Bayesian Networks: Aspects of Approximate Inference

Bayesiaanse Netwerken: Aspecten van Benaderend Rekenen
(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht
op gezag van de rector magnificus, prof.dr. J.C. Stoof, ingevolge het besluit
van het college voor promoties in het openbaar te verdedigen op
maandag 21 april 2008 des middags te 4.15 uur door

Jeanne Henriëtte Bolt

geboren op 26 september 1962, te Roermond

ISBN 978-90-393-47850

SIKS Dissertation Series No. 2008-08. The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The research reported in this thesis was supported by the Netherlands Organisation for Scientific Research (NWO).

Bayesian Networks: Aspects of Approximate Inference

Bayesiaanse Netwerken: Aspecten van Benaderend Rekenen
(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht
op gezag van de rector magnificus, prof.dr. J.C. Stoof, ingevolge het besluit
van het college voor promoties in het openbaar te verdedigen op
maandag 21 april 2008 des middags te 4.15 uur door

Jeanne Henriëtte Bolt

geboren op 26 september 1962, te Roermond

Promotor: prof.dr.ir. L.C. van der Gaag

Contents

Chapter 1

General Introduction

Bayesian networks [51] provide for a concise representation of probabilistic knowledge with respect to a given problem domain and can be used to derive the probabilities of events in this domain. A Bayesian network consists of a graphical structure capturing the probabilistic relationships between the variables of the domain, supplemented with quantitative information concerning the joint probability distribution over the variables. Especially in domains in which uncertainty is predominant, Bayesian networks are suited to solving problems. By now, networks have been developed for a range of areas such as medical diagnosis, traffic prediction, weather prediction and technical troubleshooting; some examples can be found in [9, 21, 30, 33, 39, 66, 69, 79]

A Bayesian network can, in theory, be used for computing any probability of interest for the modelled variables. The problem of establishing the (conditional) marginal probabilities for a single variable, which is known as probabilistic inference, is NP-hard in general, however [10]. Algorithms have been designed that solve the inference problem in time exponential in the treewidth of the graphical structure [20, 37, 51, 60]. Still, for larger densely connected networks inference may become infeasible. Also the problem of establishing approximate probabilities with guaranteed error bounds is NP-hard in general [12]. Nevertheless, various useful approximation algorithms have been designed, although their results are not guaranteed to lie within specific error bounds [32]. The two classes of approximate algorithms that include the most widely used approximation methods are the Monte Carlo algorithms and the variational methods. In Monte Carlo methods, such as Gibbs sampling and importance sampling approximations are obtained by sampling [7, 8]. In variational methods [31] the inference problem is viewed as an optimisation problem, that is, inference is viewed as finding the minimal free energy of a probability distribution. This minimal free energy now can be approximated by the minimal free energy of a distribution from a restricted family of related distributions for which the free energy is easier to establish. In a mean field approximation, for example, related distributions are considered in which all variables are independent. Another algorithm from this second class is the *loopy-propagation algorithm* [51]. In this algorithm, in fact, the required probability distribution is approximated by considering the related distributions in which the pairwise dependencies between variables represented by neighbouring nodes in the graphical structure are taken into account. The performance of this algorithm has been analysed extensively for undirected graphical models. However, only a few studies are available in which its performance in directed models is addressed analytically. In the second part of this thesis some of the particulars of the loopy-propagation algorithm when applied to a Bayesian network will be stated. First, however, the focus will be on *qualitative probabilistic networks*. A qualitative probabilistic network also models statistical knowledge with respect to some underlying problem domain and also comprises a directed graphical structure. However, a qualitative network only includes qualitative information of the distribution of its variables. A qualitative network can be used to infer qualitative information about the influence of the observation of a variable on the probability distribution over the other variables in the network. Due to their high abstraction level, however, qualitative networks tend to yield uninformative results. In the first part of this thesis the formalism of qualitative probabilistic networks is extended by the introduction of *situational signs*. This extension provides for a more informative result upon inference with a qualitative network.

To summarise, in this thesis insight is gained in the local behaviour of the loopy-propagation algorithm in Bayesian networks, thereby supplementing earlier work on loopy propagation in undirected networks. Also, inference in qualitative probabilistic networks is improved. At first sight, loopy propagation and probabilistic inference in qualitative networks show few similarities, other than their not resulting in exact numerical probabilities. The analyses in this thesis, however, reveal that the two methods for approximate inference do share some common ground. More specifically, the two concepts of additive synergy and product synergy which are commonly used in qualitative probabilistic networks, are shown to constitute important factors in the local behaviour of the loopy-propagation algorithm when applied to Bayesian networks.

This thesis is organised as follows. Chapter 2 introduces the relevant basic concepts. The thesis then is divided basically into two parts. The first part concerns probabilistic inference in qualitative probabilistic networks. After the introduction in Chapter 3, Chapter 4 introduces the notion of situational sign into the framework of qualitative probabilistic networks and adapts the sign propagation algorithm for inference with qualitative networks to render it applicable to networks with such signs. In Chapter 5, the practicability of the situational signs is investigated in a realistic setting. The first part is concluded with Chapter 6 which summarises the results and indicates directions for further research. Parts of these chapters were published in [3, 6]. The second part concerns the loopy-propagation algorithm for approximate inference with Bayesian networks. After the introduction in Chapter 7, the two different types of error that arise in the probabilities computed by this algorithm are introduced in Chapter 8. These errors are further investigated in the subsequent Chapters 9, 10 and 11. Chapter 12 summarises the results and indicates directions for further research. Parts of these chapters also appeared in [2, 4, 5].

Chapter 2

General Preliminaries

This section reviews the basic concepts that are relevant for this thesis. Further details can be found in introductory texts on Bayesian networks such as [29, 51]; this section is partly based on the introductions given in [52, 64].

2.1 Probability Theory and Graph Theory

This section briefly states some of the basic concepts of modern probability theory as founded by Kolmogorov in 1933 [34]; a modern overview of the field can be found in for example [28, 58].

Central in probability theory are stochastic variables which can take a value from a set of values, called the value space. In this thesis, only discrete value spaces will be considered. In statements involving stochastic variables and their values, the following notational conventions are used. Single variables are denoted by upper-case letters (A), possibly supplemented with a superscript (A^i). The values of a variable are indicated by subscripted lower-case letters (a_i); for a binary variable A, the value a_1 will often be written as a and the value a_2 as $\bar{a}$. Sets of variables are denoted by bold-face upper-case letters ($\mathbf{A}$) and a joint value assignment to such a set is indicated by a bold-face lower-case letter ($\mathbf{a}$). Where no operator is given, the conjunction is meant; for example, $a_i b_j$ has to be read as $a_i \wedge b_j$. The upper-case letter is also used to indicate the whole range of possible value assignments; for example, for the binary variables A and B, the notation A, B may be used to indicate $\{ab, a\bar{b}, \bar{a}b, \bar{a}\bar{b}\}$. A value assignment to a variable will often be abbreviated by the value itself; for example, $A = a_i$ may be indicated by a_i.

A *joint probability distribution* is a function $\Pr$ on the set of joint value assignments to a set of variables $\mathbf{V}$, that adheres to the following properties

- $\Pr(\mathbf{v}) \geq 0$, for any value assignment $\mathbf{v}$ to $\mathbf{V}$
- $\sum_{\mathbf{V}} \Pr(\mathbf{V}) = 1$, and
- for all $\mathbf{A}, \mathbf{B} \subseteq \mathbf{V}$ and any value assignment $\mathbf{a}$ to $\mathbf{A}$ and $\mathbf{b}$ to $\mathbf{B}$, if $\Pr(\mathbf{ab}) = 0$ then $\Pr(\mathbf{a} \vee \mathbf{b}) = \Pr(\mathbf{a}) + \Pr(\mathbf{b})$.

Let $\Pr$ be a joint probability distribution on a set of variables $\mathbf{V}$ and let $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$. For any combination of value assignments $\mathbf{x}$ and $\mathbf{y}$ with $\Pr(\mathbf{y}) > 0$, the *conditional probability* of $\mathbf{x}$

given $\mathbf{y}$, denoted $\Pr(\mathbf{x} \mid \mathbf{y})$, is defined as

$$\Pr(\mathbf{x} \mid \mathbf{y}) = \frac{\Pr(\mathbf{xy})}{\Pr(\mathbf{y})}$$

Throughout this thesis all specified conditional probabilities are assumed to be properly defined, that is, for each $\Pr(\mathbf{x} \mid \mathbf{y})$ it is implicitly assumed that $\Pr(\mathbf{y}) > 0$.

From the definitions of joint probability distribution and conditional probability, the chain rule, the conditioning property, the marginalisation property and Bayes' rule can be derived. The *chain rule* states that for any joint probability distribution $\Pr$ on a set of variables $\mathbf{V} = \{V^1, \ldots, V^n\}, n \geq 1$, for any value assignment $v^1_{i1}, \ldots, v^n_{in}$ to $\mathbf{V}$ the following property holds

$$\Pr(v^1_{i1} \ldots v^n_{in}) = \Pr(v^1_{i1} \mid v^2_{i2} \ldots v^n_{in}) \cdot \ldots \cdot \Pr(v^{n-1}_{i(n-1)} \mid v^n_{in}) \cdot \Pr(v^n_{in})$$

The *marginalisation property* states that, for any joint probability distribution $\Pr$ on a set of variables $\mathbf{V}$, for all subsets $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ for any value assignment $\mathbf{x}$ to $\mathbf{X}$

$$\Pr(\mathbf{x}) = \sum_{\mathbf{Y}} \Pr(\mathbf{xY})$$

For any proper subset $\mathbf{X}$ of $\mathbf{V}$, the probability $\Pr(\mathbf{x})$ is called a marginal probability. The *conditioning property* further states that for any value asssignment $\mathbf{x}$ to $\mathbf{X}$

$$\Pr(\mathbf{x}) = \sum_{\mathbf{Y}} \Pr(\mathbf{x} \mid \mathbf{Y}) \cdot \Pr(\mathbf{Y})$$

with $\mathbf{X}$ and $\mathbf{Y}$ as before. *Bayes' rule*, to conclude, states that for any value asssignment $\mathbf{x}$ to $\mathbf{X}$ and any value assignment $\mathbf{y}$ to $\mathbf{Y}$

$$\Pr(\mathbf{x} \mid \mathbf{y}) = \frac{\Pr(\mathbf{y} \mid \mathbf{x}) \cdot \Pr(\mathbf{x})}{\Pr(\mathbf{y})}$$

For a joint probability distribution $\Pr$, further the concept of *(conditional) (in)dependence* is defined. For any joint probability distribution $\Pr$ on a set of variables $\mathbf{V}$, the sets of variables $\mathbf{X}$ and $\mathbf{Y}$ are independent given $\mathbf{Z}$, with $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$, denoted as $I_{\Pr}(\mathbf{X},\mathbf{Z},\mathbf{Y})$, if for any value assignment $\mathbf{x}$ to $\mathbf{X}$, $\mathbf{y}$ to $\mathbf{Y}$ and $\mathbf{z}$ to $\mathbf{Z}$

$$\Pr(\mathbf{xy} \mid \mathbf{z}) = \Pr(\mathbf{x} \mid \mathbf{z}) \cdot \Pr(\mathbf{y} \mid \mathbf{z})$$

Otherwise, $\mathbf{X}$ and $\mathbf{Y}$ are dependent given $\mathbf{Z}$. For $\mathbf{Z} \neq \emptyset$, the (in)dependence is called conditional on $\mathbf{Z}$.

Also some concepts from graph theory that are relevant for this thesis are reviewed. More details can, for example, be found in [23].

An *undirected graph* G is a pair $G = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ is a finite set of nodes and $\mathbf{E}$ is a set of unordered pairs (V, W) with $V, W \in \mathbf{V}$, called edges. The nodes V and W are neighbours if $(V, W) \in \mathbf{E}$. A path from V^0 to V^k is an ordered sequence of nodes and edges

$V^0, E^1, V^1, \ldots, V^k$ with $(V^{i-1}, V^i) \in \mathbf{E}$ for all $i = 1, \ldots, k$; k is called the length of the path. A cycle is a path of length at least one from a node back to itself. An undirected graph is called acyclic if it does not contain any cycles.

A *directed graph* G is a pair $G = (\mathbf{V}, \mathbf{A})$ where $\mathbf{V}$ is a finite set of nodes and $\mathbf{A}$ is a set of ordered pairs (V, W) with $V, W \in \mathbf{V}$, called arcs. W is a parent of V if $(W, V) \in \mathbf{A}$; W is a child of V if $(V, W) \in \mathbf{A}$. The set of all parents of V is denoted as $\rho(V)$; the set of all children as $\sigma(V)$. W is an ancestor of V if it is an element of the reflexive and transitive closure of $\rho(V)$. W is a descendant of V if it is an element of the reflexive and transitive closure of $\sigma(V)$. The ancestors and descendants of V are denoted by $\rho^*(V)$ and $\sigma^*(V)$ respectively. A trail in G is a sequence of nodes and arcs $V^0, A^1, V^1, \ldots, V^k$, such that either $A^i = (V^i, V^{i-1}) \in \mathbf{A}$, or $A^i = (V^{i-1}, V^i) \in \mathbf{A}$ for all $i = 1, \ldots, k$; k is called the length of the trail. A loop is a trail of length at least one from a node back to itself. A loop will be called simple if none of its arcs is shared by another loop; otherwise, the loop will be called compound. A loop node with one or more arcs on the loop will be called a convergence node; the other loop nodes will be called inner nodes. A directed graph is acyclic if it does not include a loop in which either $A^i = (V^i, V^{i-1}) \in \mathbf{A}$ for every i or $A^i = (V^{i-1}, V^i)$ for every i. The graph is singly connected if it does not include any loops; otherwise, it is multiply connected.

2.2 Graphical Representations of Probabilistic Independence

In the field of probabilistic graphical models, graphs are used to represent probabilistic (in)dependence [11, 36]. In a graphical representation of probabilistic (in)dependence, the nodes in the graph represent stochastic variables and the links in the graph capture probabilistic (in)dependencies among these variables. To read the (in)dependencies off the graphical representation, a graphical criterion is used. For an undirected graph the concept of separation is used for this purpose; for a directed graph the concept of d-separation is applied. In the sequel the terms node and variable will be used interchangeably.

For an undirected graph $G = (\mathbf{V}, \mathbf{E})$, the set of nodes $\mathbf{Z}$ is said to *separate* the sets of nodes $\mathbf{X}$ and $\mathbf{Y}$, where $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$ are mutually disjoint, denoted as $\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle$, if for each $V^i \in \mathbf{X}$ and each $V^j \in \mathbf{Y}$ every path from V^i to V^j contains at least one node from $\mathbf{Z}$. The following example illustrates this separation criterion for undirected graphs.

Example 2.1 *Consider the network from Figure 2.1. In this network, for example, the nodes A and D are separated by the set of nodes $\{B, C\}$, that is, $\langle \{A\} \mid \{B, C\} \mid \{D\} \rangle$. The nodes A and D are not separated by either B or C.*

For an acyclic directed graph $G = (\mathbf{V}, \mathbf{A})$, a trail t in G is said to be *blocked* by the set of nodes $\mathbf{Z} \subseteq \mathbf{V}$ if one of the following conditions holds:

- arcs (V^1, V^2) and (V^2, V^3) are on t, and $V^2 \in \mathbf{Z}$;
- arcs (V^2, V^1) and (V^2, V^3) are on t, and $V^2 \in \mathbf{Z}$;
- arcs (V^1, V^2) and (V^3, V^2) are on t, and $\sigma^*(V^2) \cap \mathbf{Z} = \emptyset$.

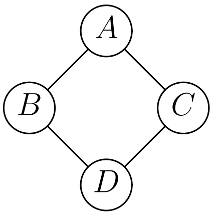


Figure 2.1: An example undirected graph.

For mutually disjoint sets of nodes $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$, the set of nodes $\mathbf{Z}$ is said to *d-separate* $\mathbf{X}$ and $\mathbf{Y}$ in G, denoted as $\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle^d$, if for each $V^i \in \mathbf{X}$ and each $V^j \in \mathbf{Y}$ every trail between V^i to V^j is blocked by $\mathbf{Z}$.

Example 2.2 *Consider the network from Figure 2.2. In this network, for example, the nodes A and D are d-separated by the set of nodes $\{B, C\}$, that is, $\langle \{A\} \mid \{B, C\} \mid \{D\} \rangle^d$. The nodes B and C on the other hand, are d-separated by node A, that is, $\langle \{B\} \mid \{A\} \mid \{C\} \rangle^d$. Nodes B and C are not d-separated by the set of nodes $\{A, D\}$, since with this set the trail $B \rightarrow D \leftarrow C$ is not blocked.*

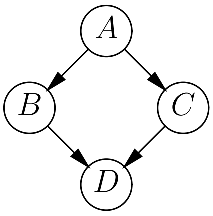


Figure 2.2: An example directed graph.

Ideally, a graph represents all dependencies and all independencies holding in a joint probability distribution, by means of the (d)-separation criterion. Unfortunately, there are probability distributions whose set of dependencies and independencies cannot be perfectly represented in a graph [51]. Graphical models of probabilistic (in)dependence now are mostly built in such a way that, if two nodes are (d-)separated in the graph, then the variables represented by these nodes are (conditionally) independent in the distribution at hand. The graph then is called an *independence map* for the distribution. An undirected graph G thus is called an undirected independence map for a joint probability distribution $\Pr$ if

$$\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle \Rightarrow \Pr(\mathbf{xy} \mid \mathbf{z}) = \Pr(\mathbf{x} \mid \mathbf{z}) \cdot \Pr(\mathbf{y} \mid \mathbf{z})$$

and an acyclic directed graph G is called an independence map for $\Pr$ if

$$\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle^d \Rightarrow \Pr(\mathbf{xy} \mid \mathbf{z}) = \Pr(\mathbf{x} \mid \mathbf{z}) \cdot \Pr(\mathbf{y} \mid \mathbf{z})$$

for any mutually disjoint sets of nodes $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$ and any value assignment $\mathbf{x}$ to $\mathbf{X}$, $\mathbf{y}$ to $\mathbf{Y}$ and $\mathbf{z}$ to $\mathbf{Z}$.

2.3 Probabilistic Models and Probabilistic Inference

A joint probability distribution over a set of stochastic variables $\mathbf{V}$ can be represented concisely by a graphical representation of its probabilistic independencies together with information about the strengths of the dependencies between neighbouring variables in the graph. Section 2.3.1 describes the representation of a joint probability distribution by a directed graph in a *Bayesian network* and Section 2.3.2 gives the representation of a joint probability distribution by an undirected graph in a *Markov network*. Section 2.3.3 to conclude outlines how a Bayesian network can be abstracted into a qualitative probabilistic network. A network that represents just binary variables will be called a binary network.

A probabilistic model is often used to find the probability distribution for a single variable, possibly given some observations for the other variables in the model. In this thesis, the term *probabilistic inference*, is used to refer to the computation of the marginal probability distributions for the individual variables in a network. The complexity of probabilistic inference is NP-hard [10] in general. Inference in a Bayesian network or a Markov network, however, may be feasible, because the independencies which are reflected in the graph can be employed to simplify the computations. In fact, inference can be solved in time exponential in the treewidth [20]. Roughly speaking, the more independencies are reflected in a network's graph, that is, the sparser its graph, the lower its treewidth will tend to be. Pearl [51] developed an efficient algorithm for exact inference in singly-connected Bayesian networks. This algorithm is reviewed in the next section; in Section 2.3.2 an equivalent algorithm for Markov networks [71] is outlined. Although designed for singly connected networks, these algorithms appeared to be very useful as approximate algorithms for multiply connected networks. Pearl's propagation algorithm can also be used for exact reasoning with multiply connected networks. Then, however, a repeated application of the algorithm for a single problem is required [51]. For exact reasoning with networks of arbitrary topology, however, the so-called junction tree algorithm shows a more favourable runtime complexity. A junction tree algorithm can be viewed as a generalised version of the algorithm discussed in Section 2.3.2. For further details of this algorithm, the reader is referred to for example [20, 26, 37, 60]. Section 2.3.3, to conclude describes inference in qualitative probabilistic networks [14]. The complexity of qualitative probabilistic inference is linear in the number of arcs, however, obviously with qualitative networks only qualitative results can be obtained.

2.3.1 Bayesian Networks

A Bayesian network captures a joint probability distribution $\Pr$ over a set of variables $\mathbf{V} = \{V^1, \ldots, V^n\}, n \geq 1$ by an acyclic directed graph and an associated set of conditional probability distributions. The graph is a directed independence map of the joint probability distribution; the strengths of the dependencies between the variables are captured by the conditional probability distributions $\Pr(V^i \mid \rho(V^i))$, specified for all nodes V^i, where $\rho(V^i)$ denotes the joint value assignments of the parents of V^i. The joint probability distribution now is factorised as

$$\Pr(\mathbf{V}) = \prod_i \Pr(V^i \mid \rho(V^i))$$

The following example illustrates this factorisation.

Example 2.3 *Consider the small Bayesian network from Figure 2.3. For the probability distribution represented by this network it holds, for example, that* $\Pr(a\bar{b}\bar{c}d) = \Pr(a \mid d) \cdot \Pr(\bar{b} \mid d) \cdot \Pr(\bar{c} \mid a\bar{b}) \cdot \Pr(d) = y \cdot (1-p) \cdot (1-s) \cdot x$.

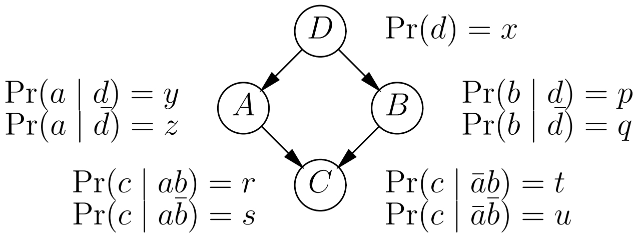

D
$\Pr(d) = x$
$\Pr(a \mid d) = y$
$\Pr(a \mid \bar{d}) = z$
A
B
$\Pr(b \mid d) = p$
$\Pr(b \mid \bar{d}) = q$
$\Pr(c \mid ab) = r$
$\Pr(c \mid a\bar{b}) = s$
C
$\Pr(c \mid \bar{a}b) = t$
$\Pr(c \mid \bar{a}\bar{b}) = u$

Figure 2.3: An example Bayesian network.

