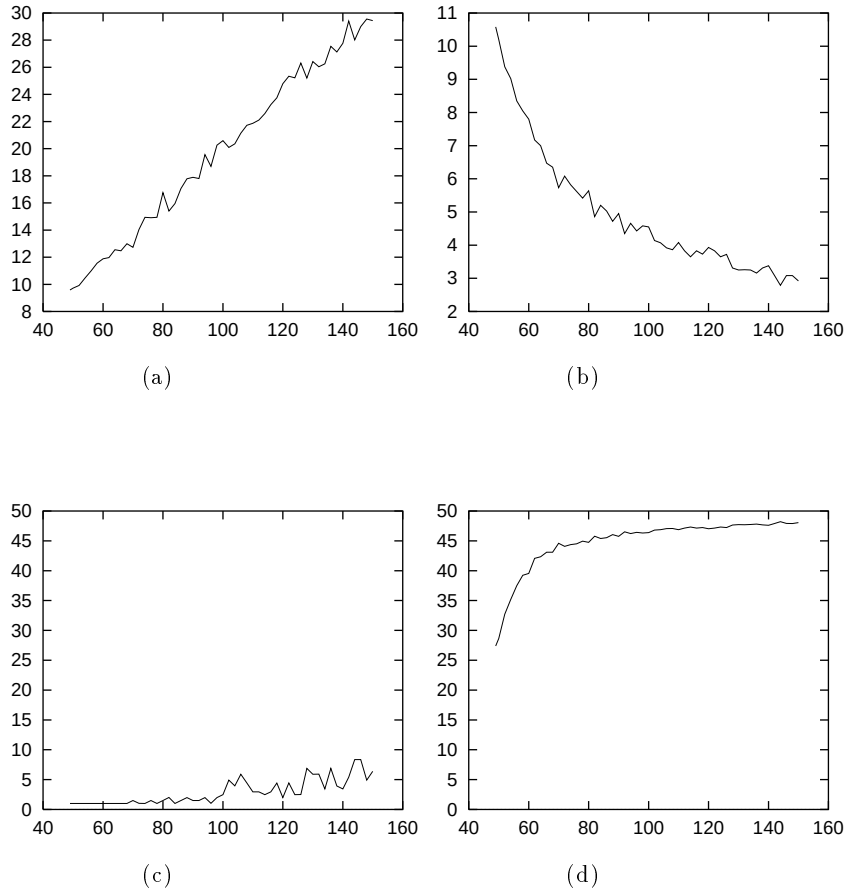

Figure 3: The Results of the First Experiment — (a) The Average Number of Deleted Arcs, (b) The Average Number of Components, (c) The Average Size of the Minimum Component, (d) The Average Size of the Maximum Component

The Results of the Experiments

The aim of the first experiment reported here has been to study, in isolation, the influence of the degree of connectivity on the behaviour of a digraph's topology under evidence absorption. In this experiment, we have generated several sets of one hundred digraphs comprising fifty vertices each. We have varied the number of arcs of the generated digraphs from forty-nign, and fifty up to one hundred and fifty, increasing by two for each set. To each digraph generated, we have applied the method of evidence absorption for ten randomly selected pieces of evidence. For the modified digraphs, we have found the statistics summarised in Figure 3; Figure 3(a) shows the average number of deleted arcs, in Figure 3(b) the average number of components of the modified digraphs is shown, and Figures 3(c) and 3(d) plot the average sizes of the minimum and maximum component of the modified digraphs, respectively.

The second experiment reported here is similar to the first one in the sense that its aim also is to study the impact of one of the parameters defining the search space for experimentation in isolation: it is the number of pieces of evidence that is varied in this experiment. In this