Since a probability distribution is fully described by a Bayesian network, any probability of interest can be computed from the network by building upon the distribution's factorisation and using the properties of marginalisation and conditioning. Such an approach would be highly demanding from a computational point of view and more efficient algorithms have been designed. Here Pearl's propagation algorithm [51] is briefly reviewed. This algorithm is designed for exact probabilistic inference with singly connected Bayesian networks. In the algorithm, each node $X \in \mathbf{V}$ is provided with a limited set of rules that enable the node to compute its marginal probability distribution $\Pr(X \mid \mathbf{e})$ given the available evidence $\mathbf{e}$, from messages it receives from its neighbours. The messages that a node receives from its parents are called *causal messages* and the messages that a node receives from its children are called *diagnostic messages*. A node uses the causal messages that it receives to compute its *compound causal parameter* and it uses the diagnostic messages it receives to compute its *compound diagnostic parameter*. A node combines its compound parameters by the *data fusion rule* to obtain its marginal distribution. The rules of the algorithm are applied in parallel by the various nodes at each time step. The data fusion rule used by node X for establishing the probability distribution $\Pr(X \mid \mathbf{e})$ at time t is

$$\Pr^t(X \mid \mathbf{e}) = \alpha \cdot \lambda^t(X) \cdot \pi^t(X)$$

where the compound diagnostic parameter $\lambda^t(X)$ is computed from the diagnostic messages $\lambda^t_{Y^j}(X)$ it receives from each of its children Y^j

$$\lambda^t(X) = \prod_j \lambda^t_{Y^j}(X)$$

and the compound causal parameter $\pi^t(X)$ is computed from the causal messages $\pi^t_X(U^i)$ it receives from each of its parents U^i

$$\pi^t(X) = \sum_{\mathbf{U}} \Pr(X \mid \mathbf{U}) \cdot \prod_i \pi^t_X(U^i)$$

where $\mathbf{U}$ denotes the set of all parents of node X. The rule for computing the diagnostic messages to be sent to its parent U^i is

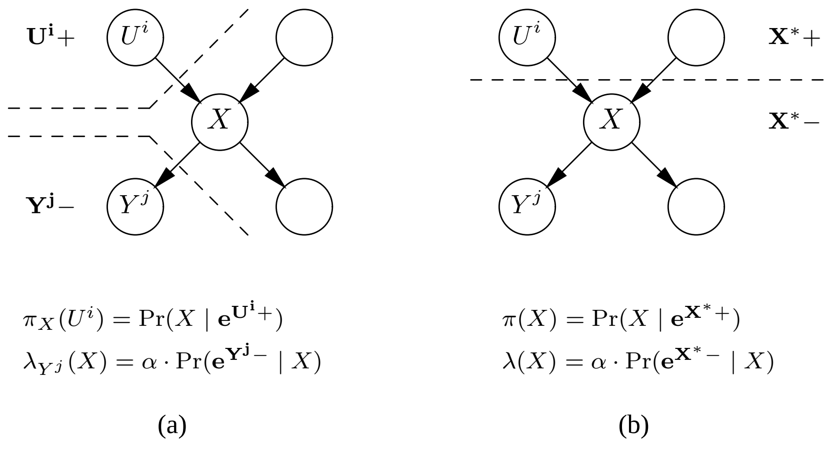
$$\lambda_X^{t+1}(U^i) = \alpha \cdot \sum_X \lambda^t(X) \cdot \sum_{\mathbf{U}/\{U^i\}} \Pr(X \mid \mathbf{U}) \cdot \prod_{k \neq i} \pi_X^t(U^k)$$

and the rule for computing the causal messages to be sent to its child Y^j is

$$\pi_{Y^j}^{t+1}(X) = \alpha \cdot \pi^t(X) \cdot \prod_{k \neq j} \lambda_{Y^k}^t(X)$$

In the above computation rules, α denotes a normalisation constant. All messages are initialised to contain just 1s. An observation for a node X is entered into the network by multiplying the components of $\lambda^0(X)$ and $\pi_{Y^j}^1(X)$ by 1 for the observed value of X and by 0 for the other value(s).

In a singly connected network, after a limited number of time steps, proportional to the diameter of the graph, the parameters will not change any more and an equilibrium state is reached. The rules of Pearl's algorithm are constructed in such a way that at this equilibrium a causal message $\pi_X(U^i)$ equals $\Pr(X \mid \mathbf{e}^{\mathbf{U^i}+})$ where $\mathbf{e}^{\mathbf{U^i}+}$ denotes the observations found in the subgraph $\mathbf{U^i}+$, as indicated in Figure 2.4(a) and a diagnostic message $\lambda_{Y^j}(X)$ equals $\alpha \cdot \Pr(\mathbf{e}^{\mathbf{Y^j}-} \mid X)$ where $\mathbf{e}^{\mathbf{Y^j}-}$ denotes the observations found in the subgraph $\mathbf{Y^j}-$ as indicated in the same figure. Furthermore, the compound causal parameter $\pi(X)$ equals $\Pr(X \mid \mathbf{e}^{\mathbf{X}^*+})$, where $\mathbf{e}^{\mathbf{X}^*+}$ denotes the observations found in the subgraph $\mathbf{X}^*+$, as indicated in Figure 2.4(b) and the compound causal parameter $\lambda(X)$ equals $\alpha \cdot \Pr(\mathbf{e}^{\mathbf{X}^*-} \mid X)$, where $\mathbf{e}^{\mathbf{X}^*-}$ denotes the observations found in the subgraph $\mathbf{X}^*-$ as indicated in the same figure. Note that node X itself is included in $\mathbf{X}^*-$. Finally, $\Pr(X \mid \mathbf{e}) = \alpha \cdot \lambda(X) \cdot \pi(X)$ at equilibrium.



$\pi_X(U^i) = \Pr(X \mid \mathbf{e}^{\mathbf{U^i}+})$
$\lambda_{Y^j}(X) = \alpha \cdot \Pr(\mathbf{e}^{\mathbf{Y^j}-} \mid X)$

(a)

$\pi(X) = \Pr(X \mid \mathbf{e}^{\mathbf{X}^*+})$
$\lambda(X) = \alpha \cdot \Pr(\mathbf{e}^{\mathbf{X}^*-} \mid X)$

(b)

Figure 2.4: The subgraphs $\mathbf{U^i}+$ and $\mathbf{Y^j}-$ (a) and the subgraphs $\mathbf{X}^*+$ and $\mathbf{X}^*-$ (b).

Note that, the number of elements in a message which a node sends to a child equals the number of its own values, and the number of elements in a message that a node sends to a parent equals the number of values of this parent. For a binary parent U^i of X, the diagnostic message sent by X may be written as $(\lambda_X(u^i), \lambda_X(\bar{u}^i))$; for a binary child Y^j of X, the causal message sent by X may be written as $(\pi_{Y^j}(x), \pi_{Y^j}(\bar{x}))$. From here on, diagnostic and causal messages may also be termed message vectors.

2.3.2 Markov Networks

A Markov network captures a joint probability distribution $\Pr$ over a set of variables $\mathbf{V}$ by an undirected graph and an associated set of clique potentials. The graph is an undirected independence map of the joint probability distribution; the strengths of the dependencies between the variables are captured by the clique potentials. A clique in the graph is a set of nodes $\mathbf{C^i} \subseteq \mathbf{V}$ such that there is an edge (V^j, V^k) between any pair of nodes $V^j, V^k \in \mathbf{C^i}$; the union of all cliques equals $\mathbf{V}$. For each clique the network specifies a potential function $\psi(\mathbf{C^i})$ which assigns a non-negative real number to each configuration of $\mathbf{C^i}$. The joint probability now is factorised as

$$\Pr(\mathbf{V}) = \frac{1}{Z} \cdot \prod_i \psi(\mathbf{C^i})$$

where $Z = \sum_{\mathbf{V}} \prod_i \psi(\mathbf{C^i})$ is a normalising factor to ensure that $\sum_{\mathbf{V}} \Pr(\mathbf{V}) = 1$.

In the sequel, the focus will be mainly on pairwise Markov networks which are Markov networks with cliques of maximal two nodes only. In a pairwise Markov network, the transition matrices M^{AB} and M^{BA} can be associated with the edge (A, B) between two neighbouring nodes A and B.

$$M^{AB}_{ji} = \psi(A = a_i, B = b_j)$$

Note that the matrix M^{BA} equals M^{AB^T}. The matrices M^{AB} and M^{BA} are called the transition matrices of the edge (A, B).

Example 2.4 *Consider a Markov network consisting of the nodes A and B with the potentials $\psi(ab) = p$, $\psi(a\bar{b}) = q$, $\psi(\bar{a}b) = r$ and $\psi(\bar{a}\bar{b}) = s$, as depicted in Figure 2.5. Then the transition matrix $M^{AB} = \begin{bmatrix} p & r \\ q & s \end{bmatrix}$ is associated with the link from A to B and its transpose $M^{BA} = \begin{bmatrix} p & q \\ r & s \end{bmatrix}$ is associated with the link from B to A.*

For pairwise Markov networks, an inference algorithm is available which is functionally equivalent to Pearl's propagation algorithm [71]. In this algorithm, in each time step, every node sends a message vector to each of its neighbours. The message from a node A to its neighbour B equals $M^{AB} \cdot m$ after normalisation, where m is the vector that results from the component wise multiplication of all message vectors sent to A except for the message vector sent by B. The marginal probability distribution of a node is obtained by combining, at the equilibrium state, all incoming messages again by component-wise multiplication and normalising the resulting vector. Using $\odot$ as symbol for component wise vector multiplication, the algorithm can more formally be defined as below. Given a node X with the neighbours U^i the marginal probabilities computed for a node X at time t equal

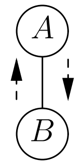
$$\Pr^t(X \mid \mathbf{e}) = \alpha \cdot \left(\odot_i \; m_{U^i}(X)^t\right)$$

where $m_{U^i}(X)^t$ denotes the message vector from U^i to X at time t and α denotes a normalisation constant. The message vector from a node X to a neighbour U^k at time $t + 1$ equals

$$m_X(U^k)^{t+1} = \alpha \cdot M^{XU^k} \cdot \left(\odot_{i \neq k} \; m_{U^i}(X)^t\right)$$

where M^{XU^k} denotes the transition matrix from X to U^k. The procedure is initialised with all message vectors set to (1,1,...,1). Observed nodes always send a vector with 1 for the observed value and zero for all other values to any of their neighbours.

$$\begin{aligned}\psi(ab) &= p\\ \psi(a\bar{b}) &= q\\ \psi(\bar{a}b) &= r\\ \psi(\bar{a}\bar{b}) &= s\end{aligned} \qquad \begin{bmatrix} p & q \\ r & s \end{bmatrix}$$



$$\begin{bmatrix} p & r \\ q & s \end{bmatrix}$$

Figure 2.5: An example pairwise Markov network and its transition matrices.

2.3.3 Qualitative Probabilistic Networks

A *qualitative probabilistic network* is essentially a qualitative abstraction of a Bayesian network. It equally comprises an acyclic directed graph modelling the variables of a joint probability distribution and the probabilistic independencies between them. Instead of conditional probability distributions, however, a qualitative probabilistic network associates with its digraph *qualitative influences* and *qualitative synergies*. These influences and synergies capture qualitative features of the represented distribution [48, 74]. In the sequel only qualitative probabilistic networks with just binary variables are considered. The concepts of qualitative influence and of qualitative synergies can be generalised to apply to non-binary variables as well, however, by building on the concept of first order stochastic dominance [13, 74].

A *qualitative influence* between two variables expresses how observing a value for the one variable affects the probability distribution over the values of the other variable. For example, a positive qualitative influence of a parent A on its child B, denoted $S^+(A, B)$, is found when observing the higher value for node A makes the higher value for its child B more likely, regardless the state of the network, that is, a positive qualitative influence of A on B is found if

$$\Pr(b \mid a\mathbf{x}) - \Pr(b \mid \bar{a}\mathbf{x}) \geq 0$$

for any combination of values $\mathbf{x}$ for the set $\mathbf{X} = \rho(B)\backslash\{A\}$ of other parents of B. The '$+$' in the notation $S^+(A, B)$ is termed the *sign* of the influence. A negative qualitative influence, denoted S^-, and a zero qualitative influence, denoted S^0, are defined analogously, replacing $\geq$ in the above formula by $\leq$ and $=$, respectively. For a positive, negative or zero influence of A on B, the difference $\Pr(b \mid a\mathbf{x}) - \Pr(b \mid \bar{a}\mathbf{x})$ has the same sign for all combinations of values $\mathbf{x}$. These influences thus describe a *monotonic* effect of a change in A's distribution on the probability distribution over B's values. If the influence of A on B is positive for one combination $\mathbf{x}$ and negative for another combination, however, the influence is *non-monotonic*. Non-monotonic influences are associated with the sign '$?$', indicating that their effect is ambiguous.

Qualitative influences exhibit various properties that are important for inference purposes [14, 74]. The property of *symmetry* states that, if the network includes the influence $S^\delta(A, B)$, then it also includes $S^\delta(B, A)$, $\delta \in \{+, -, 0, ?\}$. The *transitivity* property asserts that the qualitative influences along a trail that specifies at most one incoming arc for each variable, combine into a net influence whose sign is defined by the $\otimes$-operator from Table 2.1. The property of *composition* asserts that multiple influences between two variables along parallel trails, each specifying at most one incoming arc per variable, combine into a net influence whose sign is defined by the $\oplus$-operator. The three properties with each other provide for establishing the sign of the net influence between any two variables in a qualitative network.

Table 2.1: The $\otimes$- and $\oplus$-operators for combining signs.

$\otimes$	$+$	$-$	0	$?$
$+$	$+$	$-$	0	$?$
$-$	$-$	$+$	0	$?$
0	0	0	0	0
$?$	$?$	$?$	0	$?$

$\oplus$	$+$	$-$	0	$?$
$+$	$+$	$?$	$+$	$?$
$-$	$?$	$-$	$-$	$?$
0	$+$	$-$	0	$?$
$?$	$?$	$?$	$?$	$?$

For a qualitative probabilistic network furthermore additive synergies and product synergies can be specified. These synergies capture the joint interactions among three variables. The *additive synergy* expresses how two nodes interact in each other's influence on the probability distribution over a third node. For example, a positive additive synergy of the parents A and B with respect to their common child C, denoted $Y^{+}(\{A, B\}, C)$, is found when the parents strengthen each other's influence, that is, when

$$\Pr(c \mid ab\mathbf{x}) - \Pr(c \mid a\bar{b}\mathbf{x}) - \Pr(c \mid \bar{a}b\mathbf{x}) + \Pr(c \mid \bar{a}\bar{b}\mathbf{x}) \geq 0$$

for any combination of values $\mathbf{x}$ for the set $\mathbf{X} = \rho(B)\backslash\{A\}$ of other parents of C. A negative additive synergy, denoted Y^{-}, and a zero additive synergy, denoted Y^{0}, are defined analogously, replacing $\geq$ in the above formula by $\leq$ and $=$, respectively. A non-monotonic additive synergy of the variables A and B with respect to C is denoted $Y^{?}(\{A, B\}, C)$.

A *product synergy* [48] also concerns the interaction between three nodes A, B and C. It expresses how, if a common descendant C of A and B, is observed, its influence on the one of its ancestors would change after a change in the probability distribution over its other ancestor. Since the strength of the influence of node C on, for example, node B depends on the probability distribution over node A the original strength of the influence of the observation of C on B will change after a change in the distribution over A. Effectively, therefore, an (additional) influence of node A on node B is induced after the observation of C. The (additional) influence that is induced between two nodes by the observation of a third node is also termed the *intercausal influence*. The sign of the product synergy, or intercausal influence, between nodes A and B given the observation $C = c_i$ for a common descendant equals the sign of $\Pr(b \mid ac_i) - \Pr(b \mid \bar{a}c_i)$, considering A and B independent if C is unobserved. To a child with exactly two uninstantiated ancestors the definition of *product synergy I* can be applied to derive the sign of the product synergy. For example, a positive product synergy, between the parents A and B with respect to the observation of $C = c_i$ for their common child C, and given the instantiation $\mathbf{x}$ for the set $\mathbf{X}\backslash\{A, B\}$ of other parents of C, denoted $X^{+}(\{A, B\}, c_i)$ is found if

$$\Pr(c_i \mid ab\mathbf{x}) \cdot \Pr(c_i \mid \bar{a}\bar{b}\mathbf{x}) - \Pr(c_i \mid a\bar{b}\mathbf{x}) \cdot \Pr(c_i \mid \bar{a}b\mathbf{x}) \geq 0$$

Note that the product synergy pertains to a specific value of the child node C; the additional influence between A and B after the observation of node C depends on the observed value for C. Note furthermore that product synergy I is always monotonic. A negative product synergy, denoted X^-, and a zero additive synergy, denoted X^0, are defined analogously with replacing $\geq$ in the above formula by $\leq$ and $=$ respectively.

For a child node with more than two uninstantiated parents, unfortunately, it is less easy to retrieve the sign for the product synergy. Recall that for the additive synergy the correct sign could be established by comparing the signs given the different instantiations for the node(s) $\mathbf{X}$. In the case of the product synergy, however, it may be that the signs for all different value combinations for $\mathbf{X}$ are the same, yet for an 'intermediate' probability distribution over the nodes $\mathbf{X}$ an opposite sign is found [15]. For more than two uninstantiated parents, the definition of *product synergy II* can be applied to establish the sign of the product synergy; further details can be found in [15].

The various qualitative influences and synergies are illustrated by the following example.

Example 2.5 *Consider the small Bayesian network from Figure 2.6. The network represents a fragment of fictitious knowledge about the effect of training and fitness on one's feeling of well-being. Variable F captures one's fitness and variable T models whether or not one has undergone a training session; variable W models whether or not one has a feeling of well-being. All variables are binary, with the values* yes $>$ no. *From this network a qualitative abstraction can be constructed. For the conditional probability distributions specified for the variable W, the properties $\Pr(w \mid ft) - \Pr(w \mid \bar{f}t) \geq 0$ and $\Pr(w \mid f\bar{t}) - \Pr(w \mid \bar{f}\bar{t}) \geq 0$ hold and thus it is found that $S^+(F,W)$; fitness favours a feeling of well-being regardless of training. It is further found that $\Pr(w \mid ft) - \Pr(w \mid f\bar{t}) > 0$ and that $\Pr(w \mid \bar{f}t) - \Pr(w \mid \bar{f}\bar{t}) < 0$ and thus that $S^?(T,W)$; the effect of a training session on one's well-being depends on one's fitness. From $\Pr(w \mid ft) + \Pr(w \mid \bar{f}\bar{t}) \geq \Pr(w \mid f\bar{t}) + \Pr(w \mid \bar{f}t)$, to conclude, it follows that $Y^+(\{F,T\},W)$. The resulting qualitative network is shown in Figure 2.7; the signs of the qualitative influences are shown along the arcs, and the sign of the additive synergy is indicated over the curve over the variable W.*

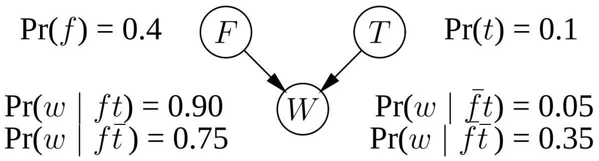

Pr(f) = 0.4
F
T
Pr(t) = 0.1
Pr($w \mid ft$) = 0.90
Pr($w \mid f\bar{t}$) = 0.75
W
Pr($w \mid \bar{f}t$) = 0.05
Pr($w \mid \bar{f}\bar{t}$) = 0.35

Figure 2.6: An example Bayesian network, modelling the effects of fitness (F) and training (T) on a feeling of well-being (W).

For probabilistic inference with a qualitative network, an efficient algorithm is available [14]. This algorithm provides for computing the effect of an observation that is entered into the network, upon the probability distributions over the other variables. It is based on the idea of propagating and combining signs, and builds upon the properties of symmetry, transitivity and composition of qualitative influences. The algorithm is summarised in pseudo-code in Figure 2.8. The algorithm takes for its input a qualitative probabilistic network (Q), a variable for which an

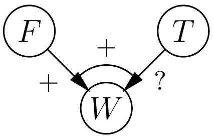


Figure 2.7: The qualitative abstraction of the Bayesian network from Figure 2.6.

observation has become available (O), and the sign of this observation (*sign*), that is, either a '$+$' for the observation $O = o_1$ or a '$-$' for the observation $O = o_2$. The algorithm now traces the effect of the observation throughout the network, by passing messages between neighbouring variables. For each variable, it determines a node sign (sign[.]) that indicates the direction of change in probability distribution that is occasioned by the observation; initially all node signs are set to '0'. The actual inference is started by the observed variable receiving a message (*message*) with the sign of the observation. Each variable that receives a message, updates its node sign using the $\oplus$-operator and subsequently sends a message to each active neighbour; a neighbour is active if it is not blocked from the observed variable along the trail (*trail*) that is currently being followed. The sign of the message that the variable sends to a neighbour, is the $\otimes$-product of its own node sign and the sign of the influence that the message will traverse (*linksign*). This process of message passing between neighbouring variables is repeated iteratively. A trail of messages ends as soon as there are no more active neighbours to visit or as soon as the current message does not change the node sign of the visited variable. Since the node sign of each variable can change at most twice, once from '0' to '$+$', '$-$' or '$?$', and then only to '$?$', the process visits each variable at most twice and is guaranteed to halt in polynomial time.

```
procedure Process-Observation($Q$,$O$,sign):

  for all $A^i \in V(G)$ in $Q$
  do sign[$A^i$] $\leftarrow$'0';
  Propagate-Sign($Q$,$\emptyset$,$O$,sign).

procedure Propagate-Sign($Q$,trail,$to$,message):

  sign[$to$] $\leftarrow$ sign[$to$] $\oplus$ message;
  trail $\leftarrow$ trail $\cup$ $\{to\}$;
  for each active neighbour $A^i$ of $to$ in $Q$
  do  linksign $\leftarrow$ sign of influence between $to$ and $A^i$;
      message $\leftarrow$ sign[$to$] $\otimes$ linksign;
      if $A^i \notin$ trail and sign[$A^i$] $\neq$ sign[$A^i$] $\oplus$ message
      then Propagate-Sign($Q$,trail,$A^i$,message).
```

Figure 2.8: The basic sign-propagation algorithm.

The sign-propagation algorithm reviewed above serves to compute the effect of a *single* observation on the marginal distributions of *all* other variables in a qualitative network. In realistic applications, often the effect of *multiple* simultaneous observations on a single variable is of interest. This joint effect can be computed as the $\oplus$-sum of the effects of each of the separate observations on the variable of interest [13]. A more elaborate algorithm that prevents unnecessary uninformative results may also be applied for this purpose [56].

Example 2.6 *Consider the qualitative probabilistic network from Figure 2.9 and suppose that the evidence $E = e_2$ has been entered. Prior to the propagation, the node signs of all variables are set to '0'. The actual inference is started by entering the observation into the network, that is, by sending the message '$-$' to the variable E. E updates its node sign to $0 \oplus - = -$ and subsequently sends the message $- \otimes + = -$ to its neighbour C. Upon receiving this message, C updates its node sign to $0 \oplus - = -$. It subsequently sends the messages $- \otimes - = +$ to B, $- \otimes ? = ?$ to A, and $- \otimes + = -$ to F. The variables B, A and F then update their node signs to $0 \oplus + = +$, $0 \oplus ? = ?$, and $0 \oplus - = -$, respectively. Variable B sends the message $+ \otimes + = +$ to D. The inference ends after D has updated its node sign to $0 \oplus + = +$. The resulting node signs are shown in the figure.*

Now suppose that the joint effect of the simultaneous observations $D = d_1$ and $E = e_2$ on the variable A has to be assessed. The effect of the observation $E = e_2$ on the probability distribution for A is ambiguous, as illustrated above. The effect of the observation $D = d_1$ on the probability distribution over A is also determined from the initial *state of the network. The inference runs comparably, except that the variable A is blocked from the observed variable D: upon receiving its message from B, variable C does not send any information to A. Propagation of the observation $D = d_1$ thus results in the node sign '0' for variable A. The combined effect of the two observations on A is $? \oplus 0 = ?$.*

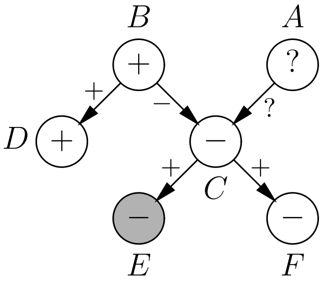


Figure 2.9: A qualitative network and its node signs after propagating the observation $E = e_2$.

Part I

Situational Signs

A qualitative probabilistic network is a graphical model of the probabilistic influences among a set of stochastic variables, in which each influence is associated with a qualitative sign. A non-monotonic influence between two variables is associated with the ambiguous sign '?', which indicates that the actual sign of the influence depends on the state of the network. The presence of such ambiguous signs is undesirable as it tends to lead to uninformative results upon inference. In this part of the thesis it is argued that, although a non-monotonic influence may have varying effects, in each specific state of the network, its effect is unambiguous. To capture the current effect of the influence, the concept of *situational sign* is introduced. It is shown how situational signs can be used upon inference and how they are updated as the state of the network changes. By means of a real-life qualitative network in oncology it is demonstrated that the use of situational signs can effectively forestall uninformative results upon inference.

Chapter 3

Introduction

In Chapter 2 Bayesian networks and their qualitative abstractions *qualitative probabilistic networks* were reviewed. Qualitative networks model the probabilistic relationships between their variables at a higher abstraction level than Bayesian networks and as a consequence, inference with such a network can lead to results that are not informative. Obviously in the application of a qualitative network it is desirable that inference yields results that are as informative as possible. In the past decade, therefore, various researchers have addressed the tendency of qualitative networks to yield uninformative results upon inference, and have proposed extensions to the basic formalism. These extensions include the exploitation of the relative strength of influences, the inclusion of the possibility to revert to quantitative reasoning for resolving trade-offs whenever necessary and the exploitation of the fact that ambiguous signs may become unambiguous given particular observations [38, 50, 53, 55–57]. Qualitative networks can, for example, be exploited in the construction of a fully quantified Bayesian network [52, 54]. For Bayesian networks, the usually large number of probabilities required tends to be a major obstacle to their construction [16, 65]. By using the intermediate step of building a qualitative network, the reasoning behaviour of the Bayesian network can be studied and validated prior to probability assessment. The signs of the validated qualitative network, moreover, can be used as constraints on the probabilities to be obtained for the Bayesian network, thereby simplifying the quantification task.

In this part of the thesis, a new extension of the framework of qualitative probabilistic networks is proposed which enhances the informative results upon inference. Closely linked with the high abstraction level of representation in qualitative probabilistic networks is the issue of non-monotonicity. An influence of a variable A on a variable C is called non-monotonic if it is positive in one state and negative in another state of the network under consideration. A non-monotonic influence cannot be assigned an unambiguous sign of general validity and is associated with the uninformative ambiguous sign '$?$'. Although a non-monotonic influence may have varying effects, in each particular state of the network its effect is unambiguous. The framework of qualitative probabilistic networks now will be extended with signs that capture information about the current effect of non-monotonic influences. These signs are termed *situational signs* to express that they are dynamic and valid only in particular states of the network. It is shown how situational signs can be used and updated upon inference and how they may forestall uninformative results.

To investigate the practicability of situational signs, the effect of their introduction into a real-life qualitative network in the field of oesophageal cancer is studied. The difference in performance between the qualitative network with ambiguous signs for its non-monotonic influences and the same network in which these ambiguous signs have been supplemented with situational signs is established. It is demonstrated that the situational network tends to yield more informative results upon inference than the original network.

This part is organised as follows. Chapter 4 introduces the concept of situational sign, details the dynamics of situational signs and gives an adapted algorithm for inference with a situational qualitative network. In Chapter 5 the effect of introducing situational signs upon the performance of the qualitative oesophageal cancer network is described and the part ends with some concluding observations in Chapter 6.

Chapter 4

Situational Signs

The presence of influences with ambiguous signs in a qualitative probabilistic network is likely to give rise to ambiguous, and therefore uninformative, results upon inference, as was illustrated in Section 2. From the definitions of the $\otimes$- and $\oplus$-operators, moreover, it follows that, once an ambiguous sign is encountered upon inference, it tends to spread throughout the network. The use of ambiguous signs to indicate non-monotonicity thus has undesirable consequences. Section 4.1 now introduces the concept situational sign as an alternative. With situational signs ambiguous results upon inference may be forestalled. Section 4.2 details the dynamics of situational signs and gives an adapted algorithm for inference with a situational qualitative network.

4.1 Defining Situational Signs

The ambiguous sign of a non-monotonic influence has its origin in the fact that, for a qualitative influence of a node A on a node C along an arc $A \rightarrow C$ to be unambiguous, the difference $\Pr(c \mid a\mathbf{x}) - \Pr(c \mid \bar{a}\mathbf{x})$ has to have the same sign for *all* combinations of values $\mathbf{x}$ for the set $\mathbf{X} = \rho(C) \setminus \{A\}$ of parents of C other than A. This sign then is valid for any probability distribution over $\mathbf{X}$ and hence, in all possible states of the network under study, that is, given any (possibly empty) set of observations for the network's nodes. If the difference $\Pr(c \mid a\mathbf{x}) - \Pr(c \mid \bar{a}\mathbf{x})$ yields contradictory signs for different combinations $\mathbf{x}$, then the sign of the influence is dependent upon the probability distribution over $\mathbf{X}$ and can differ for different states of the network. The influence then is assigned the ambiguous sign '$?$'.

Now consider a non-monotonic qualitative influence of A on C along the arc $A \rightarrow C$. Although the influence is non-monotonic, observing the higher value for A cannot make the higher value for C more likely and less likely at the same time. In each specific state of the network the influence of A on C will be either positive, negative or zero. In each specific state of the network, therefore, the effect of the influence of A on C is unambiguous. To capture information about the current effect of a non-monotonic influence, associated with a specific state, the concept of *situational sign* is introduced into the formalism of qualitative probabilistic networks. A *positive situational sign* for the influence of a node A on a node C along an arc $A \rightarrow C$ indicates that

- $S^?(A, C)$ and

- $\sum_{\mathbf{X}} \big(\Pr(c \mid a\mathbf{X}) - \Pr(c \mid \bar{a}\mathbf{X}) \big) \cdot \Pr(\mathbf{X} \mid \mathbf{e}) \geq 0$,

where $\mathbf{X} = \rho(C) \setminus \{A\}$ and $\mathbf{e}$ is the entered evidence. Negative, zero and unknown situational signs are defined analogously. Note that, while the regular signs of qualitative influences and additive synergies have general validity, a situational sign is dynamic in nature: it pertains to a specific state of a network and may lose its validity as the network's state changes. An influence with a situational sign δ now is called a *situational influence*; the sign of this situational influence is denoted '$?(\delta)$'. A qualitative probabilistic network with situational signs is termed a *situational qualitative network*. Note furthermore that, since the situation sign pertains to the influence between A and C in isolation, in the computation of the situational sign just the local conditional probabilities of C given its parents are used. To further elaborate on this observation the example network from Figure 4.1 is considered.

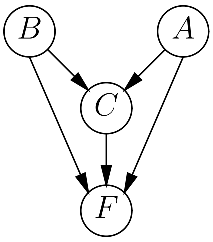


Figure 4.1: An example network with a node C with the parents A and B, and a child F.

Suppose that the influence of A on C over the arc $A \rightarrow C$ is non-monotonic. After, for example, the observation $F = f$ has been entered into the network, the situational sign of the influence between A and C is captured by $\sum_B \big(\Pr(c \mid aB) - \Pr(c \mid \bar{a}B) \big) \cdot \Pr(B \mid f)$ rather than by $\sum_B \big(\Pr(c \mid aBf) - \Pr(c \mid \bar{a}Bf) \big) \cdot \Pr(B \mid f)$. Compared to the second expression, the first expression does not incorporate the direct influence of the observation of F on the probabilities for node C. It thus pertains to just the influence over the arc $A \rightarrow C$. Note that upon inference, the sign-propagation algorithm would provide for the combination of all separate influences into the overall influence.

The following example illustrates the concept of situational signs.

Example 4.1 *Consider again the Bayesian network from Figure 2.6 and its qualitative abstraction from Figure 2.7 as presented in Chapter 2. Recall that the qualitative influence of T on W was found to be non-monotonic. The effect of this influence therefore depends on the state of the network. In the prior state of the network $\Pr(f) = 0.4$. Given this probability is found that $\Pr(w \mid t) = 0.39$ and that $\Pr(w \mid \bar{t}) = 0.51$. From the difference $\Pr(w \mid t) - \Pr(w \mid \bar{t}) = -0.12$ being negative now can be concluded that, in this particular state, the influence of T on W is negative; in one's current state of fitness, a training session has a negative influence on one's feeling of well-being. The current sign of the situational influence of T on W therefore is '$?(-)$'. The situational qualitative network for the prior state is shown in Figure 4.2.*

The dynamic nature of the situational sign of the influence of T on W is demonstrated by entering the observation $F = f$ into the network. As a consequence of this observation, the state of the network changes. More specifically, now $\Pr(f) = 1.0$. Given this probability, the *difference* $\Pr(w \mid t) - \Pr(w \mid \bar{t}) = 0.90 - 0.75 = 0.15$ *is positive. In the new state of the network, therefore, the sign of the situational influence of* T *on* W *is '*$?(+)$*'.*

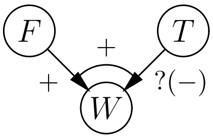


Figure 4.2: The network from Figure 2.7, with the prior situational influence of T on W.

Note that, although in the previous example the prior situational sign for the non-monotonic influence of T on W was computed from the corresponding quantitative Bayesian network, in a realistic application it would be elicited directly from a domain expert. In the remainder is assumed that initially situational signs are specified for the prior state of a network.

4.2 Inference with a Situational Qualitative Network

For inference with a regular qualitative probabilistic network, the efficient sign-propagation algorithm reviewed in Section 2 is available. This algorithm is readily extended to apply to situational qualitative networks by building upon the observation that the situational sign of an influence of A on C indicates the effect of the observation of A on the probability distribution over C's values just like a regular sign, albeit only for a particular state of the network. A situational sign can therefore be used as a regular sign upon inference *provided* that it is valid in the state under consideration. In Section 4.2.1, a method is presented for verifying the validity of the situational signs in a network as observations become available and the network converts to another state; this method also provides for updating the signs if necessary. In Section 4.2.2 this method is incorporated into the sign-propagation algorithm to provide for inference with a situational qualitative network.

4.2.1 The Dynamics of Situational Signs

To investigate the dynamics of a situational sign, the network fragment from Figure 4.3 will be studied. The network fragment consists of a node C with just two parents A and B. Now consider that in the state of the network in which the evidence $\mathbf{e}$ has been entered, the non-monotonic influence $?(\delta_1)$ is found between the nodes A and C and consider that the sign of the additive synergy of A and B with respect C equals δ_2. The situational sign δ_1 of the influence between A and C equals the sign of the expression

$$\begin{aligned} & \Pr(c \mid ab) \cdot \Pr(b \mid \mathbf{e}) + \Pr(c \mid a\bar{b}) \cdot \Pr(\bar{b} \mid \mathbf{e}) - \Pr(c \mid \bar{a}b) \cdot \Pr(b \mid \mathbf{e}) - \Pr(c \mid \bar{a}\bar{b}) \cdot \Pr(\bar{b} \mid \mathbf{e}) \\ = \; & \Pr(b \mid \mathbf{e}) \cdot \big(\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})\big) + \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}\bar{b}) \end{aligned}$$

Note that the sign of the term $\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})$ of this expression equals δ_2, that is, it equals the sign of the additive synergy of A and B with respect to C.

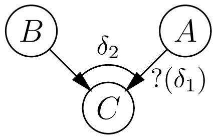


Figure 4.3: A fragment of a situational network, consisting of node C and its parents A and B, with $S^{?(\delta_1)}(A, C)$ and $Y^{\delta_2}(\{A, B\}, C)$.

Now suppose that an additional observation $F = f_i$ is entered into the network from which the network from Figure 4.3 is a fragment. The situational sign of the influence between A and B then equals the sign of

$$\Pr(b \mid \mathbf{e}f_i) \cdot \big(\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})\big) + \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}\bar{b})$$

The only adjustment thus is a change in the probability of b. Whether or not the situational sign changes now depends on the current sign of the situational influence, the sign of the additive synergy and the change in the probability of b. From the expression above, it is readily observed that in case of a positive additive synergy, a positive situational sign will persist to hold if $\Pr(b \mid \mathbf{e}f_i) > \Pr(b \mid \mathbf{e})$ and a negative sign will persist if $\Pr(b \mid \mathbf{e}f_i) < \Pr(b \mid \mathbf{e})$; in case of a negative additive synergy, a positive situational sign will persist to hold if $\Pr(b \mid \mathbf{e}f_i) < \Pr(b \mid \mathbf{e})$ and a negative sign will persist if $\Pr(b \mid \mathbf{e}f_i) > \Pr(b \mid \mathbf{e})$. In all other cases, the situational sign may become invalid. The updating of the situational sign δ_1 in the network fragment of Figure 4.3 thus is captured by

$$\delta_1 \leftarrow \delta_1 \oplus (\mathrm{sign}[B] \otimes \delta_2)$$

The updating of the situational sign is illustrated graphically for the network fragment from Figure 4.3 given that initially the network is in its prior state and given that the nodes A and B are independent in the prior network. The probability of $\Pr(c)$ now equals

$$\begin{aligned} \Pr(c) &= \Pr(a) \cdot \big(\Pr(c \mid a) - \Pr(c \mid \bar{a})\big) + \Pr(c \mid \bar{a}) \\ &= \Pr(a) \cdot \Big((\Pr(b) \cdot \big(\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})\big) + \\ &\quad \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}\bar{b})\Big) + \Pr(b) \cdot \big(\Pr(c \mid \bar{a}b) - \Pr(c \mid \bar{a}\bar{b})\big) + \Pr(c \mid \bar{a}\bar{b}) \end{aligned}$$

Note that for a specific $\Pr(b)$, the prior probability $\Pr(c)$ is a linear function of $\Pr(a)$; this function equals $\Pr(c \mid \bar{a})$ at $\Pr(a) = 0$ and $\Pr(c \mid a)$ at $\Pr(a) = 1$. The sign of the difference $\Pr(c \mid a) - \Pr(c \mid \bar{a})$ of these two extremes now is the situational sign of the influence of A on C in the state of the network that is associated with that particular $\Pr(b)$. This sign also is the sign of the gradient of the function that expresses $\Pr(c)$ in terms of $\Pr(a)$. Figures 4.4 and 4.5 show examples of $\Pr(c)$ as a function of $\Pr(a)$ and $\Pr(b)$, for the network fragment of Figure 4.3; Figure 4.4 results from a specification of the network in which the situational influence between A and B is negative for smaller values of the probability of b and positive for larger values; Figure 4.5 results from a specification for which the signs are reversed.

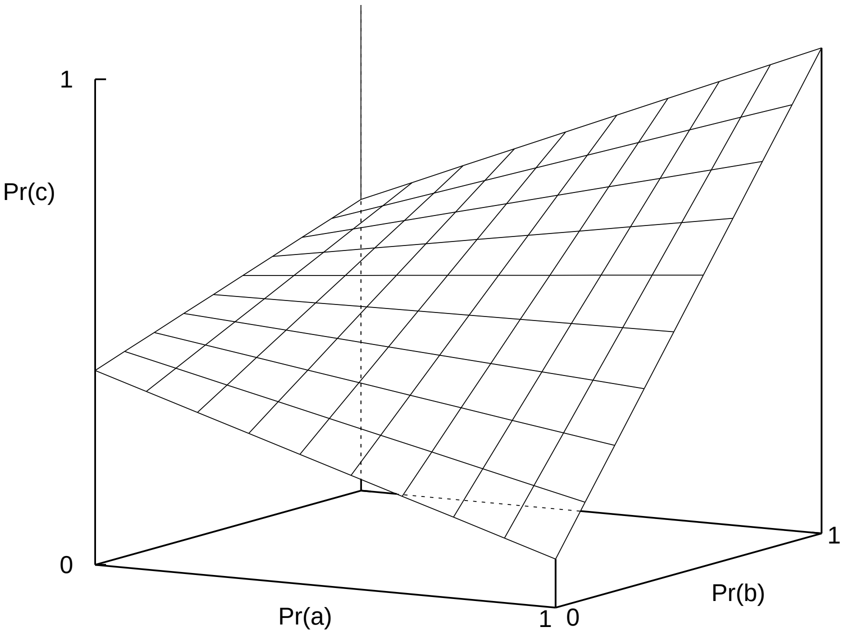


Figure 4.4: The probability $\Pr(c)$ as a function of $\Pr(a)$ and $\Pr(b)$ for the conditional probabilities $\Pr(C \mid AB)$ such that $S^{?}(A, C)$, $S^{+}(B, C)$ and $Y^{+}(\{A, B\}, C)$.

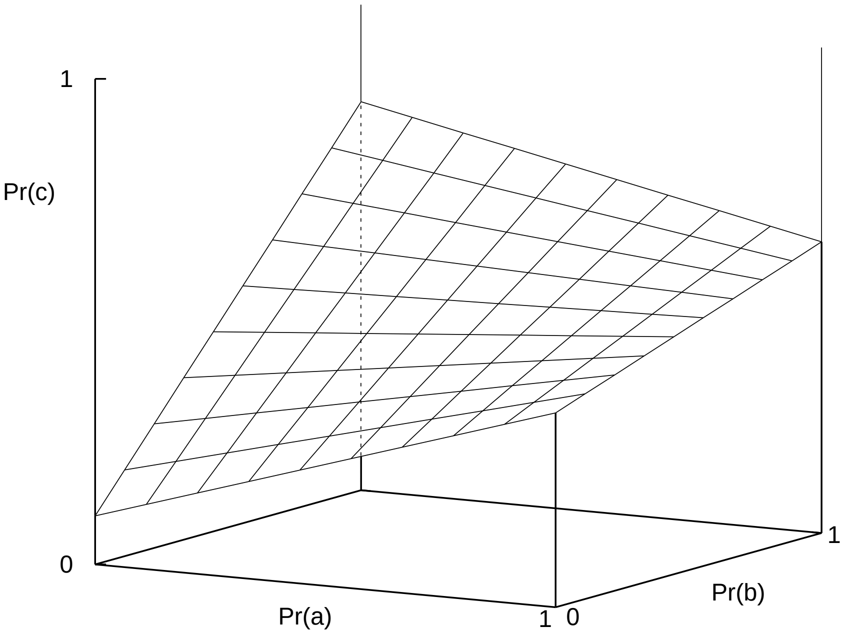


Figure 4.5: The probability $\Pr(c)$ as a function of $\Pr(a)$ and $\Pr(b)$ for the conditional probabilities $\Pr(C \mid AB)$ such that $S^{?}(A, C)$, $S^{+}(B, C)$ and $Y^{-}(\{A, B\}, C)$.