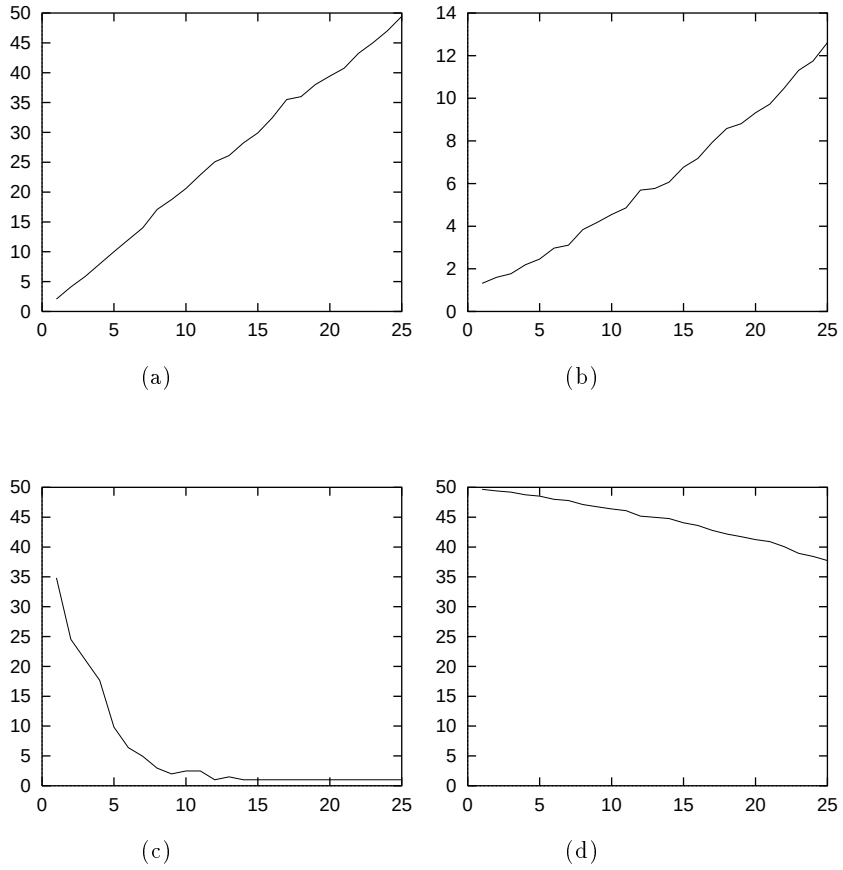

Figure 4: The Results of the Second Experiment — (a) The Average Number of Deleted Arcs, (b) The Average Number of Components, (c) The Average Size of the Minimum Component, (d) The Average Size of the Maximum Component

experiment, we have generated several sets of digraphs comprising fifty vertices each; we have fixed the number of arcs of these digraphs to one hundred. The pieces of evidence entered into these digraphs have been generated randomly; the number of pieces of evidence entered is varied from one up to twenty-five, increasing by one for each set of digraphs. To each digraph generated, we have applied the method of evidence absorption for the pieces of evidence selected. For the modified digraphs, we have found the statistics summarised in Figure 4; Figure 4(a) shows the average number of deleted arcs, in Figure 4(b) the average number of components of the modified digraphs is shown, and Figures 4(c) and 4(d) plot the average sizes of the minimum and maximum component of the modified digraphs, respectively.

Discussion

We begin our discussion of the results obtained from our experiments by considering the average numbers of deleted arcs. From a theoretical point of view, we observe that in a digraph comprising n vertices and m arcs, the average number of arcs emanating from a vertex equals $\frac{m}{n}$. When applying the method of evidence absorption for one piece of evidence,

the number of deleted arcs therefore is expected to approximate this ratio. Since deleting the arcs emanating from one vertex does not influence the number of arcs emanating from any of the other vertices in the digraph, we find that for k pieces of evidence the number of deleted arcs is expected to approximate $k \cdot \frac{m}{n}$. For a given digraph, this formula indicates a linear relation between the number of pieces of evidence entered and the number of arcs deleted by evidence absorption. The results of our experiments confirm this observation; Figure 4(a) shows a linear increase in the number of deleted arcs for an increasing number of pieces of evidence entered. From the formula $k \cdot \frac{m}{n}$, we further observe that the number of arcs deleted by evidence absorption for a fixed number of pieces of evidence is related linearly to the total number of arcs comprised in the digraph at hand. This observation is also confirmed by our experiments; Figure 3(a) indicates a linear increase in the number of deleted arcs for an increasing total number of arcs.

We now address the average numbers of components and their respective sizes found in the experiments. For this purpose, we first consider the generation of a random digraph by successive addition of arcs between randomly selected vertices [12]. It will be evident that the more arcs are added to a digraph in the making, the more likely it is to become connected. A well-known result from random graph theory is that a random digraph with n vertices is almost always connected if it comprises $O(n \cdot \log n)$ arcs or more. Moreover, a random digraph with between $O(n)$ and $O(n \cdot \log n)$ arcs typically comprises one large component of $O(n)$ vertices, called the *giant component*, and many small components of size at most $O(\log n)$ each. Now consider adding to a digraph having the topology just described an arc between two randomly selected vertices. We distinguish between three situations:

- the new arc connects two vertices comprised in the giant component — the probability that this situation will occur is rather high and increases as the giant component increases in size;

- the new arc connects one vertex from within the giant component and one vertex from within one of the tiny components — the probability that this situation will occur is fairly small and even diminishes as the giant component grows; note that since adding such an arc results in the giant component encapsulating a tiny one, we have that the probability that the giant component will increase in size is inversely proportional to its current size;

- the new arc connects two vertices not yet comprised in the giant component — the probability that this situation will occur is very small and even diminishes as the giant component grows.