For the network resulting in Figure 4.4 the properties $\Pr(c \mid ab) - \Pr(c \mid \bar{a}b) > 0$ and $\Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}\bar{b}) < 0$ hold, from which it is found that $\Pr(c \mid ab) + \Pr(c \mid \bar{a}\bar{b}) > \Pr(c \mid a\bar{b}) + \Pr(c \mid \bar{a}b)$. The two properties thus imply a positive additive synergy of A and B with respect to C. Similarly, for the network resulting in Figure 4.5, a negative additive synergy of A and B with respect to C is found. Figure 4.6 depicts for both situations the sign of the gradient of $\Pr(c)$ as a function of $\Pr(a)$ relative to $\Pr(b)$. From Figure 4.6(a) it is easily verified that, in case of a positive additive synergy of A and B with respect to C, the situational sign δ_1 will definitely persist if it is negative in the prior network and the probability of b decreases as a consequence of the entered evidence, or if it is positive and the probability of b increases. Similarly it is seen from Figure 4.6(b) that, in case of a negative additive synergy, the situational sign will definitely persist if it is negative and the probability of b increases, or if it is positive and the probability of b decreases. The two figures further show that in all other cases, the situational sign may indeed become invalid upon a change in the probability of b.

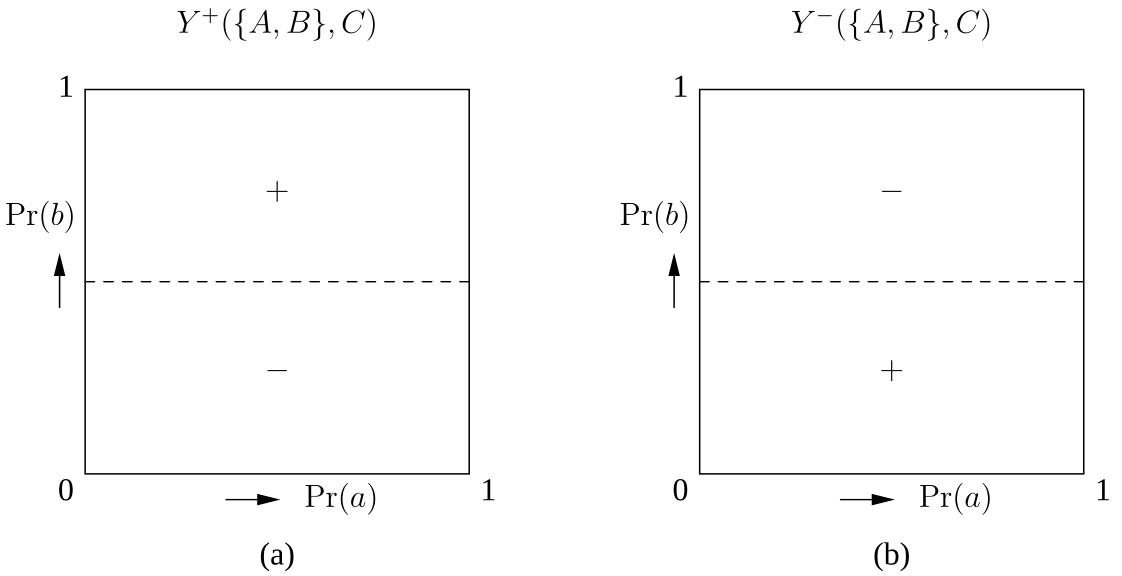


Figure 4.6: The sign of the gradient of $\Pr(c)$ as a function of $\Pr(a)$ relative to $\Pr(b)$, for the two different manifestations of a non-monotonic influence of A on C in the network fragment of Figure 4.3.

Without further substantiation, the previous observations are extended to the more general situation in which C has multiple parents. Consider a node C with parents A and B^i, $i = 1, \ldots, n, n \geq 1$, with $S^{?(\delta)}(A, C)$ and $Y^{\delta^i}(\{A, B^i\}, C)$. Informally, if for each parent B^i the direction of change in the probability of b^i supports the current sign of the situational influence given the sign of the corresponding additive synergy, then the situational sign persists. More formally, the updating of the situational sign of the influence of A on C is captured by

$$\delta \leftarrow \delta \oplus_i (\text{sign}[B^i] \otimes \delta^i)$$

4.2.2 The Adapted Sign-Propagation Algorithm

The basic sign-propagation algorithm for inference with a qualitative network has to be adapted to render it applicable to situational qualitative networks. In essence, the following modifications are required. First, for non-monotonic influences, the situational signs should be used for the propagation instead of the original ambiguous '$?$'. In the process of sign propagation, moreover, it may occur that a sign is propagated over a situational influence of a node A on a node C, while the fact that the probability distribution of another parent of C has changed does not become apparent until later in the inference. It may then turn out that the situational sign of the influence should have been updated and that incorrect signs were propagated. The algorithm therefore has to verify the validity of a situational sign as soon as information to this end becomes available and, if the situational sign is no longer valid, to restart the inference with the network with the updated situational sign. Since a situational sign can change at most twice, from '0' to '$+$' or '$-$' and then only to '$?$', the number of restarts is limited to twice the number of situational influences in the network.

The adapted sign-propagation algorithm is summarised in pseudo-code in Figure 4.7. The algorithm takes for its input a situational qualitative network (Q), a node for which an observation has become available (O), and the sign of this observation (*sign*). The algorithm constructs from the network the set *ARC*$_{nm}$ of all arcs with an associated non-monotonic influence, and the set *COP*$_{nm}$ of all nodes that are co-parents of a node exerting a non-monotonic influence; the function *COP–ARC*$_{nm}(A)$ takes for its argument a node A and returns all arcs to its children, $\sigma(A)$, that have associated a non-monotonic influence exerted by a co-parent of A. While the procedure 'Process-Observation' is identical to that in the regular algorithm, the procedure 'Propagate-Sign' is modified. After the assignment 'sign[to] $\leftarrow$ sign[to] $\oplus$ *message*', which may have led to a change of node sign, a call to the new function 'Effect-On-SitSign' is inserted. This function serves to verify and update the situational signs of the network. Following the call to the function 'Effect-On-SitSign', the inference is either resumed or restarted, depending upon whether or not a situational sign has changed.

Like the basic sign-propagation algorithm, the adapted algorithm serves to compute the effect of a single observation on the separate marginal distributions for all other nodes in a network. The joint effect, on a node of interest, of multiple simultaneous observations can again be computed as the $\oplus$-sum of the effects of the separate observations. However, when a situational sign changes during the propagation of one of the observations, the propagation of all other observations has to be performed anew with the adapted network before establishing the joint effect. Again, because a situational sign can change at most twice, the number of restarts is limited.

The following example illustrates the various steps of the extended sign-propagation algorithm.

Example 4.2 *Consider the situational qualitative network from Figure 4.8; the network is identical to the regular qualitative network from Figure 2.9, except that it is supplemented with a situational sign for the non-monotonic influence of node A on node C for the prior state of the network. From the situational network, the sets $ARC_{nm} = \{A \rightarrow C\}$ and $COP_{nm} = \{B\}$ are established. Suppose that again the interest is in the effect of observing $E = e_2$ on the marginal probability distributions over the other nodes in the network. The inference is started by sending*

```
$ARC_{nm} = \{A^i \rightarrow A^j \mid S^{?(\delta)}(A^i, A^j)\}$
$COP_{nm} = \{A^k \mid A^k \in \pi(A^j) \setminus \{A^i\}, A^i \rightarrow A^j \in ARC_{nm}\}$
$COP\text{–}ARC_{nm}(A) = \{A^i \rightarrow A^j \mid A^i \rightarrow A^j \in ARC_{nm}, A^j \in \sigma(A), A^i \neq A\}$;

procedure Process-Observation($Q$,$O$,sign):

   for all $A^i \in V(G)$ in $Q$
   do sign[$A^i$] ←'0';
   Propagate-Sign($Q$,∅,$O$,sign).

procedure Propagate-Sign($Q$,trail,$to$,message):

   sign[$to$] ← sign[$to$] $\oplus$ message;
   trail ← trail $\cup$ $\{to\}$;
   if Effect-On-SitSign($Q$,$to$)
   then exit and restart with Process-Observation($Q$,$O$,sign);
   for each active neighbour $A^i$ of $to$ in $Q$
   do  linksign ← (situational) sign of influence between $to$ and $A^i$;
       message ← sign[$to$] $\otimes$ linksign;
       if $A^i \notin$ trail and sign[$A^i$] $\neq$ sign[$A^i$] $\oplus$ message
       then Propagate-Sign($Q$,trail,$A^i$,message).

function Effect-On-SitSign($Q, A^i$):

   if $A^i \in COP_{nm}$
   then for all $A^j \rightarrow A^k \in COP\text{–}ARC_{nm}(A^i)$
          do Verify-Update($S^{?(\delta)}(A^j, A^k)$);
          if a $\delta$ changes
          then $Q \leftarrow Q$ with adapted signs;
                 return true;
   return false.
```

Figure 4.7: The adapted sign-propagation algorithm.

the message '$-$' to the node E. E updates its node sign to $0 \oplus - = -$ and subsequently sends the message $- \otimes + = -$ to its neighbour C. node C updates its node sign to $0 \oplus - = -$ and subsequently sends the messages $- \otimes - = +$ to B, $- \otimes + = -$ to A, and $- \otimes + = -$ to F. Upon receiving these messages, nodes B, A and F update their node signs to $0 \oplus + = +$, $0 \oplus - = -$ and $0 \oplus - = -$, respectively. The algorithm now establishes that A is a co-parent, with B, of C and that the influence of A on C is non-monotonic. Because the node sign of B has changed, the validity of the situational sign of the influence of A on C needs to be verified. The algorithm therefore checks if the current situational sign equals the product of the node sign of B and the sign of the additive synergy involved. Since $+ = + \otimes +$, the algorithm concludes that the situational sign remains valid. The inference resumes with node B sending the message $+ \otimes + = +$ to D. D updates its node sign to $0 \oplus + = +$ and the inference ends. The resulting *node signs are indicated in the figure.*

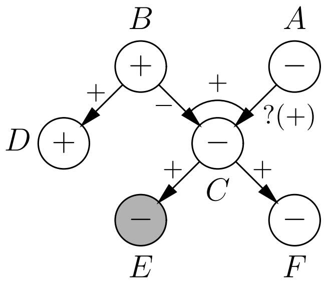


Figure 4.8: A situational qualitative network and its node signs after propagating the observation $E = e_2$.

The inference results from the examples 2.6 and 4.2 demonstrate that inference with a situational network can yield more informative results than inference with the corresponding regular qualitative network.

Chapter 5

An Experimental Study

By means of a small, artificially constructed network, it was demonstrated in the previous chapter that a situational qualitative network can yield more informative results upon inference than a corresponding regular qualitative network. In this section, the practicability of situational signs is investigated by studying the effects of their introduction into a real-life qualitative network in the field of oesophageal cancer. Since real-life truly qualitative networks were lacking a fully quantified real-life Bayesian network was abstracted qualitatively for this purpose. Section 5.1 provides some background information on the oesophageal cancer network and its qualitative abstraction. Section 5.2 describes the performance of the qualitative oesophageal cancer network before and after the introduction of situational signs, for a number of real patients. In Section 5.3 the results of this study are discussed.

5.1 The Oesophageal Cancer Network

A chronic lesion of the inner wall of the oesophagus may develop into a malignant tumour, which invades the oesophageal wall and will, eventually, invade organs adjacent to the oesophagus. The tumour may in time give rise to metastases, or secondary tumours, in lymph nodes and in other organs. The depth of invasion and extent of metastasis indicate how far the cancer has progressed or, phrased alternatively, in which stage it is. To establish the stage of a patient's cancer, various diagnostic tests are performed. The state-of-the-art knowledge about oesophageal cancer is captured in a Bayesian network [66]. This network includes 42 stochastic nodes and some thousand conditional probabilities. Its main diagnostic node is the node *Stage*, classifying a patient's cancer in one of six possible stages of disease. The leaves of the network capture the possible results of the different diagnostic tests.

For this study, the oesophageal cancer network was abstracted to a qualitative network. To this end, first all nodes were summarised into binary nodes; the original six-valued node *Stage*, for example, was translated into the binary node *Stage* with the values *early* and *late*. Furthermore, orderings were defined on the values of the resulting binary nodes; for example, *early* was considered to be 'smaller than' *late*. Given these orderings, the signs for the influences and the additive synergies between the nodes were established from the probabilities specified for the original network. Since the translation of the non-binary nodes had resulted in various (nearly)

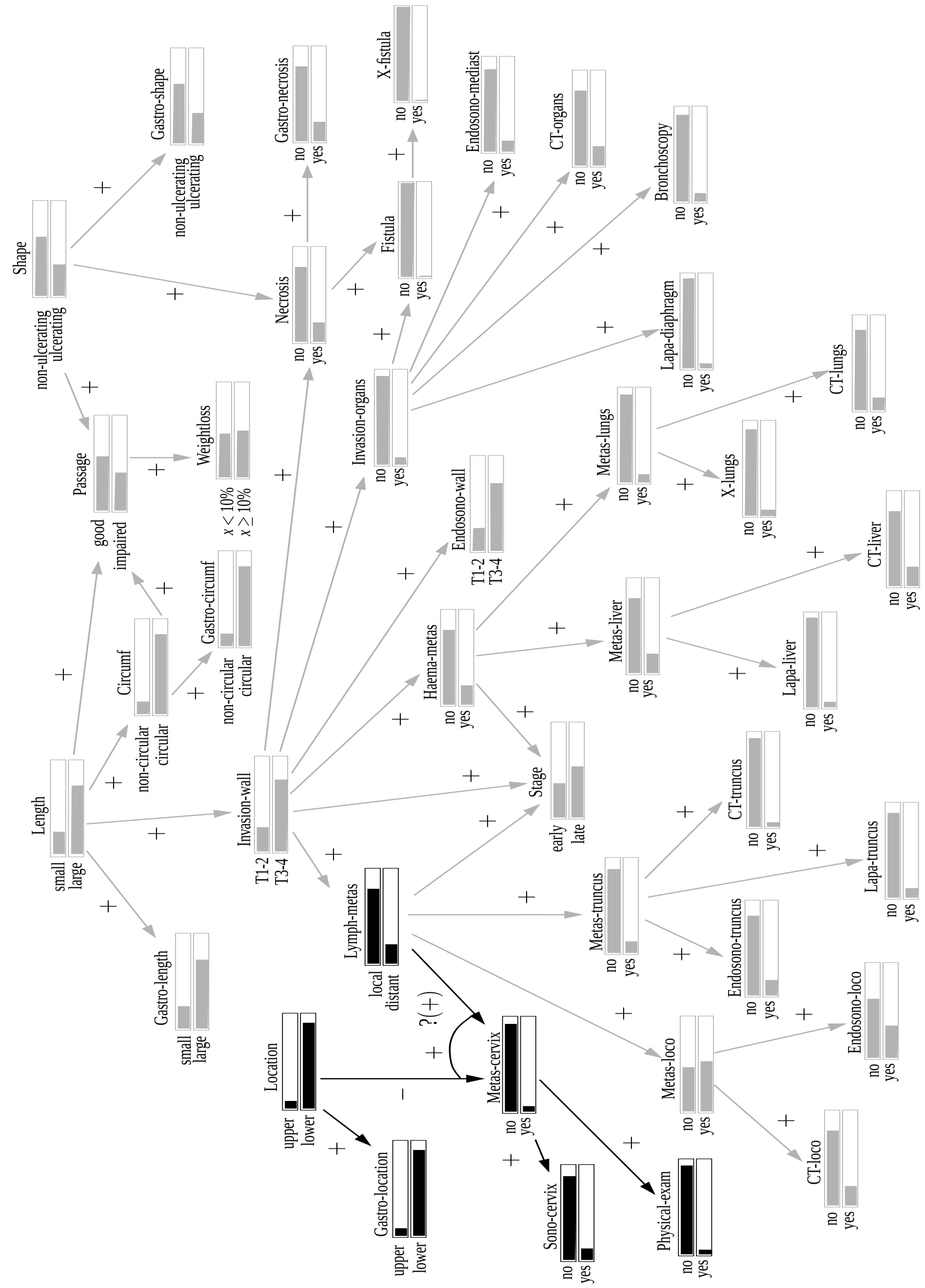


Figure 5.1: The combined binary and qualitative oesophageal cancer networks.

zero influences, it was decided to delete the arcs that were associated with these influences. This resulted in the removal of 15 arcs and two nodes. Figure 5.1 shows the binary quantitative oesophageal cancer network as well as its qualitative abstraction. For each node, its name, its values, and its prior marginal probability distribution are shown; for each arc, moreover, the sign of the associated qualitative influence is depicted.

The qualitative oesophageal cancer network includes a single non-monotonic influence, located between the nodes *Lymph-metas* and *Metas-cervix*. The non-monotonicity arises from the knowledge that metastases in the lymph nodes in the neck are considered to be local to a primary tumour in the upper one-third and distant to a primary tumour in the lower two-thirds of the oesophagus. The influence thus depends on the node *Location* that models the location of the primary tumour in the oesophageal tract. For a primary tumour located in the upper part of the oesophagus, the presence of metastases in distant lymph nodes has a negative effect on the probability of metastases in the neck; if, on the other hand, the primary tumour is located in the lower part, the presence of distant lymphatic metastases has a positive effect on this probability. In the initial state of the network, where no evidence has been entered, the probability of the tumour being located in the lower two-thirds of the oesophagus is quite high, and the situational sign of the non-monotonic influence, accordingly, is '$+$'.

The non-monotonic influence resides in a pivotal part of the qualitative oesophageal cancer network, since knowledge of the extent of the lymphatic metastases is of primary importance for establishing the stage of a patient's cancer. The nodes *Physical-exam* and *Sono-cervix* model the diagnostic tests that are generally performed to establish the presence or absence of lymphatic metastases in the neck; upon observation, these nodes influence the node sign of *Lymph-metas*. The location of the primary tumour is established through a gastroscopic examination of the oesophagus; the node *Gastro-location* models the result of this examination. *Gastro-location* bears no influence on *Lymph-metas*, because, in the prior state of the network, *Gastro-location* is independent of *Lymph-metas*. The node sign of *Location* is influenced by observations for all three nodes and is instrumental in updating the situational sign of the non-monotonic influence after observations have caused the network's state to change.

5.2 The Effect of the Introduction of Situational Signs

To gain insight into the practicability of situational signs, the performance of the qualitative oesophageal cancer network, before and after the introduction of a situational sign for its non-monotonic influence, is studied. The focus of the analysis is on the part of the network that serves for interpreting the findings with regard to metastases in the neck; the part of the network under study is indicated in black in Figure 5.1. It is investigated whether useful information from this part of the network is propagated towards the node *Lymph-metas* upon inference. In this study, the data of 156 real patients diagnosed with cancer of the oesophagus are used. As an example, first the effect of the introduction of the situational sign for a single patient is demonstrated; thereafter the effects for all patients from the data collection are summarised.

Example 5.1 *For patient 90-1042, a gastroscopic examination showed a primary tumour in the lower two-thirds of the oesophagus. Physical examination did not reveal any enlarged lymph* *nodes in the patient's neck. A sonography was not performed. The two available observations are entered into the network as a '$+$' for the node* Gastro-location *and a '$-$' for* Physical-exam*, respectively. Upon inference with the regular qualitative network, the node* Lymph-metas *receives the message* $- \otimes + \otimes ? = ?$ *from the observed node* Physical-exam *over the non-monotonic influence between* Lymph-metas *and* Metas-cervix*. The observation of* Gastro-location *does not affect the node sign of* Lymph-metas *and inference results in an overall influence of sign '$?$' on this node.*

In the situational oesophageal cancer network, the non-monotonic influence between Lymph-metas *and* Metas-cervix *is supplemented with a situational sign. Note that* Metas-cervix *has the node* Location *for its other parent. Because the two available observations change the node sign of* Location*, the sign-propagation algorithm identifies that the situational sign needs updating. The node sign of* Location *captures the combined effect of the two observations: since both observations have a positive effect on* Location*, its node sign is '$+$'. The additive synergy of* Location *and* Lymph-metas *on* Metas-cervix *also is '$+$'. Updating the situational sign of the influence between* Metas-cervix *and* Lymph-metas *now gives* $+ \oplus (+ \otimes +) = +$*, that is, the situational sign retains its validity and, hence, its informativeness. The part of the network that pertains to metastases in the neck now exerts an overall influence of sign* $- \otimes + \otimes + = -$ *on* Lymph-metas*. Note that, if the node sign of* Location *would have changed to '$-$', then the situational sign would have been updated to '$?$'. The observation for the node* Physical-exam *would then have exerted an ambiguous influence on* Lymph-metas*. A similar observation holds if the node sign of* Location *would have changed to '$?$'. Such a change occurs if the available observations exert discordant influences on* Location*, for example* Physical-exam = yes *and* Gastro-location = lower.

Table 5.1: The available observations for the relevant nodes for 156 patients.

Sono-cervix and *Physical-exam*	*Gastro-location* upper	lower
consistently positive	4	7
consistently negative	7	52
inconsistent	-	1
not observed	2	83

The data collection available for the study includes the medical records of 156 patients diagnosed with oesophageal cancer. For 11 of these patients, either *Sono-cervix* = *yes* and *Physical-exam* = *yes*, or one of these observations is *yes* and the other one is unknown. In the sequel such combinations of observations will be called consistently positive; negative consistency has an analogous meaning. For 59 patients consistently negative observations were reported from the sonography and the physical examination; for one patient contradictory results were found from the two diagnostic procedures. For the remaining 85 patients, no observations are available from a sonography of the neck and from a physical examination. These and some additional statistics are summarised in Table 5.1.

For the 85 (55%) patients for whom no observations are available for *Sono-cervix* and *Physical-exam*, the part of the network under study does not partake in establishing the node sign of *Lymph-metas*. The non-monotonic influence, therefore, is not used upon inference for these patients. For the remaining 71 (45%) patients, inference with the regular qualitative oesophageal cancer network results in an unknown influence on the node *Lymph-metas*.

Upon using the corresponding situational network, the availability of the situational sign makes no difference for the 85 patients without any observations for *Sono-cervix* and *Physical-exam*. For the other 71 patients, the situational sign of the non-monotonic influence is now used upon inference, instead of the original '$?$'. For all these patients, the available observations result in a change of the node sign of the node *Location*, thereby enforcing the situational sign to be updated. For 19 (12% of all patients) of the 71 patients, the node sign of *Location* changes to a '$-$' or a '$?$'. As for these patients the situational sign is changed to '$?$', inference again results in an unknown effect on the node *Lymph-metas*. For the remaining 52 (33%) patients, namely those with consistently positive observations for *Sono-cervix* and *Physical-exam*, and *Gastro-location = lower*, however, the node sign of *Location* changes to a '$+$' and the situational sign retains its validity. For these patients, inference yields an overall negative influence on the node *Lymph-metas* and, hence, an informative result. The inference results obtained with the regular and situational qualitative oesophageal cancer networks are summarised in Table 5.2.

Table 5.2: The signs propagated from the part of the network under consideration to the node *Lymph-metas* for 156 patients.

	$+$	$-$	$?$	0
regular	-	-	71 (45%)	85 (55%)
situational	-	52 (33%)	19 (12%)	85 (55%)

5.3 Discussion

Before the introduction of situational signs into the qualitative oesophageal cancer network, for 45% of the patients ambiguous information was propagated from the part of the network under consideration. This percentage reduced to 12% after introducing a situational sign for the non-monotonic influence involved. The introduction of the situational sign thus served to considerably increase the expressive power of the qualitative oesophageal cancer network.

In this study, for all 71 patients for whom one or more observations were available for the nodes *Sono-cervix* and *Physical-exam*, the situational sign had to be verified upon inference. For 52 of these patients, the sign proved to retain its validity. Also for the other 85 patients, the situational sign was reconsidered upon inference, even though it was not used for further propagation. For two of these patients, the situational sign changed to '$?$' and for 83 of these patients, the situational sign remained a '$+$'. For a total of 135 (87%) patients, therefore, the situational sign retained its validity after updating. This apparent robustness of the situational sign is not coincidental. The initial positive situational sign depends on the relatively high prior probability of the tumour being located in the lower two-thirds of the oesophagus and the positive additive synergy of the nodes *Location* and *Lymph-metas* on *Metas-cervix*. Because of the high probability of a lower tumour, moreover, it is more likely to find observations that lead to a change of the node sign of the node *Location* to ‘$+$’. Given the positive additive synergy, these observations are exactly the ones that do not induce a change of the situational sign.

Chapter 6

Conclusions

Qualitative probabilistic networks capture the probabilistic influences among their nodes by means of qualitative signs. If an influence between two nodes is non-monotonic, it has associated the ambiguous sign '?', even though the effect of the influence is unambiguous in any specific state of the network. The presence of such ambiguous signs tends to lead to ambiguous and, hence, uninformative results upon inference. The concept of situational sign was introduced to capture information about the current effect of non-monotonic influences and it was shown that situational signs can be used upon inference and may effectively forestall ambiguous results. The conditions were identified under which situational signs retain their validity and a method was presented for updating them if necessary. Although the dynamics of situational signs were studied in networks where the non-monotonicity involved originates from a single node, the presented ideas and methods are readily generalised to networks where the non-monotonicity is provoked by more than one node.

The previous chapters dealt with binary qualitative networks only. The definitions and observations can be generalised to networks involving non-binary nodes by building on the concept of general statistic dominance, however [74]. For non-binary nodes, another type of non-monotonicity can arise from the ordering of the values of the nodes. This second type of non-monotonicity remains to be examined.

To investigate the practicability of situational signs, the effect of their introduction into a real-life qualitative network in the field of oncology was studied. In the study, the performance of the network before and after the introduction of situational signs was compared, using the data from 156 patients. It was found that the introduction of situational signs served to considerably increase the expressive power of the network. As the studied network is in no aspect exceptional, similar results may be expected for other real-life qualitative networks in a variety of problem domains; further investigation to corroborate this expectation is required, however.

Part II

Loopy Propagation

When Pearl's algorithm for reasoning with singly connected Bayesian networks is applied to a network with loops, the algorithm is no longer guaranteed to result in exact probabilities. In this part of the thesis, two types of error are identified that may arise in the probabilities yielded by the algorithm, called the *convergence error* and the *cycling error*. These types of error then are investigated in more detail for the convergence nodes and the inner nodes of a loop separately. First, the focus is on the *convergence nodes* of a loop. A general expression is derived for the error that is found in the approximate probabilities computed for these nodes in a network's prior state. This expression includes a weighting factor that is captured by the notion of *quantitative parental synergy*. Subsequently the changes induced by evidence are studied. Then the focus is on the *inner nodes* of a loop. For these nodes, the effect of the cycling error on the decisiveness of their approximate probabilities is analysed. More specifically, the over- or underconfidence of the computed approximations is linked to two concepts of qualitative probabilistic networks. This part of the thesis concludes with an analysis of an algorithm for probabilistic inference with undirected networks which is equivalent to the loopy-propagation algorithm. It is shown how, although in undirected networks all errors arise from the cycling of information, the convergence error yet is embedded in the algorithm.

Chapter 7

Introduction

The complexity of probabilistic inference is known to be NP-hard in general [10]. For networks of complex topology for which exact inference is infeasible, the question arises whether good approximations can be computed in reasonable time. Unfortunately, also the problem of establishing approximate probabilities with guaranteed error bounds is NP-hard in general [12]. Although their results are not guaranteed to lie within specific error bounds, various approximation algorithms have been designed for which good performance has been reported. One of these algorithms is the *loopy-propagation algorithm*. The basic idea of this algorithm is to apply Pearl's propagation algorithm, which is designed for singly connected networks, to a Bayesian network regardless of its topological structure. From an experimental point of view, Murphy *et al.* [49] reported good approximation behaviour of the loopy-propagation algorithm used on Bayesian networks provided there was rapid convergence. The (conditional) probabilities specified in a network appeared to be an important factor in the algorithm's convergence behaviour. Murphy *et al.* in fact conjectured that oscillations forestalling (rapid) convergence were caused by a combination of observations with a very small probability. Excellent performance has also been reported for algorithms which are equivalent to the loopy-propagation algorithm, such as the turbo-decoding algorithm in coding theory [1, 35, 41, 42] and algorithms used in image analysis [18, 19, 70].

Several researchers analysed the approximation behaviour of the loopy-propagation algorithm from a more fundamental point of view. Most of this research, however, was performed for algorithms equivalent to the loopy propagation algorithm, used on undirected networks, such as factor graphs or pairwise Markov networks. The use of undirected networks is motivated by the relatively easier analysis of these networks and is justified by the observation that any Bayesian network can be converted into such an undirected network [77]. Weiss and Freeman, studied the performance of an equivalent algorithm on pairwise Markov networks [71, 72]. For pairwise Markov networks with a single loop Weiss, derived an analytical relationship between the exact probabilities and the approximate probabilities computed for the nodes in the loop [71]. The approximation error was found to be related to the convergence rate of the messages. Weiss and Freeman further studied loopy propagation in networks of arbitrary topology with normally distributed continuous variables. They showed that in such a network, given that the algorithm converges, correct posterior means are found for all variables but that the covariance estimates are generally incorrect. Further insights into the performance of the loopy-propagation algorithm were gained by relating the approximations to the notion of Bethe-free energy from the field of statistical physics. It was shown that the loopy-propagation algorithm can only converge to a minimum of the Bethe-free energy [24, 76]. With this insight also a generalisation of the loopy-propagation algorithm could be developed. The Bethe-free energy is the simplest from among a whole family of free energies called the Kikuchi-free energies. Informally speaking, the more complex a free energy, the larger the clusters of nodes which are taken into account. Now by paying an adjustable amount of computational costs, in generalised loopy-propagation one of the more complex Kikuchi energies can be obtained, with concordant better approximations [75,76]. Other variations on the loopy-propagation algorithm have also been proposed, such as algorithms that minimise the Bethe/Kikuchi free energy explicitly [25, 59, 63, 73, 78]; algorithms in which tree structures underlying the graph at hand are exploited [22, 43, 44, 67, 68]; methods in which a correction step is included [45, 46], and methods in which the message updating scheme is sequential rather than synchronous [17, 62]. Also progress is made with respect to bounding the error in the approximations [27, 40, 68]

In this thesis, the focus is on the performance of the loopy-propagation algorithm on Bayesian networks. Although global results are very difficult to derive for Bayesian networks, the following chapters show that by studying local structures, some interesting insights can be gained in the particulars of the performance of the algorithm on directed networks. In a Bayesian network, a distinction can be made between loop nodes with at most one incoming arc on the loop in which it is included and nodes with two or more incoming arcs on the loop; the former will be called the *inner nodes* and the latter will be called the *convergence nodes* of the loop. It will be argued that, in Bayesian networks, two different types of error are introduced in the computed approximate probabilities, which will be termed the convergence error and the cycling error. A convergence error arises whenever messages that originate from dependent variables within a loop are combined as if they were independent. Such an error arises at the convergence nodes only. A cycling error arises when messages are being passed within a loop repetitively. Cycling of information can occur as soon as for all the convergence nodes of a loop, either the convergence node itself or one of its descendants is observed. A cycling error arises at all the inner nodes of a loop.

Chapter 9 focusses on the errors that are found in the approximate probabilities established by the loopy-propagation algorithm for the convergence nodes of a network. First an expression is given for the convergence error which arises in the probabilities computed for a convergence node in a network in its prior state. The factors of this expression partly pertain to the degree of dependence between the parents of the convergence node in the loop and partly consist of weighting factors, composed of the conditional probabilities of the convergence node itself, that determine the extent to which this dependence can affect the computed probabilities. To capture these weighting factors, the notion of *quantitative parental synergy* is introduced. Thereafter the errors found for convergence nodes in a network in its posterior state are studied. Thereby, a distinction is made between the posterior convergence error itself and the additional error that results from the incorrect messages from the parents of the convergence node due to the cycling of information.

Chapter 10 then focuses on the error that arises at the inner loop nodes. As mentioned above, Weiss [71] derived, for a pairwise Markov network with just a single simple loop, an expression that relates the exact probability for a loop node to the approximate probabilities computed by the loopy-propagation algorithm. He notes that the approximate probabilities are overconfident as a result of double counting of evidence. In Chapter 10 is observed, however, that both overconfident and underconfident approximations can result in loopy propagation. The term *decisiveness* will be used to refer to the over- or underconfidence of an approximate probability. Decisiveness is an important concept: when a network is used for a diagnostic task, for example, knowledge of the decisiveness can be used to decide whether or not a particular probability is ‘certain enough’. The effect of the cycling error on the decisiveness of the approximations is studied for the inner nodes of a single simple loop in a Bayesian network with binary nodes. It is shown that the decisiveness depends on the sign of the qualitative influence between the parents of the loop’s convergence node and on the sign of the intercausal influence that is induced between these parents by the entered evidence. If the two influences have equal signs, then the cycling error results in overconfident approximations; if the two influences have opposite signs, the approximations are underconfident.

In the analysis of loopy propagation in undirected networks, no distinction between different error types can be made and, on first sight, there is no equivalent for the convergence error. In Chapter 11, this difference in results is investigated. The simplest Bayesian network in which a convergence error may occur is constructed and converted into its equivalent Markov network. The convergence error was found to be converted into a cycling error in the equivalent Markov network. Furthermore, was found that the prior convergence error in Markov networks is characterised by the fact that the relationship between the exact and the approximate probabilities, as established by Weiss, does not exist for this node.

Chapter 8

Errors in Loopy Propagation

Pearl's propagation algorithm is designed for probabilistic reasoning with singly connected Bayesian networks. When applied to a singly connected network, after a limited number of time steps proportional to the diameter of its digraph, the network reaches an equilibrium state and exact probabilities are yielded. When applied to a multiply connected network, that is, upon loopy propagation, the algorithm is no longer guaranteed to halt. Experience shows, however, that for many applications the algorithm nonetheless does converge to an equilibrium state[1]. The probabilities that are returned by the algorithm may then deviate from the true probabilities of the modelled distribution, however. In this chapter, two types of error are distinguished that may arise upon applying Pearl's propagation algorithm to a multiply connected network: convergence errors and cycling errors. The first type of error arises in the convergence nodes of a loop and the second type arises in a loop's inner nodes. In the sequel the probabilities, causal messages and causal parameters that result from loopy propagation will be supplemented with a tilde to distinguish them from their exact counterparts.

8.1 The Convergence Error

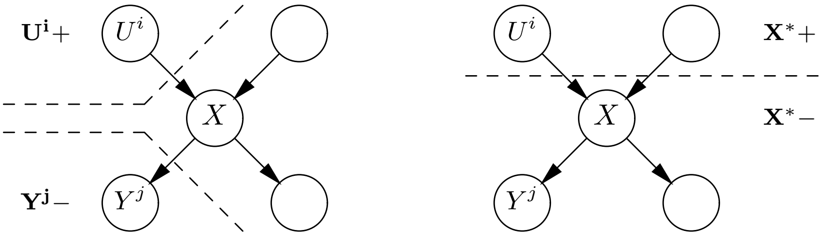
Convergence errors arise in the convergence nodes of a network and may already be found upon loopy propagation in a multiply connected network in its prior state, that is, they may be found even if no evidence has been entered as yet into the network. The error may arise in the computation of the compound causal parameter for a convergence node and in the computation of the diagnostic messages that a convergence node will send to its parents in the loop. This section, elaborates on the origin of the convergence error by discussing the computation of the compound causal parameter for a convergence node. Section 8.3, illustrates, by means of an example, that the same origin serves to cause an error in the diagnostic messages sent by the convergence node.

As reviewed in Section 2.3.1, and referring to Figure 8.1(b), the compound causal parameter $\pi(X)$ for a node X should equal the marginal probability distribution over X given the observations that have been entered in the upper graphs $\mathbf{X}^{*}+$ of X. Pearl's propagation algorithm uses the following rule for the computation of the compound causal parameter:

[1] The algorithm will be considered to have converged as soon as all causal and diagnostic messages and all computed probabilities have changed by less than a pre-specified threshold value in the previous time step.

$$\pi(X) = \sum_{\mathbf{U}} \Pr(X \mid \mathbf{U}) \cdot \prod_i \pi_X(U^i)$$

where $\mathbf{U}$ denotes the set of parents $\{U^i\}$ of node X. This computation rule assumes mutual independence of the parents U^i of X. In a singly connected network, this independence assumption holds for any node X. The rule then serves for computing the correct compound causal parameter for the network's equilibrium state. In a multiply connected network, however, the parents U^i of a node X may be dependent. Application of the above rule for dependent parent nodes now may introduce an error in the compound causal parameter computed by the convergence node, which in turn results in a convergence error in the probabilities computed for the convergence node. A convergence error thus arises from combining the information of the dependent parents of a convergence node as if these parents were independent. Note that any error that results from a dependency between a node from the upper graph $\mathbf{X}^*+$ and a node from the lower graph $\mathbf{X}^*-$ given X does not arise in the convergence node itself. This type of error is discussed in the next section.



$\pi_X(U^i) = \Pr(X \mid \mathbf{e}^{\mathbf{U^i}+})$

$\lambda_{Y^j}(X) = \alpha \cdot \Pr(\mathbf{e}^{\mathbf{Y^j}-} \mid X)$

(a)

$\pi(X) = \Pr(X \mid \mathbf{e}^{\mathbf{X}^*+})$

$\lambda(X) = \alpha \cdot \Pr(\mathbf{e}^{\mathbf{X}^*-} \mid X)$

(b)

Figure 8.1: The subgraphs $\mathbf{U^i}+$ and $\mathbf{Y^j}-$ (a) and the subgraphs $\mathbf{X}^*+$ and $\mathbf{X}^*-$ (b).

As an example of the convergence error, consider the networks from Figures 8.2(a) and 8.2(b). The computation of the compound causal parameter is addressed for the convergence node C. In their prior states, in both networks, $\Pr(C)$ equals $\pi(C)$. Using Pearl's computation rule, the parameter $\pi(C)$ is computed to be

$$\pi(C) = \sum_{AB} \Pr(C \mid AB) \cdot \pi_C(A) \cdot \pi_C(B)$$

for both networks. For both networks, moreover, the causal parameters $\pi_C(A) = \Pr(A)$ and $\pi_C(B) = \Pr(B)$ are established. For the network from Figure 8.2(a), the computation rule thus results in the correct causal parameter

$$\pi(C) = \sum_{AB} \Pr(C \mid AB) \cdot \Pr(AB) = \sum_{AB} \Pr(C \mid AB) \cdot \Pr(A) \cdot \Pr(B)$$

for node C. For the network from Figure 8.2(b), however, the compound causal parameter should be equal to

$$\pi(C) = \sum_{AB} \Pr(C \mid AB) \cdot \Pr(B \mid A) \cdot \Pr(A)$$

which, in general, is not equal to the parameter $\widetilde{\pi}(C) = \sum_{AB} \Pr(C \mid AB) \cdot \Pr(A) \cdot \Pr(B)$ computed with the rule. For this network, therefore, an error is introduced in the compound causal parameter and thus in the computed probability distribution $\widetilde{\Pr}(C)$. The origin of the error is the dependence between the parents A and B of the convergence node C.

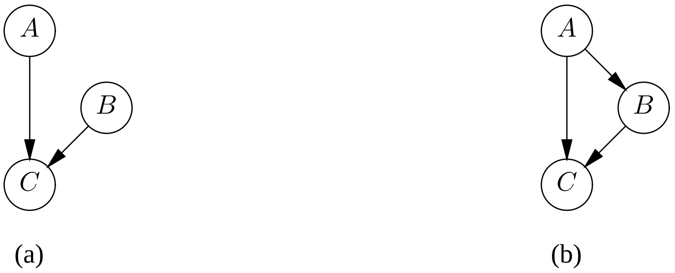


Figure 8.2: The graphical parts of two small example Bayesian networks, one of which is singly connected (a) and one of which is multiple connected (b).

8.2 The Cycling Error

A *cycling error* arises at the inner nodes of a loop and may do so as soon as for each convergence node of the loop either the convergence node itself or one of its descendants has been observed. In contrast with convergence errors, cycling errors thus cannot arise in a network in its prior state. The cycling error is conceptually more complex than the convergence error and involves all rules of Pearl's algorithm, except for the rule for computing the compound causal parameter.

Consider the simple network from Figure 8.3(b). The diagnostic message from node C to node A is now addressed. As reviewed in Section 2.3.1, and referring to Figure 8.1(a), the diagnostic message from a node Y^j to its parent X should equal the normalised probabilities of the observations entered into the lower graph $\mathbf{Y^j}-$ of X given X. For the network from Figure 8.3(b), after the observation of $C = c$, for example, the diagnostic message from node C to node A should be equal to $\alpha \cdot \sum_B \Pr(c \mid AB) \cdot \Pr(B \mid c)$. Note that this message includes the factor $\Pr(B \mid c)$ because node B is included in the lower graph $\mathbf{C}^-$, about which node C has to send probabilistic information to node A. At the same time, however, node C is part of the upper graph $\mathbf{B}^+$, from which node B has to receive information for its message to C. Thus, in the computation of the correct diagnostic message from node C to node A, the probability distribution $\Pr(B \mid c)$ is needed before it is actually available and the correct diagnostic message cannot be derived from the local messages in Pearl's algorithm.

More in detail, after the observation of $C = c$, the compound diagnostic parameter for node

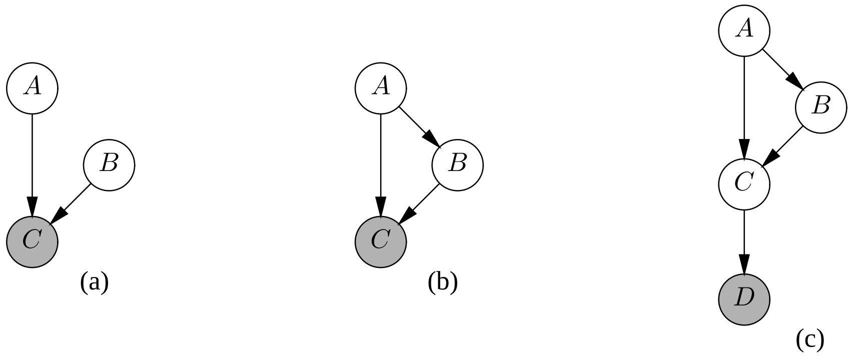


Figure 8.3: The graphical parts of three small example Bayesian networks; network (a) is singly connected and networks (b) and (c) are multiply connected. The shaded nodes have been observed.

C equals $\lambda(c) = 1$ and $\lambda(\bar{c}) = 0$. Recall that Pearl's rule for computing diagnostic messages is

$$\lambda_{Y^j}(X) = \alpha \cdot \sum_{Y^j} \lambda(Y^j) \cdot \sum_{\mathbf{X} \setminus \{X\}} \Pr(Y^j \mid \mathbf{X}) \cdot \prod_k \pi_{Y^j}(X^k)$$

where $\mathbf{X}$ is the set of parents $\{X^k\}$ of node Y^j[2]. Applying the rule to the example network from Figure 8.3(b) would result in the following diagnostic message from node C to node A

$$\lambda_C(A) = \alpha \cdot \sum_B \Pr(c \mid AB) \cdot \pi_C(B)$$

Node C thus requires the causal message $\pi_C(B)$ from node B to compute its diagnostic message to node A. In order to compute this message, node B in turn needs to receive the causal message $\pi_B(A)$ from node A. Pearl's computation rule for causal messages equals

$$\pi_{Y^j}(X) = \alpha \cdot \pi(X) \cdot \prod_{k \neq j} \lambda_{Y^k}(X)$$

For node B in the example network, the rule reads

$$\pi_B(A) = \alpha \cdot \Pr(A) \cdot \lambda_C(A)$$

To compute the causal message $\pi_B(A)$ to be sent to node B, node A needs to receive the diagnostic message $\lambda_C(A)$ from node C. This, however, is the message for which $\pi_B(A)$ was needed in the first place. In order to compute the correct message from node C to node A, the algorithm thus needs information that will only be available after the message from C to A is known. The message cannot be computed correctly by the algorithm and a cycling of messages will be observed. Similarly, the diagnostic message from node C to node B cannot be computed correctly.