We now observe that the behaviour of the topology of a random digraph under arc deletion is dual to its behaviour under arc addition. From this observation we have that by successive arc deletion a connected random digraph will at first stay connected until it has shrunk to comprise approximately $O(n \cdot \log n)$ arcs. Further arc deletion will tend to yield a topology in which one giant component can be discerned and many tiny ones.

The digraphs generated in our experiments with the method of evidence absorption are rather sparse and therefore are likely to exhibit the behaviour outlined above. In fact, the presence and behaviour of the giant component is reflected in Figures 3(d) and 4(d). Figure 3(d) shows that as the number of arcs of the generated digraphs increases, the size of the giant component rapidly rises to approximate the number of vertices of the digraphs; note

that the amount of increase in size of the giant component for an increase in the number of arcs is inversely proportional to the size the component already has. Figure 4(d) shows that as the number of pieces of evidence entered, and hence the number of deleted arcs, increases, the giant component slowly decreases in size. Figures 3(b) and 4(b) depict the average number of components found in our experiments. Figure 3(b) shows that the number of components rapidly decreases as the number of arcs of the digraphs, and hence the size of the giant component, increases; Figure 4(b) shows that the number of components increases as the number of pieces of evidence entered increases. Both Figure 3(c) and 4(c) demonstrate that the size of the minimum component, and hence the size of the tiny components, is very small compared to the size of the giant component.

5.2 Experiments on Real-life Belief Networks

So far we have considered the impact of the method of evidence absorption in view of experiments on randomly generated belief networks. A close examination of the results obtained from the experiments reveals several interesting properties. From the discussion in the previous section it will be evident, however, that these properties to a large extent derive from applying the method to randomly generated digraphs — in fact, the results obtained from our experiments cannot be generalised to apply to the method's behaviour on belief networks that do not incorporate a random digraph. In addition to our experiments on randomly generated belief networks, we therefore have also done some experiments on real-life networks, among which is the HEPAR belief network [13].

The *HEPAR* belief network is a small medical belief network for the diagnosis of *Wilson's disease*. Wilson's disease is a recessively inherited derangement of the copper metabolism in the human body; it typically results in progressive copper accumulation in the liver, causing cirrhosis, and in copper deposits in other organs, causing extrahepatic disorders, such as renal and neurological disease. The qualitative part of the HEPAR belief network is shown in Figure 5. The digraph comprises 21 vertices and 23 arcs. Note that, although the digraph does not have a high degree of connectivity, it includes several loops. In the figure, the vertices for which evidence may be obtained are drawn with shading. Of these, the vertices labeled *Free serum copper*, *Serum caeruloplasmin*, and *Urinary copper* represent the concentration of copper in various body fluids which can be determined by laboratory tests; the values of the other vertices are directly available from a patient's interview and physical examination.

In the digraph, almost all vertices pertaining to readily available evidence have no arcs emanating from them. Applying the method of evidence absorption for these vertices therefore has no impact on the digraph's topology whatsoever. The only exception is the vertex labeled *Age*. Upon application of the method of evidence absorption for this vertex, two arcs are deleted from the digraph; note that the deletion of these arcs reduces the number of loops in the digraph. If, in addition to the data from interview and physical examination, the laboratory test results for *Free serum copper* and *Serum caeruloplasmin* are available, then another four arcs are deleted from the digraph and several loops are cut. Moreover, the digraph falls apart into several components. The largest of these components comprises 12 vertices and 12 arcs; it contains a single loop. As this component includes the *Wilson's disease* vertex modeling the hypothesis, further probabilistic inference is restricted to this very component. We would like to note that the HEPAR belief network concerns one hepatic disorder only and is projected to be part of a larger network modeling some 80 disorders of the liver and biliary tract.

Figure 5: The Digraph of the *HEPAR* Belief Network.

The impact of the method of evidence absorption as outlined above for the HEPAR belief network appears to be typical for (small-scaled) *diagnostic* belief networks: we have found similar results for other networks such as for example the *ALARM Monitoring* system [14].

We would like to note that at present only few full-scaled, real-life belief networks are available from the literature, rendering extensive experiments on such networks practically infeasible. Also, most present-day belief networks have been designed for the task of diagnosis and therefore are expected to share the characteristic of evidence vertices being located mainly in the lower part of the network's digraph. Furthermore, most existing networks are tailored to state-of-the-art methods for reasoning with a belief network which tend to impose restrictions on the topology of the graphical part of the network. Since research on reasoning methods rapidly progresses, future belief networks may very well differ considerably from present-day ones. We feel that as applications grow larger, the digraphs involved will tend to have a topology in which subgraphs with a high degree of connectivity can be discerned modelling different focal areas of attention of the domain at hand; these dense subgraphs will tend to be loosely interconnected. As long as this tendency is not confirmed by full-scaled real-life belief networks, we should be careful in drawing any decisive conclusions as to the true ability of the method of evidence absorption.