[2] Note that here the rule is given from the perspective of X being a parent while in Section 2.3.1 the rule was given from the perspective of X being a child.

Note that in the singly connected network from Figure 8.3(a), nodes A and B can compute their causal messages $\pi_C(A) = \Pr(A)$ and $\pi_C(B) = \Pr(B)$ correctly. In this network, therefore, the problem described above does not occur.

Now suppose that the correct causal messages from nodes A and B to node C would be known. In the example network from Figure 8.3(b) then still incorrect probabilities may be computed by the loopy-propagation algorithm due to the incorrect combination of information. Underlying the data fusion rule of Pearl's algorithm

$$\Pr(X \mid \mathbf{e}) = \alpha \cdot \lambda(X) \cdot \pi(X)$$

is the assumption that the nodes in the two subgraphs $\mathbf{X}^*+$ and $\mathbf{X}^*-$ are conditionally independent given X. In the network from Figure 8.3(b), however, this assumption does not hold since the graphs $\mathbf{B}^*+$ and $\mathbf{B}^*-$ include the same nodes. By assuming independence upon application of the rule above, therefore, an error may be introduced in the probabilities computed for node B. Note that in the network from Figure 8.3(a), the subgraphs $\mathbf{B}^*+$ and $\mathbf{B}^*-$ are distinct and information from the two graphs is correctly combined. Underlying the computation rule of Pearl's algorithm for the compound diagnostic messages

$$\lambda(X) = \prod_j \lambda_{Y^j}(X)$$

moreover, is the assumption that the children Y^j of a node X are conditionally independent given X. In the network from Figure 8.3(b), however, the children C and B of node A are dependent. By assuming independence upon application of the rule, therefore, again an error may be introduced in the probabilities computed for node A. In the network from Figure 8.3(a), node A has only node C for its child and the error will not arise. The cycling error now is a combination of the constituent errors described above.

To conclude, the earlier observation that a cycling error cannot arise in a Bayesian network in its prior state, not even if the network is multiply connected, is considered in more detail. In the absence of observations, the two properties $\Pr(\mathbf{e}^{\mathbf{Y^j}-} \mid X) = \Pr(true \mid X) = 1$ and $\Pr(\mathbf{e}^{\mathbf{X}^*-} \mid X) = \Pr(true \mid X) = 1$ hold. In the network's prior state, therefore, all elements of the diagnostic parameter and the diagnostic messages of a node X should be equal. Now recall that in Pearl's algorithm, all message vectors are initialised to contain just 1s. As a result, all elements of the compound diagnostic parameter computed from

$$\lambda(X) = \prod_j \lambda_{Y^j}(X)$$

will be equal to 1. Re-computing the diagnostic messages in a next step of the algorithm, using

$$\lambda_X(U^i) = \alpha \cdot \sum_X \lambda(X) \cdot \sum_{\mathbf{U}\backslash\{U^i\}} \Pr(X \mid \mathbf{U}) \cdot \prod_{k \neq i} \pi_X(U^k)$$

will then again result in messages of which all elements are equal, etcetera. In a network's prior state, therefore, the correct diagnostic messages and parameters will indeed be computed.

8.3 Combined Errors

As discussed above, upon loopy propagation convergence errors arise in the convergence nodes of a loop of a Bayesian network, and cycling errors arise in the loop's inner nodes. Cycling errors, however, may be passed on to convergence nodes and convergence errors may be passed on to inner loop nodes. In a network's posterior state, all messages which are passed on in a loop may include cycling errors. A cycling error thus may enter a convergence node through the causal messages it receives from its parents in the loop. For example, after an observation has been entered for node D in the network from Figure 8.3(c), a cycling error may be introduced in the probabilities computed for node C. These probabilities then include both types of error.

For a simple loop, the convergence error is restricted to the probabilities computed for the convergence node and is not passed on to the inner nodes of the loop, since in such a loop, in the computation of the diagnostic messages, no combination of the information of the parents in the loop is required. In a compound loop, however, a convergence error may be passed on to the inner loop nodes through the diagnostic messages from the convergence node. Consider as an example the network from Figure 8.4 which includes a convergence node C with the three parents B^1, B^2 and B^3. For computing the diagnostic messages to be sent to its parents, node C applies the following rule

$$\lambda_X(U^i) = \alpha \cdot \sum_X \lambda(X) \cdot \sum_{\mathbf{U} \setminus \{U^i\}} \Pr(X \mid \mathbf{U}) \cdot \prod_{k \neq i} \pi_X(U^k)$$

where $\mathbf{U}$ is the set of parents U^k of node X. In the computation of the diagnostic message for node B^1, for example, the causal messages of the two other parents B^2 and B^3 of C are combined. Upon doing so, node C assumes that B^2 and B^3 are independent while in fact they are not, which results in an error in the computed message. This error has a similar origin as the error that arises in the compound diagnostic parameter, as discussed in Section 9.2.1, and thus is considered a convergence error. This error is included in the diagnostic message from node C to node B^1 and hence is passed on to an inner loop node.

In the above discussion the propagation of the two types of error upon loopy propagation are considered within the loop from which the errors originated. It will be evident that in a more involved network errors may be transported to nodes outside the loop and that errors from outside

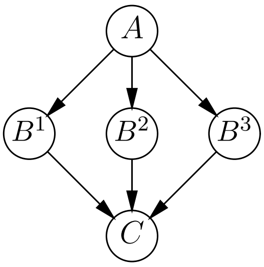


Figure 8.4: The graphical part of a small example Bayesian network including a convergence node C with the parents B^1, B^2 and B^3.

the loop may enter it. This thesis aims at analysing the various errors at their origin. An analysis of the effects of propagation of errors throughout a more involved network is an interesting area for future research.

Chapter 9

Convergence Nodes

When Pearl's propagation algorithm is applied to a Bayesian network with a multiply connected digraph, errors may arise in the probability distributions established for the various nodes. In this chapter the errors that result for the convergence nodes of a loop are investigated. In Sections 9.2 and 9.3 the errors arising in the prior state of a network are studied. In Sections 9.4 and 9.5, the change that is occasioned in the errors after observations have been entered into the network, are addressed; these latter two sections focus on networks with a single simple loop. In the following section, first the notion of *quantitative parental synergy* is introduced. This synergy will prove to be an important factor of the convergence error.

9.1 The Quantitative Parental Synergy

In Chapter 2, the notions of *qualitative influence* and *additive synergy* were reviewed for binary networks. Recall that a qualitative influence expresses how observing a value for the one node qualitatively affects the probability distribution over the values of the other node; an additive synergy expresses how two nodes interact in one another's effect on the probability distribution over a common descendent. Note that these two notions pertain to the entire distributions of the involved nodes. In this section a closely related quantitative notion, called the *quantitative parental synergy* is defined. In contrast with the notions of qualitative influence and additive synergy, the notion of quantitative parental synergy is defined with respect to a specific value of the child node and a specific value assignment to its parent node(s).

Before formally defining the quantitative parental synergy, the indicator function δ on the joint value assignments $a_{i1}^1, \ldots, a_{in}^n$ to a set of nodes $A^1, \ldots, A^n$, $n \geq 0$, given a specific assignment $a_{s1}^1, \ldots, a_{sn}^n$ to these nodes is introduced:

$$\delta(a_{i1}^1, \ldots, a_{in}^n \mid a_{s1}^1, \ldots, a_{sn}^n) = \begin{cases} 1 & \text{if } \sum_{k=1,\ldots,n} a_{ik}^k \neq a_{sk}^k \text{ is even} \\ -1 & \text{if } \sum_{k=1,\ldots,n} a_{ik}^k \neq a_{sk}^k \text{ is odd} \end{cases}$$

where $\mathrm{true} \equiv 1$ and $\mathrm{false} \equiv 0$. The indicator function compares the joint value assignment $a_{i1}^1, \ldots, a_{in}^n$ with the joint assignment $a_{s1}^1, \ldots, a_{sn}^n$, and counts the number of differences: the assignment $a_{i1}^1, \ldots, a_{in}^n$ is mapped to the value 1 if the number of differences is even and is mapped to -1 if the number of differences is odd. For the binary variables A and B, for example, $\delta(ab \mid ab) = 1$, $\delta(a\bar{b} \mid ab) = -1$, $\delta(\bar{a}b \mid ab) = -1$ and $\delta(\bar{a}\bar{b} \mid ab) = 1$.

Building upon the indicator function δ, the notion of quantitative parental synergy is defined as follows. Let $\mathbf{B}$ be a Bayesian network, representing a joint probability distribution $\Pr$ over a set of nodes $\mathbf{V}$. Let $\mathbf{A} = \{A^1, \ldots, A^n\} \subseteq \mathbf{V}, n \geq 0$, and let $C \in \mathbf{V}$ such that C is a child of all nodes in the set $\mathbf{A}$, that is, $A^j \rightarrow C, j = 1, \ldots, n$. Let $\mathbf{a}$ be a joint value assignment to $\mathbf{A}$ and let c_i be a value of C. Furthermore, let $\mathbf{X} \subseteq \rho(C) \backslash \mathbf{A}$ and let $\mathbf{x}$ be a value assignment to $\mathbf{X}$. The *quantitative parental synergy* of $\mathbf{a}$ with respect to c_i given $\mathbf{X} = \mathbf{x}$, denoted as $Y^\star_\mathbf{x}(\mathbf{a}, c_i)$, is

$$Y^\star_\mathbf{x}(\mathbf{a}, c_i) = \sum_{\mathbf{A}} \delta(\mathbf{A} \mid \mathbf{a}) \cdot \Pr(c_i \mid \mathbf{A}\mathbf{x})$$

Example 9.1 *Consider an arbitrary-valued node C with the two ternary parents A and B; the conditional probabilities for the value c_i of C given A and B, are listed in Table 9.1(a). The quantitative parental synergy $Y^\star(a_2b_2, c_i)$ of a_2 and b_2 with respect to c_i, for example, is computed from* $\Pr(c_i \mid a_1b_1) - \Pr(c_i \mid a_1b_2) + \Pr(c_i \mid a_1b_3) - \Pr(c_i \mid a_2b_1) + \Pr(c_i \mid a_2b_2) - \Pr(c_i \mid a_2b_3) + \Pr(c_i \mid a_3b_1) - \Pr(c_i \mid a_3b_2) + \Pr(c_i \mid a_3b_3) = 2.0$. *Table 9.1(b) lists all quantitative parental synergies $Y^\star(a_jb_k, c_i), j, k = 1, 2, 3$.*

Table 9.1: The conditional probabilities $\Pr(c_i \mid AB)$ for a node C with the ternary parents A and B (a), and the matching quantitative parental synergies (b).

(a)

$\Pr(c_i \mid AB)$	a_1	a_2	a_3
b_1	0.7	0.2	0.3
b_2	0.2	1.0	0.8
b_3	0.4	0.1	0.9

(b)

$Y^\star(AB, c_i)$	a_1	a_2	a_3
b_1	2.4	0.4	-0.6
b_2	-1.2	2.0	-0.2
b_3	0.8	-0.4	1.4

From the definition of quantitative parental synergy, it is readily seen that for a binary parent A^k of C, we have that $Y^\star_\mathbf{x}(\mathbf{a}a^k, c_i) = -Y^\star_\mathbf{x}(\mathbf{a}\bar{a}^k, c_i)$ for any value assignments $\mathbf{a}$ and $\mathbf{x}$. For a node C with binary parents only, therefore, the quantitative parental synergy of just a single combination of parent values with respect to a particular value c_i of C, uniquely determines the other parental synergies with respect to c_i. From the definition it further follows for a binary parent A^k that $Y^\star_\mathbf{x}(\mathbf{a}a^k_m, c_i) = Y^\star_{\mathbf{x}a^k_m}(\mathbf{a}, c_i) - Y^\star_{\mathbf{x}a^k_n}(\mathbf{a}, c_i)$.

Example 9.2 *Consider an arbitrary-valued node C with the ternary parent A and the binary parent B; the conditional probabilities for the value c_i of C given A and B, are listed in Table 9.2(a); the matching quantitative parental synergies are listed in Table 9.2(b). It is easily verified that $Y^\star(Ab, c_i)$ equals $-Y^\star(A\bar{b}, c_i)$ for all possible values of A. Furthermore, from for example* $Y^\star(a_1b, c_i) = (0.8 - 0.3 - 0.5) - (0 - 0.4 - 0.9) = 1.3$, $Y^\star_b(a_1, c_i) = 0.8 - 0.3 - 0.5 = 0$ *and* $Y^\star_{\bar{b}}(a_1, c_i) = 0 - 0.4 - 0.9 = -1.3$, *it is readily verified that* $Y^\star(a_1b, c_i) = Y^\star_b(a_1, c_i) - Y^\star_{\bar{b}}(a_1, c_i)$.

Table 9.2: The conditional probabilities $\Pr(c_i \mid AB)$ for a node C with the ternary parent A and the binary parent B (a) and the matching quantitative parental synergies (b).

(a)

$\Pr(c_i \mid AB)$	a_1	a_2	a_3
b	0.8	0.3	0.5
$\bar{b}$	0.0	0.4	0.9

(b)

$Y^\star(AB, c_i)$	a_1	a_2	a_3
b	1.3	-0.5	-1.1
$\bar{b}$	-1.3	0.5	1.1

9.2 The Prior Convergence Error in Simple Binary Loops

Recall from Chapter 8 that upon applying Pearl's propagation algorithm to a Bayesian network with loops, convergence errors may arise in the probabilities computed for the network's convergence nodes. It was argued that this type of error originates in the computation of the compound causal parameter for a convergence node. More specifically, Pearl's rule for computing this parameter explicitly builds on the assumption that the parents $\mathbf{U}$ of a node X are mutually independent. While this assumption holds for the singly connected networks for which the algorithm was developed, it does not hold for multiply connected networks in general, however. Violation of the independence assumption underlying the computation rule now may cause a convergence error to arise in the compound causal parameter. An erroneous compound parameter in turn results in an error in the probabilities computed for the convergence node. In this section, the prior convergence error is studied for the convergence node of a binary simple loop in a network with a single loop. In Section 9.3, the results are extended to loops with arbitrary-valued nodes and to more complex loops.

9.2.1 An Expression for the Prior Convergence Error

To study the prior error in the probabilities computed for a convergence node, the example network from Figure 9.1 is considered. This network is composed of a single simple loop with binary nodes. An expression is derived for the prior convergence error in the probability $\widetilde{\Pr}(c_i)$ of the value c_i of the loop's convergence node C in terms of the specification of the network. This expression then is used to study the various factors that govern the size of the error.

When Pearl's propagation algorithm is applied to the network from Figure 9.1, nodes A and B send the causal messages $\pi_C(A) = \Pr(A)$ and $\pi_C(B) = \Pr(B)$ to node C. Upon receiving these messages, node C establishes the following value for its compound causal parameter:

$$\begin{aligned}\widetilde{\pi}(c_i) &= \sum_{A,B} \Pr(c_i \mid AB) \cdot \pi_C(A) \cdot \pi_C(B) \\ &= \sum_{A,B} \Pr(c_i \mid AB) \cdot \Pr(A) \cdot \Pr(B)\end{aligned}$$

Since in the prior state of the network all diagnostic messages are uninformative constants and therefore do not influence the computed probabilities, the algorithm yields $\widetilde{\Pr}(c_i) = \widetilde{\pi}(c_i)$. The exact probability $\Pr(c_i)$, however, equals

$$\begin{aligned}\Pr(c_i) &= \sum_{A,B} \Pr(c_i \mid AB) \cdot \Pr(AB) \\ &= \sum_{A,B,D} \Pr(c_i \mid AB) \cdot \Pr(AB \mid D) \cdot \Pr(D) \\ &= \sum_{A,B,D} \Pr(c_i \mid AB) \cdot \Pr(A \mid D) \cdot \Pr(B \mid D) \cdot \Pr(D)\end{aligned}$$

Note that the expression for the exact probability $\Pr(c_i)$ correctly captures the dependence between A and B, while the expression for the computed probability $\widetilde{\Pr}(c_i)$ does not. By simply subtracting $\widetilde{\Pr}(c_i)$ from $\Pr(c_i)$, the following expression is found for the prior convergence error v_i for the value c_i of node C:

$$v_i = \Pr(c_i) - \widetilde{\Pr}(c_i) = l \cdot m \cdot n \cdot w$$

where

$$\begin{aligned}l &= \Pr(d) - \Pr(d)^2 \\ m &= \Pr(a \mid d) - \Pr(a \mid \bar{d}) \\ n &= \Pr(b \mid d) - \Pr(b \mid \bar{d}) \\ w &= Y^{\star}(ab, c_i)\end{aligned}$$

Note that the expression for the convergence error includes terms such as $\Pr(a \mid d) \cdot \Pr(b \mid \bar{d})$, in which mutually exclusive probabilistic contexts are combined. Note furthermore that the convergence error pertains to a specific value of the convergence node. The probabilities computed for the values of an arbitrary-valued convergence node may thus include different convergence errors. For a binary node C with the values c_i and c_j, however, is found that $v_i = -v_j$. In the sequel, as long as no ambiguity can occur, the prior convergence error will be denoted by v.

The following example illustrates the expression derived for the prior convergence error.

Example 9.3 *Consider again the network from Figure 9.1. For the value c of the convergence node C, the four terms in the expression for the prior convergence error equal $l = 0.25$, $m = -0.6$, $n = -0.8$ and $w = 2$. With these terms, a prior convergence error of $0.25 \cdot -0.6 \cdot -0.8 \cdot 2 = 0.24$ is found. With Pearl's propagation algorithm, the approximate probability $\widetilde{\Pr}(c) = 0.54$ is computed. Using the above expression for the convergence error, the exact probability $\Pr(c) = 0.78$ can indeed be reconstructed from $\Pr(c) = \widetilde{\Pr}(c) + 0.24 = 0.78$.*

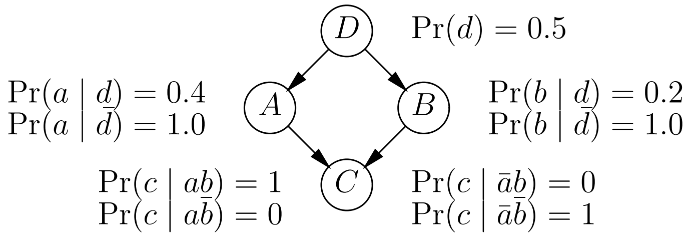

D
$\Pr(d) = 0.5$
A
$\Pr(a \mid d) = 0.4$
$\Pr(a \mid \bar{d}) = 1.0$
B
$\Pr(b \mid d) = 0.2$
$\Pr(b \mid \bar{d}) = 1.0$
C
$\Pr(c \mid ab) = 1$
$\Pr(c \mid a\bar{b}) = 0$
$\Pr(c \mid \bar{a}b) = 0$
$\Pr(c \mid \bar{a}\bar{b}) = 1$

Figure 9.1: A multiply connected Bayesian network including a convergence node C with the dependent parents A and B.

9.2.2 The Four Factors of the Convergence Error

The four factors l, m, n and w of the prior convergence error derived in the previous section are illustrated graphically by means of the surface and line segment in Figure 9.2.

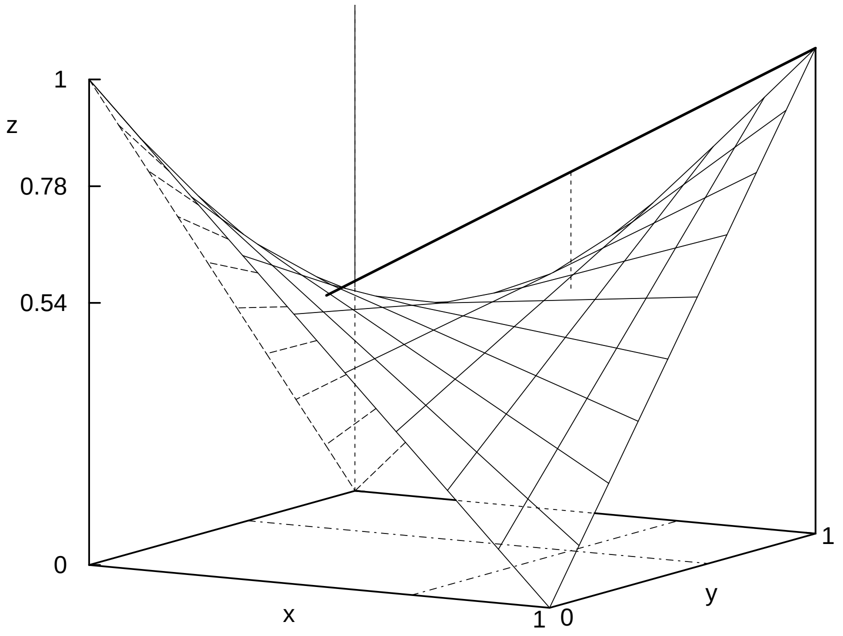


Figure 9.2: The line segment capturing $\Pr(c)$ and the surface capturing $\widetilde{\Pr}(c)$ for the example network from Figure 9.1.

The surface shown in the figure equals

$$z = 2 \cdot x \cdot y - x - y + 1$$

and captures the approximate probability $z = \widetilde{\Pr}(c)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$, given the conditional probabilities for node C of the example network. Note that for a fixed

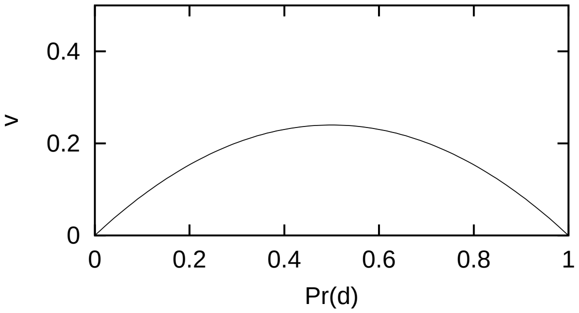


Figure 9.3: The prior convergence error v as a function of $\Pr(d)$, for the example network from Figure 9.1.

value of x, The function z is linear in y, and that for a fixed value of y, the function z is linear in x. The approximate probability $\widetilde{\Pr}(c)$ that is computed for the example network is found at $\pi_C(a) = 0.7$ and $\pi_C(b) = 0.6$, and equals 0.54. Figure 9.2 further shows a line segment that has its two endpoints on the depicted surface. This line segment expresses the exact probability $\Pr(c)$ as a function of $\Pr(d)$ given the conditional probabilities for the nodes A, B and C of the example network. The line segment equals

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.60 \\ -0.80 \\ -0.46 \end{bmatrix} \cdot \Pr(d) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

The two endpoints of the segment are found for $\Pr(d) = 0$ and $\Pr(d) = 1$, and hence at $\pi_C(a) = 1$ and $\pi_C(b) = 1$ and at $\pi_C(a) = 0.4$ and $\pi_C(b) = 0.2$, respectively. The exact probability $\Pr(c)$ for the example network is found at $\Pr(d) = 0.5$, hence at $\pi_C(a) = 0.7$ and $\pi_C(b) = 0.6$, and equals 0.78. While the surface depicts the probability of c under the assumption of independence of A and B, the line segment takes the dependence between these two nodes into consideration. The convergence error now equals the distance between the point on the line segment that matches $\Pr(d)$ and its orthogonal projection on the surface. For the example network, the difference between $\widetilde{\Pr}(c)$ and $\Pr(c)$ is indicated by the vertical dotted line segment in the figure.

The different factors that govern the size of the prior convergence error are algebraically independent. The four factors therefore do not have any interaction effects and can be studied independently:

- The factor $l = \Pr(d) - \Pr(d)^2$ is related to the location of the exact probability $\Pr(c)$ on the line segment which expresses $\Pr(c)$ as a function of $\Pr(d)$. Note that at the two endpoints of the segment, for $\Pr(d) = 0$ and $\Pr(d) = 1$, the factor l equals zero and the convergence error is zero. For $\Pr(d) = 0$ and $\Pr(d) = 1$, indeed, the nodes A and B in the example network are independent, causing the assumption underlying Pearl's computation rule to hold. From its first-order derivative $l' = 1 - 2 \cdot \Pr(d)$ it further follows that the factor l has an extreme at $\Pr(d) = 0.5$; this extreme equals $l = 0.25$. A probability

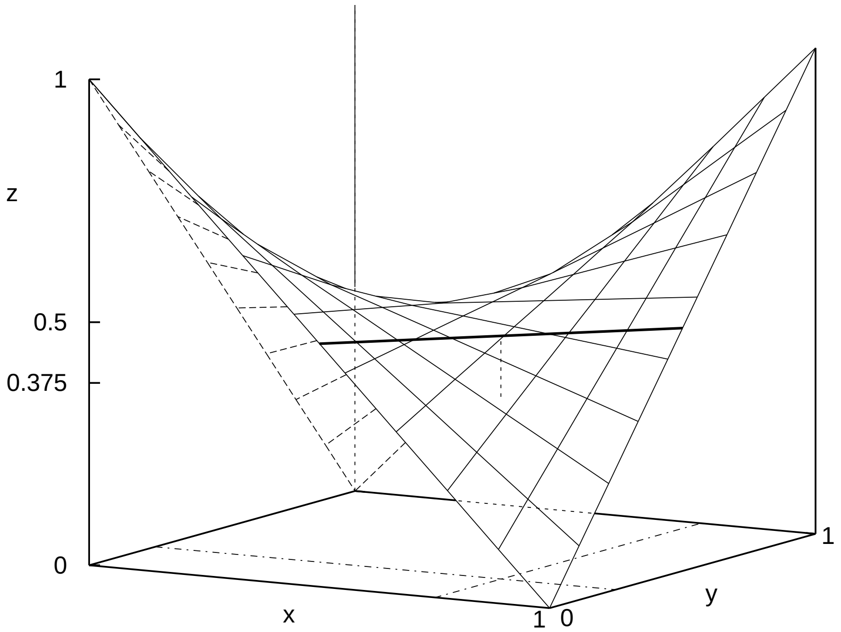


Figure 9.4: The line segment capturing $\Pr(c)$ and the surface capturing $\widetilde{\Pr}(c)$ for the example network from Figure 9.1 with the alternative conditional probabilities $\Pr(a \mid d) = 0.9$ and $\Pr(a \mid \bar{d}) = 0.9$ for node A and $\Pr(b \mid d) = 0.5$ and $\Pr(b \mid \bar{d}) = 0$ for node B.

$\Pr(d) = 0.5$, that is, a location of the exact probability $\Pr(c)$ on the mid-point of the line segment thus maximises the prior convergence error. Figure 9.2, already shows the effect on the convergence error of changing the probability $\Pr(d)$. In addition, Figure 9.3 explicitly depicts the size of the prior convergence error as a function of $\Pr(d)$; its functional form is $v = 0.96 \cdot \big(\Pr(d) - \Pr(d)^2\big)$.

- The factors $m = \Pr(a \mid d) - \Pr(a \mid \bar{d})$ and $n = \Pr(b \mid d) - \Pr(b \mid \bar{d})$ are related to the orientation of the line segment expressing $\Pr(c)$. If $\Pr(a \mid d) = \Pr(a \mid \bar{d})$, for example, the line segment is oriented parallel to the y-axis describing $\pi_C(b)$. Since $\widetilde{\Pr}(c)$ is linear in y for a fixed value of x, the segment then is located on the surface, which implies a convergence error equal to zero for all $\Pr(d)$. With $\Pr(a \mid d) = \Pr(a \mid \bar{d})$, indeed, $m = 0$ and the convergence error is zero. Note that $\Pr(a \mid d) = \Pr(a \mid \bar{d})$ in fact implies that A is independent of D which causes the assumption underlying Pearl's computation rule to hold. Similar observations apply to the factor m. The product of the factors m and n ranges between -1 and 1. The maximum of 1, for example, is found for $\Pr(a \mid d) = 1$, $\Pr(a \mid \bar{d}) = 0$, $\Pr(b \mid d) = 1$ and $\Pr(b \mid \bar{d}) = 0$. The line segment then is parallel to one of the diagonals of the x, y-plane, and the distance between the surface and the line segment is maximal. In the example network, $m = -0.6$, $n = -0.8$ and $m \cdot n = 0.48$. Figure 9.4 shows the effect of a change of the conditional probabilities for node A to $\Pr(a \mid d) = 1.0$ and $\Pr(a \mid \bar{d}) = 0.5$ and for node B to $\Pr(b \mid d) = 0.5$ and $\Pr(b \mid \bar{d}) = 0$ on the

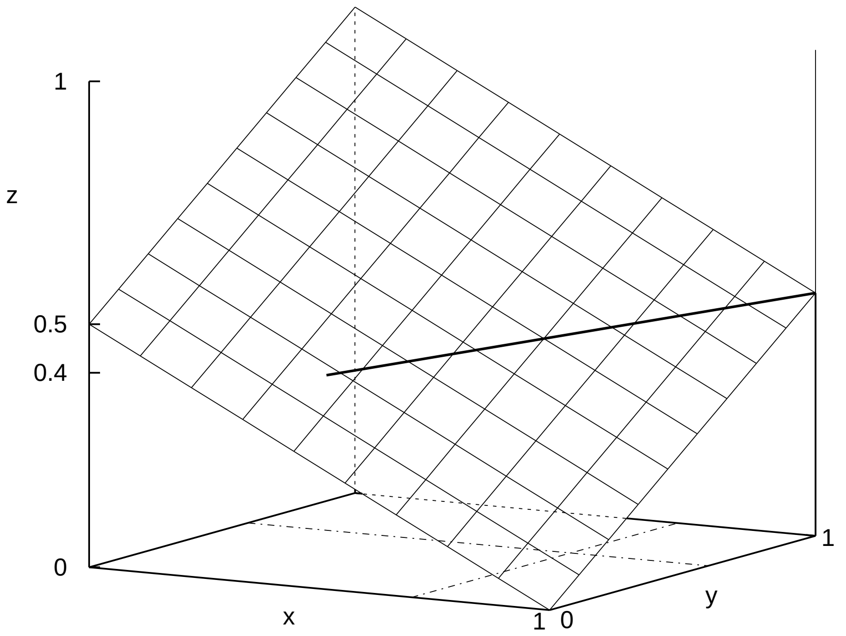


Figure 9.5: The line segment capturing $\Pr(c)$ and the surface capturing $\widetilde{\Pr}(c)$ for the example network from Figure 9.1 with the alternative conditional probabilities $\Pr(c \mid ab) = 0.5$, $\Pr(c \mid a\bar{b}) = 0$, $\Pr(c \mid \bar{a}b) = 1$ and $\Pr(c \mid \bar{a}\bar{b}) = 0.5$ for node C.

orientation of the line segment. For the line segment now applies that $m = 0.5$, $n = 0.5$ and $m \cdot n = 0.25$. Note that, compared to Figure 9.2, the distance between the surface and the point with the exact probability $\Pr(c)$ on the line segment indeed has decreased.

- The factor $w = Y^{\star}(ab, c_i)$ is related to the curvature of the surface and ranges between -2 and 2. The higher the absolute value of w, the more curved the surface is and the more curved the surface is, the larger the distance between a point on the line segment and its projection on the surface can be. For the example network, $w = 2$ and the curvature of the surface is maximal. Figure 9.5 shows the effect of changing the conditional probabilities for node C to $\Pr(c \mid ab) = 0.5$, $\Pr(c \mid a\bar{b}) = 0$, $\Pr(c \mid \bar{a}b) = 1$ and $\Pr(c \mid \bar{a}\bar{b}) = 0.5$. The factor w now equals zero and the surface is a plane:

$$z = -0.5 \cdot x + 0.5 \cdot y + 0.5$$

From $w = 0$, the prior convergence error is found to be equal to zero. The line segment expressing $\Pr(c)$ as a function of $\Pr(d)$ now is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.60 \\ -0.80 \\ -0.10 \end{bmatrix} \cdot \Pr(d) + \begin{bmatrix} 1.0 \\ 1.0 \\ 0.5 \end{bmatrix}$$

and in fact lies on the surface.

Informally speaking, the factors l, m and n with each other capture the degree of dependence between the nodes A and B along the trail through node D and the factor w indicates to what extent this dependence can affect the computed probabilities.

In general, for a Bayesian network with the same graphical structure as the example network from Figure 9.1, the surface z that captures the approximate probability $\widetilde{\Pr}(c)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ equals

$$\begin{aligned} z \;=\; & \big(\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})\big) \cdot x \cdot y \\ & + \big(\Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}\bar{b})\big) \cdot x \\ & + \big(\Pr(c \mid \bar{a}b) - \Pr(c \mid \bar{a}\bar{b})\big) \cdot y \\ & + \Pr(c \mid \bar{a}\bar{b}) \end{aligned}$$

Thus also in general, for a fixed value of x, the function z is linear in y and for a fixed value of y, it is linear in x. Now, provided that the surface is not a plane, that is, provided that $\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b}) \neq 0$, it will have a saddle point. At this point neither a change in x nor a change in y will influence the value found for z. In Section 9.5 is observed that the location of this saddle point influences the convergence behaviour of the causal messages from nodes A and B to node C.

To establish the saddle point of the surface, the partial derivatives of the function z with respect to x and with respect to y are established. The partial derivative of z with respect to x equals

$$\begin{aligned} \frac{\partial z}{\partial x} &= \big(\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})\big) \cdot y + \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}\bar{b}) \\ &= Y^{\star}(ab, c) \cdot y + Y^{\star}_{\bar{b}}(a, c) \end{aligned}$$

This derivative is a constant function in x which equals zero for

$$y = \frac{-Y^{\star}_{\bar{b}}(a, c)}{Y^{\star}(ab, c)}$$

The partial derivative of z with respect to y equals

$$\begin{aligned} \frac{\partial z}{\partial y} &= \big(\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})\big) \cdot x + \Pr(c \mid \bar{a}b) - \Pr(c \mid \bar{a}\bar{b}) \\ &= Y^{\star}(ab, c) \cdot x + Y^{\star}_{\bar{a}}(b, c) \end{aligned}$$

This derivative is a constant function in y which equals zero for

$$x = \frac{-Y^{\star}_{\bar{a}}(b, c)}{Y^{\star}(ab, c)}$$

The point of the surface at which both partial derivatives equal zero thus is found at

$$(x, y) = \left(\frac{-Y^{\star}_{\bar{a}}(b, c)}{Y^{\star}(ab, c)}, \frac{-Y^{\star}_{\bar{b}}(a, c)}{Y^{\star}(ab, c)} \right)$$

From the determinant of the Hessian matrix of z, det $H(z) = -(Y^{\star}(ab, c))^2$, being negative given that $Y^{\star}(ab, c) \neq 0$, it follows that the point $\left(\frac{-Y^{\star}_{\bar{a}}(b,c)}{Y^{\star}(ab,c)}, \frac{-Y^{\star}_{\bar{b}}(a,c)}{Y^{\star}(ab,c)}\right)$, indeed is a saddle point given that $Y^{\star}(ab, c) \neq 0$. Note that the x,y-coordinates of the saddle point are not necessarily found within the interval $[0, 1] \times [0, 1]$. This point thus may be unfeasible as $\widetilde{\Pr}(c)$.

9.2.3 The Extremes of the Convergence Error

From the analysis of the four factors in the previous section, it follows that the prior convergence error ranges between -0.5 and 0.5. The size that convergence error can actually adopt, however, is restricted by the exact probability $\Pr(c)$. Obviously, it must hold that $\widetilde{\Pr}(c) = \big(\Pr(c) - v\big) \in [0, 1]$, but the size of the convergence error for a given $\Pr(c)$ is even further restricted. Consider again the example network from Figure 9.1. To establish the relationship between $\Pr(c)$ and the maximum v_{max} of the prior convergence error, (conditional) probabilities have to be found for the nodes A, B, C and D which result in $\Pr(c)$ and at the same time maximise the convergence error. In terms of the graphical representation of the convergence error from Figure 9.2, given a specific value of $\Pr(c)$, the line segment and the surface have to be constructed in such a way that the distance between the point with the exact probability on the line segment and its orthogonal projection on the surface is maximal. Using Mathematica, it was established experimentally for a wide range of values of $\Pr(c)$, that the convergence error could attain its maximum with $\Pr(a \mid d) = 1$, $\Pr(a \mid \bar{d}) = 0$, $\Pr(b \mid d) = 1$, $\Pr(b \mid \bar{d}) = 0$, $\Pr(c \mid a\bar{b}) = 0$, $\Pr(c \mid \bar{a}b) = 0$ and $\Pr(c \mid \bar{a}\bar{b}) = 1$; the probabilities $\Pr(c \mid ab)$ and $\Pr(d)$ then are taken as variables in the remaining optimisation problem. With these conditional probabilities, it is found that $\Pr(c) = \Pr(c \mid ab) \cdot \Pr(d) - \Pr(d) + 1$ and that $v = \big(\Pr(c \mid ab) + 1\big) \cdot \big(\Pr(d) - \Pr(d)^2\big)$. The problem of finding the maximum prior convergence error v_{max} for a given value of $\Pr(c)$ now reduces to finding the maximum of

$$v = \big(\Pr(c \mid ab) + 1\big) \cdot \big(\Pr(d) - \Pr(d)^2\big)$$

under the constraints

$$\begin{aligned}
&\Pr(c) = \Pr(c \mid ab) \cdot \Pr(d) - \Pr(d) + 1 \\
&\Pr(d) \in [0, 1] \\
&\Pr(c \mid ab) \in [0, 1]
\end{aligned}$$

From the first constraint, it follows that

$$\Pr(c \mid ab) = \frac{\Pr(c) - 1 + \Pr(d)}{\Pr(d)}$$

Provided that $\Pr(d) \neq 0$[1], it now follows with the third constraint $\Pr(c \mid ab) \geq 0$, that $\Pr(d) \geq 1 - \Pr(c)$. Note that the constraint $\Pr(c \mid ab) \leq 1$ is fulfilled for all $\Pr(c) \in [0, 1]$. The prior

[1] Note that the case where $\Pr(d) = 0$ need not be further considered, since with this value there effectively is no loop present and the convergence error equals zero regardless of the conditional probabilities specified for the other nodes.

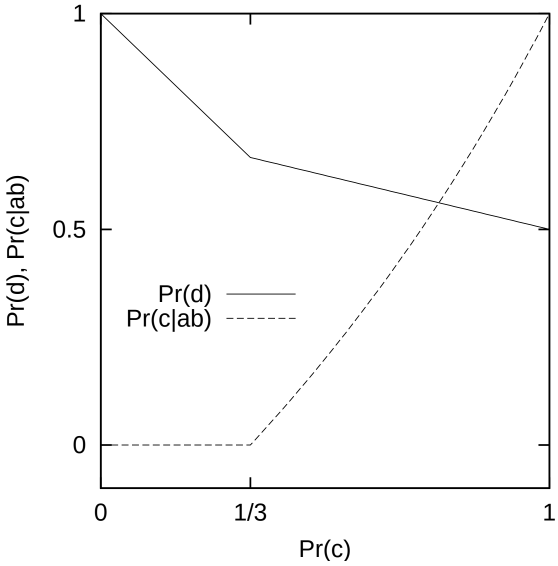


Figure 9.6: The combinations of values of $\Pr(d)$ and $\Pr(c \mid ab)$ for which a maximum prior convergence error is found, as a function of $\Pr(c)$.

convergence error $v = \big(\Pr(c \mid ab)+1\big) \cdot \big(\Pr(d) - \Pr(d)^2\big)$ thus effectively needs to be maximised under the constraints

$$\begin{aligned}&\Pr(c) = \Pr(c \mid ab) \cdot \Pr(d) - \Pr(d) + 1\\ &\Pr(d) \in [1 - \Pr(c), 1]\end{aligned}$$

Substituting $\Pr(c \mid ab) = \frac{\Pr(c)-1+\Pr(d)}{\Pr(d)}$ in the expression for the convergence error v and subsequently taking the first partial derivative of v with respect to $\Pr(d)$ gives

$$\frac{\partial v}{\partial \Pr(d)} = -4 \cdot \Pr(d) - \Pr(c) + 3$$

The partial derivative of v equals zero for $\Pr(d) = \frac{1}{4} \cdot \big(3 - \Pr(c)\big)$; it is positive for smaller values of $\Pr(d)$ and negative for larger values. At $\Pr(d) = \frac{1}{4} \cdot \big(3 - \Pr(c)\big)$, therefore, the convergence error v attains its maximum, provided that $\frac{1}{4} \cdot \big(3 - \Pr(c)\big) \in [1 - \Pr(c), 1]$. For all values of $\Pr(c)$, it holds that $\frac{1}{4} \cdot \big(3 - \Pr(c)\big) \leq 1$. The constraint $\frac{1}{4} \cdot \big(3 - \Pr(c)\big) \geq 1 - \Pr(c)$, on the other hand, is only satisfied for $\Pr(c) \geq \frac{1}{3}$. For values of $\Pr(c)$ smaller than $\frac{1}{3}$, the value for $\Pr(d)$ at which v_{max} would be found thus is smaller than allowed for by the constraint $\Pr(d) \geq \big(1 - \Pr(c)\big)$. The maximum of v will then be found at the smallest value allowed for $\Pr(d) = 1 - \Pr(c)$. At $\Pr(d) = \frac{1}{4} \cdot \big(3 - \Pr(c)\big)$ now is found that $v = \frac{1}{8} \cdot \big(1 + \Pr(c)\big)^2$ and at $\Pr(d) = 1 - \Pr(c)$ is found that $v = \Pr(c) - \Pr(c)^2$. Expressed as a function of $\Pr(c)$, the maximum prior convergence error v_{max} thus equals

$$v_{max} = \begin{cases} \Pr(c) - \Pr(c)^2 & \text{if } \Pr(c) < \frac{1}{3} \\ \frac{1}{8} \cdot \big(\Pr(c) + 1\big)^2 & \text{if } \Pr(c) \geq \frac{1}{3} \end{cases}$$

Note that at $\Pr(c) = \frac{1}{3}$, both functions in the expression above yield the value $\frac{2}{9}$. At $\Pr(c) = \frac{1}{3}$, moreover, the first derivatives of the two functions yield the value $\frac{1}{3}$. The function defined for v_{max} thus is continuously differentiable on the entire interval $[0, 1]$ for $\Pr(c)$. Note furthermore that with $\Pr(d) = \frac{1}{4} \cdot \big(3 - \Pr(c)\big)$, the matching value of $\Pr(c \mid ab)$ is $\Pr(c \mid ab) = \frac{\Pr(c)-1+\Pr(d)}{\Pr(d)} = \frac{3 \cdot \Pr(c)-1}{-\Pr(c)+3}$; with $\Pr(d) = 1 - \Pr(c)$, the matching value $\Pr(c \mid ab) = 0$ is found. Figure 9.6 shows, as a function of $\Pr(c)$, the combinations of values of $\Pr(d)$ and $\Pr(c \mid ab)$ that yield a maximum prior convergence error.