6 Conclusions

In this paper, we have addressed the tendency of the basic algorithms for probabilistic inference associated with the belief network formalism to become the major consumers of computing resources. We have mentioned that in the worst case this tendency cannot be denied as these algorithms have an exponential worst-case time complexity. It is possible, however, to improve on the average-case performance of these algorithms. To this end, we have proposed incorporating the method of *evidence absorption* into Pearl's (enhanced) algorithm for probabilistic inference. This method amounts to dynamically modifying a belief network as evidence becomes available. The ability of the method to improve on the average-case performance of probabilistic inference derives from the method's property of explicitly incorporating the new independences created by the observation of the evidence into the digraph of the network: the method tends to delete arcs and to make a digraph fall apart into separate components.

To gain some insight in the ability of the method of evidence absorption to improve on the computational expense involved in inference, we have performed several experiments on different classes of randomly generated belief networks. Unfortunately, the results obtained from these experiments to a large extent reflect the use of randomly generated belief networks and do not provide for drawing detailed conclusions as to the method's behaviour on real-life networks that do not incorporate a digraph of random topology. Also, the results of experiments on real-life belief network cannot be generalised straightforwardly to apply to all types of (future) belief networks. Since the impact of applying the method of evidence absorption on probabilistic inference is determined by the topological properties of the digraph of the network at hand, however, it can be decided for each belief network separately whether or not applying evidence absorption is expected to be advantageous. To this end, a simple investigation of the location in the network's digraph of the vertices for which evidence is likely to be entered suffices.

To conclude, we would like to note that, although in this paper we have addressed incorporation of the method of evidence absorption into Pearl's algorithm for probabilistic inference only, the method is as easily introduced into the Lauritzen and Spiegelhalter algorithm, requiring only simple operations on a junction tree. Moreover, the method effortlessly amalgamates with other methods for improving on the computational expense of probabilistic inference.

References

[1] S. Andreassen, M. Woldbye, B. Falck, S.K. Andersen. MUNIN — A causal probabilistic network for interpretation of electromyographic findings, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp. 366 – 372, 1987.

[2] D.E. Heckerman, E.J. Horvitz, B.N. Nathwani. Toward normative expert systems. Part 1: The Pathfinder project, *Methods of Information in Medicine*, vol. 31, pp. 90 – 105, 1992.

[3] P.D. Bruza, L.C. van der Gaag. Index expression belief networks for information disclosure, *The International Journal of Expert Systems: Research and Applications*, vol. 7, no. 2, pp. 107 – 138, 1994.

[4] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks, *Artificial Intelligence*, vol. 42, pp. 393 – 405, 1990.

[5] R.D. Shachter. Evidence absorption and propagation through evidence reversals, in: M. Henrion, R.D. Shachter, L.N. Kanal, J.F. Lemmer (editors). *Uncertainty in Artificial Intelligence 5*, Elsevier Science Publishers (North-Holland), Amsterdam, pp. 173 – 190, 1990.

[6] J. Pearl. *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference*, Morgan Kaufmann, Palo Alto, 1988.

[7] S.L. Lauritzen, D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems, *Journal of the Royal Statistical Society, Series B*, vol. 50, pp. 157 – 224, 1988.

[8] H.J. Suermondt and G.F. Cooper. Probabilistic inference in multiply connected belief networks using loop cutsets. *International Journal of Approximate Reasoning*, vol. 4, pp. 283-306, 1990.

[9] L.C. van der Gaag. *Evidence Absorption for Belief Networks*, Technical Report CS-RUU-93-35, Utrecht University, 1993.

[10] E. Kaspers, L.C. van der Gaag. *Dynamic Loop Cutset Reduction*, in preparation.

[11] L.C. van der Gaag. *Evidence Absorption – Experiments on Different Classes of Randomly Generated Belief Networks*, Technical Report UU-CS-94-42, Utrecht University, 1994.

[12] B. Bollobas. *Random Graphs*, Academic Press, London, 1985.

[13] M. Korver, P.J.F. Lucas. Converting a rule-based expert system into a belief network, *Medical Informatics*, vol. 18, pp. 219 – 241, 1993.

[14] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, G.F. Cooper. The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks, in: J. Hunter, J. Cookson, J. Wyatt (eds.) *AIME-89 Proceedings of the Second Conference on Artificial Intelligence in Medicine*, Springer-Verlag, Berlin, pp. 247 – 256, 1989.