Analogously, the relationship between the minimum convergence error v_{min} and $\Pr(c)$ is found to be

$$v_{min} = \begin{cases} -\frac{1}{8} \cdot \big(\Pr(c) - 2\big)^2 & \text{if } \Pr(c) \leq \frac{2}{3} \\ -\Pr(c) + \Pr(c)^2 & \text{if } \Pr(c) > \frac{2}{3} \end{cases}$$

At $\Pr(c) = \frac{2}{3}$, both functions in the expression above yield the value $-\frac{2}{9}$. At $\Pr(c) = \frac{2}{3}$, moreover, the first derivatives of the two functions yield the value $\frac{1}{3}$. The function defined for v_{min} thus is continuously differentiable on the entire interval $[0, 1]$ for $\Pr(c)$ as well. Figure 9.7(a) shows the four functions which feature in the expressions for v_{max} and v_{min} given above; Figure 9.7(b) depicts the functions v_{max} and v_{min} which define the area of feasible combinations of $\Pr(c)$ and v. Note that the two functions v_{max} and v_{min} are each other's point reflections in the point $(\Pr(c), v) = (0, 0.5)$, that is, the value found for v_{max} at $\Pr(c)$ equals $-v_{min}$ at $1 - \Pr(c)$.

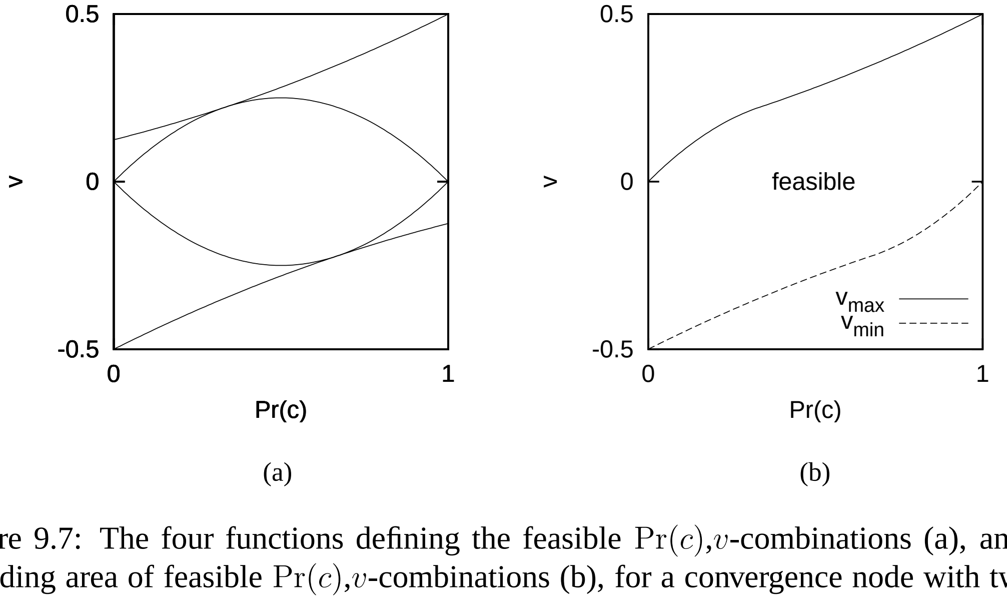


Figure 9.7: The four functions defining the feasible $\Pr(c)$,v-combinations (a), and the corresponding area of feasible $\Pr(c)$,v-combinations (b), for a convergence node with two loop parents.

The combination of conditional probabilities which was chosen above to find the maximum prior convergence error v_{max} as a function of $\Pr(c)$ is not unique. The maximum v_{max} is also found, for example, with the conditional probabilities $\Pr(a \mid d) = 1$, $\Pr(a \mid \bar{d}) = 0$, $\Pr(b \mid d) = 1$, $\Pr(b \mid \bar{d}) = 0$, $\Pr(c \mid a\bar{b}) = 0$, $\Pr(c \mid \bar{a}b) = 0$ and $\Pr(c \mid ab) = 1$ where the probabilities $\Pr(d)$ and $\Pr(c \mid \bar{a}\bar{b})$ are taken as variables in the remaining optimisation problem. Due to its symmetry, this problem is solved in the same way as above. A similar observation holds for all related combinations of conditional probabilities for the nodes involved.

9.2.4 An Alternative Expression for the Prior Convergence Error

To conclude the analysis of the error which arises at the convergence node of a simple loop, an alternative expression is proposed. Consider the convergence node C of the simple loop from the example network from Figure 9.1, and assume again that all nodes involved are binary. The convergence error $v_i = \Pr(c_i) - \widetilde{\Pr}(c_i)$ can now also be written as

$$v_i = (s - t) \cdot w$$

where w is as before and

$$s = \sum_D \Pr(a \mid D) \cdot \Pr(b \mid D) \cdot \Pr(D)$$

$$t = \left(\sum_D \Pr(a \mid D) \cdot \Pr(D)\right) \cdot \left(\sum_D \Pr(b \mid D) \cdot \Pr(D)\right)$$

The degree of dependency between the nodes A and B is now captured by the factor $s-t$ instead of by the factor $l \cdot m \cdot n$. Note that in this expression the term s equals $\Pr(ab)$ and the term t equals $\Pr(a) \cdot \Pr(b)$. The term s, therefore, correctly captures the dependency between the nodes A and B, where the term t assumes independence upon representing the joint probability distribution over these nodes.

The alternative expression for the prior convergence error presented above is more apt for generalisation and will be the starting point in the following section where it will be extended to apply to variables with more than two values and to more complex loops.

9.3 Generalising the Prior Convergence Error

In the previous section the prior convergence error was studied for the Bayesian network with a single simple loop with binary nodes from Figure 9.1. The results of the analysis are readily extended to networks including a single simple loop with additional inner nodes and to networks with nodes outside the loop. For such loops, (some of) the factors that determine the convergence error are not directly available. These factors can be established from the network's specification, however. For example, if, in the network from Figure 9.1, node A has another parent E in addition to its parent D, then the probabilities $\Pr(a \mid D)$, which are required for determining the size of the convergence error, are not specified directly in the network but can be established from the specified probabilities $\Pr(a \mid DE)$, using the observation that nodes D and E are a priori independent. The analysis further trivially holds for networks including a single simple loop with more than one convergence node. From the d-separation criterion, it follows that for such a loop, the loop parents of each convergence node are independent in the network in its prior state. Effectively, therefore, no loop is present and no convergence error will arise. The analysis, however, does not hold for loops including non-binary nodes nor for compound loops. In following sections, the expression for the prior convergence error is successively generalised to simple loops with nodes with more than two values, to compound loops with binary nodes, and to compound loops with nodes with more than two values. As the results for binary simple loops, the results in these sections are readily extended to the discussed loops with additional inner nodes and to the discussed loops with nodes outside the loop.

9.3.1 Simple Loops with Multiple-valued Nodes

To generalise the expression for the prior convergence error to networks including a single simple loop with nodes with an arbitrary number of values, the first step is the observation that the derivation from Section 9.2.1 applied to a single value c_i of the convergence node C. Since the other value of C was not involved in the derivation, the derived expression directly applies to a non-binary convergence node. Furthermore, from the alternative expression for the convergence error given in Section 9.2.4, it is readily seen that the expression is not restricted to just a binary joint ancestor D.

Now consider the parent nodes A and B of the convergence node C of the simple loop under study. It is posed as a conjecture, supported by experimental results, that the following expression captures the prior convergence error for the convergence node of a simple loop in which also the inner loop nodes A and B may have more than two values:

$$v_i = \sum_{A,B} (s_{AB} - t_{AB}) \cdot w(AB)/4$$

where

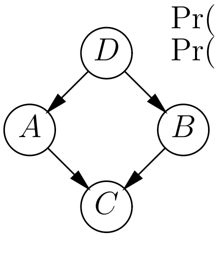
$$\begin{aligned}
s_{AB} &= \sum_D \Pr(A \mid D) \cdot \Pr(B \mid D) \cdot \Pr(D) \\
t_{AB} &= \left(\sum_D \Pr(A \mid D) \cdot \Pr(D) \right) \cdot \left(\sum_D \Pr(A \mid D) \cdot \Pr(D) \right) \\
w(AB) &= Y^\star(AB, c_i)
\end{aligned}$$

Note that, analogous to the binary case, the term s_{AB} equals $\Pr(AB)$ and thus accounts for the dependence between A and B through D and that the term t_{AB} equals $\Pr(A) \cdot \Pr(B)$ and thus captures $\Pr(AB)$ under the assumption of independence of A and B. In contrast with the binary case, now all different value combinations of the nodes A and B are considered. The impact on the convergence error of the dependency between a specific combination of values for the nodes A and B, is determined by the quantitative parental synergy of this combination with respect to the value c_i of the convergence node. Further note that the expression for the convergence error now includes a division by a constant. This constant equals 2^n, where n is the number of loop parents of the convergence node.

The following example illustrates the expression for the prior convergence error for a network with a single simple loop with multi-valued loop nodes.

Example 9.4 *Consider the example network from Figure 9.8. Table 9.3 lists the quantitative parental synergies for the convergence node* C*. In addition, Table 9.4 lists all terms* $s_{AB} - t_{AB}$ *pertaining to the loop parents* A *and* B *of the convergence node. Recall that such a term* $s_{AB} - t_{AB}$ *reflects the degree of dependency between the values* A *and* B *of the nodes* A *and* B*, respectively, along the trail* $A \leftarrow D \rightarrow B$*. The largest absolute value for this term is found for the combination of values* a_2 *and* b_2*, and equals* 0.1008*. The smallest absolute value is found for the combination of values* a_3 *and* b_3*, and equals* 0.0144*. Intuitively, the maximum term arising for the combination of values* a_2 *and* b_2 *may be explained by the differences between* $\Pr(a_2 \mid d_1)$ *and* $\Pr(a_2 \mid d_2)$ *and between* $\Pr(b_2 \mid d_1)$ *and* $\Pr(b_2 \mid d_2)$ *being maximal. The minimum arising for the combination of values* a_3 *and* b_3 *may be explained by the difference between* $\Pr(a_3 \mid d_1)$ *and* $\Pr(a_3 \mid d_2)$ *and between* $\Pr(b_3 \mid d_1)$ *and* $\Pr(b_3 \mid d_2)$ *being minimal. By exact inference the probability* $\Pr(c) = 0.59580$ *is established and upon loopy propagation, the approximate probability of* $\widetilde{\Pr}(c) = 0.49404$ *is found. The prior convergence error thus equals* $v = 0.10176$*. Using the expression above, the same error is established:*

$$
\begin{aligned}
v \;=\; & \Big(\sum_D \Pr(a_1 \mid D) \cdot \Pr(b_1 \mid D) \cdot \Pr(D) - \\
& \big(\sum_D \cdot \Pr(a_1 \mid D) \cdot \Pr(D)\big) \cdot \big(\sum_D \Pr(b_1 \mid D) \cdot \Pr(D)\big)\Big) \cdot w(a_1 b_1)/4 + \\
& \vdots \\
= \; & \big(\Pr(a_1 \mid d_1) \cdot \Pr(b_1 \mid d_1) \cdot \Pr(d_1) \\
& + \Pr(a_1 \mid d_2) \cdot \Pr(b_1 \mid d_2 \cdot \Pr(d_2)) \\
& - \Pr(a_1 \mid d_1) \cdot \Pr(d_1) \cdot \Pr(b_1 \mid d_1) \cdot \Pr(d_1) \\
& - \Pr(a_1 \mid d_1) \cdot \Pr(d_2) \cdot \Pr(b_1 \mid d_2) \cdot \Pr(d_1) \\
& - \Pr(a_1 \mid d_2) \cdot \Pr(d_1) \cdot \Pr(b_1 \mid d_1) \cdot \Pr(d_2) \\
& - \Pr(a_1 \mid d_2) \cdot \Pr(d_2) \cdot \Pr(b_1 \mid d_2) \cdot \Pr(d_2)\big) \cdot w(a_1 b_1)/4 + \\
& \vdots \\
= \; & \big(0.0384 \cdot 2.4 + (-0.0672) \cdot (-1.2) + 0.0288 \cdot 0.8 + \\
& (-0.0576) \cdot 0.4 + 0.1008 \cdot 2.0 + (-0.0432) \cdot (-0.4) + \\
& +0.0192 \cdot (-0.6) + (-0.0336) \cdot (-0.2) + 0.0144 \cdot 1.4\big)/4 \\
= \; & 0.10176
\end{aligned}
$$



$$\Pr(d_1) = 0.4 \qquad \Pr(d_2) = 0.6$$

$$\begin{aligned}
\Pr(a_1 \mid d_1) &= 0.2 & \Pr(b_1 \mid d_1) &= 0.1\\
\Pr(a_1 \mid d_2) &= 0.6 & \Pr(b_1 \mid d_2) &= 0.5\\
\Pr(a_2 \mid d_1) &= 0.7 & \Pr(b_2 \mid d_1) &= 0.8\\
\Pr(a_2 \mid d_2) &= 0.1 & \Pr(b_2 \mid d_2) &= 0.1\\
\Pr(a_3 \mid d_1) &= 0.1 & \Pr(b_3 \mid d_1) &= 0.1\\
\Pr(a_3 \mid d_2) &= 0.3 & \Pr(b_3 \mid d_2) &= 0.4
\end{aligned}$$

$$\begin{aligned}
\Pr(c \mid a_1b_1) &= 0.7 & \Pr(c \mid a_2b_1) &= 0.2 & \Pr(c \mid a_3b_1) &= 0.3\\
\Pr(c \mid a_1b_2) &= 0.2 & \Pr(c \mid a_2b_2) &= 1.0 & \Pr(c \mid a_3b_2) &= 0.8\\
\Pr(c \mid a_1b_3) &= 0.4 & \Pr(c \mid a_2b_3) &= 0.1 & \Pr(c \mid a_3b_3) &= 0.9
\end{aligned}$$

Figure 9.8: An example Bayesian network including a convergence node C with the dependent multi-valued parents A and B.

Table 9.3: The quantitative parental synergies with respect to the value c of the convergence node C for the example network from Figure 9.8.

$Y^\star(AB, c)$	a_1	a_2	a_3
b_1	2.4	0.4	-0.6
b_2	-1.2	2.0	-0.2
b_3	0.8	-0.4	1.4

Table 9.4: The terms $s_{AB} - t_{AB}$ for the example network from Figure 9.8.

		m		
	$s_{AB} - t_{AB}$	1	2	3
	1	0.0384	-0.0576	0.0192
n	2	-0.0672	0.1008	-0.0336
	3	0.0288	-0.0432	0.0144

In the previous section, the prior convergence error was found to range between -0.5 and 0.5 for simple loops with binary variables. The relationship between the true probability $\Pr(c)$ and the feasible maximum v_{max} and minimum v_{min} of the convergence error was found to be

$$v_{max} = \begin{cases} \Pr(c) - \Pr(c)^2 & \text{if } \Pr(c) < \frac{1}{3} \\ \frac{1}{8} \cdot \big(\Pr(c) + 1\big)^2 & \text{if } \Pr(c) \geq \frac{1}{3} \end{cases}$$

and

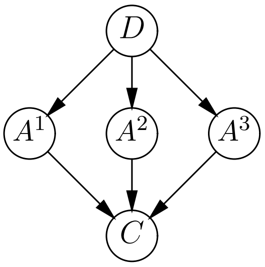
$$v_{min} = \begin{cases} -\frac{1}{8}\big(\Pr(c) - 2\big)^2 & \text{if } \Pr(c) \leq \frac{2}{3} \\ -\Pr(c) + \Pr(c)^2 & \text{if } \Pr(c) > \frac{2}{3} \end{cases}$$

It is now posed as a conjecture, that both the range and the feasible maximum and minimum of the convergence error remain unchanged for simple loops with non-binary variables. The conjecture was confirmed by preliminary experimental results.

9.3.2 Double Loops with Binary Nodes

In the previous section a general expression was proposed for the prior convergence error in networks with a single simple loop with nodes with an arbitrary number of values. In this section, the convergence error is studied in networks with a single double loop with binary nodes. A loop will be called a double loop if it can be transformed into a simple loop by the removal of a single arc. With respect to double loops there are, in essence, three possibilities: the double loop includes a single convergence node with three incoming loops arcs; the double loop has two non-consecutive convergence nodes, that is, the loop has two convergence nodes such non of the convergence node is an ancestor of the other one; or the double loop has two consecutive convergence nodes. Figure 9.9 shows an example of the first possibility; Figure 9.10 illustrates the second possibility and an example of the third possibility is shown in Figure 9.11. Note that for the second possibility, the convergence errors can be established for each convergence node separately with the expression from Section 9.2.1. In this section the other two possibilities will be adressed. First, an expression is derived for the prior convergence error in a double loop with a single convergence node. It then is shown that the same expression holds for the convergence error in a double loop with two consecutive convergence nodes.

$$
\begin{aligned}
\Pr(c \mid a^1a^2a^3) &= 0.65\\
\Pr(c \mid a^1a^2\bar{a}^3) &= 0.13\\
\Pr(c \mid a^1\bar{a}^2a^3) &= 0.35\\
\Pr(c \mid a^1\bar{a}^2\bar{a}^3) &= 0.31\\
\Pr(c \mid \bar{a}^1a^2a^3) &= 0.45\\
\Pr(c \mid \bar{a}^1a^2\bar{a}^3) &= 0.25\\
\Pr(c \mid \bar{a}^1\bar{a}^2a^3) &= 0.60\\
\Pr(c \mid \bar{a}^1\bar{a}^2\bar{a}^3) &= 0.16
\end{aligned}
$$



$$
\begin{aligned}
\Pr(d) &= 0.6\\
\Pr(a^1 \mid d) &= 0.8\\
\Pr(a^1 \mid \bar{d}) &= 0.1\\
\Pr(a^2 \mid d) &= 0.7\\
\Pr(a^2 \mid \bar{d}) &= 0.4\\
\Pr(a^3 \mid d) &= 0.2\\
\Pr(a^3 \mid \bar{d}) &= 0.6
\end{aligned}
$$

Figure 9.9: An example network with a graph including a double loop with a single convergence node C with three incoming loop arcs.

Consider the graph of the network depicted in Figure 9.9. The derivation of an expression for the prior convergence error in the probabilities computed for the convergence node C is analogous to the derivation from Section 9.2.1. Upon applying Pearl's propagation algorithm to the network from Figure 9.9, nodes A^1, A^2 and A^3 will send correct causal messages to node C. Upon receiving these messages, node C establishes the approximate compound causal parameter

$$
\begin{aligned}
\widetilde{\pi}(c_i) &= \sum_{A^1,A^2,A^3} \Pr(c_i \mid A^1A^2A^3) \cdot \pi_C(A^1) \cdot \pi_C(A^2) \cdot \pi_C(A^3)\\
&= \sum_{A^1,A^2,A^3} \Pr(c_i \mid A^1A^2A^3) \cdot \Pr(A^1) \cdot \Pr(A^2) \cdot \Pr(A^3)
\end{aligned}
$$

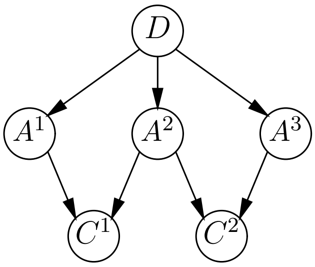


Figure 9.10: An example network including a double loop with two non-consecutive convergence nodes C^1 and C^2.

and the approximate probability $\widetilde{\Pr}(c_i) = \widetilde{\pi}(c_i)$. The exact probability $\Pr(c_i)$ equals

$$\Pr(c_i) = \sum_{A^1,A^2,A^3,D} \Pr(c_i \mid A^1A^2A^3) \cdot \Pr(A^1 \mid D) \cdot \Pr(A^2 \mid D) \cdot \Pr(A^3 \mid D) \cdot \Pr(D)$$

Subtracting $\widetilde{\Pr}(c_i)$ from $\Pr(c_i)$ and manipulating the resulting terms now results in the following expression for the convergence error:

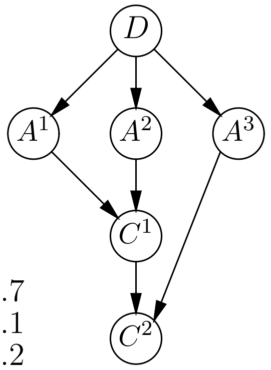
$$\begin{aligned} v_i &= (s_{a^1a^2a^3} - t_{a^1a^2a^3}) \cdot w + \\ &\quad (s_{a^2a^3} - t_{a^2a^3}) \cdot w_{\bar{a}^1}(a^2a^3) + (s_{a^1a^3} - t_{a^1a^3}) \cdot w_{\bar{a}^2}(a^1a^3) + (s_{a^1a^2} - t_{a^1a^2}) \cdot w_{\bar{a}^3}(a^1a^2) \end{aligned}$$

where

$$\begin{aligned} s_{a^1a^2a^3} &= \sum_D \prod_{i=1,2,3} \Pr(a^i \mid D) \cdot \Pr(D) \\ t_{a^1a^2a^3} &= \prod_{i=1,2,3} \sum_D \Pr(a^i \mid D) \cdot \Pr(D) \\ w &= Y^\star(a^1a^2a^3, c_i) \\ s_{a^ma^n} &= \sum_D \Pr(a^m \mid D) \cdot \Pr(a^n \mid D) \cdot \Pr(D) \\ t_{a^ma^n} &= \left(\sum_D \Pr(a^m \mid D) \cdot \Pr(D)\right) \cdot \left(\sum_D \Pr(a^n \mid D) \cdot \Pr(D)\right) \\ w_{\bar{a}^l}(a^ma^n) &= Y^\star_{\bar{a}^l}(a^ma^n, c_i) \end{aligned}$$

The convergence error is composed of the term $(s_{a^1a^2a^3} - t_{a^1a^2a^3}) \cdot w$ pertaining to the entire double loop, and the three terms $(s_{a^ma^n} - t_{a^ma^n}) \cdot w_{\bar{a}^l}(a^ma^n)$ which pertain to the three simple loops that are included within the double loop. Note that the expression above again involves

$$\begin{aligned}
\Pr(a^1 \mid d) &= 0.8\\
\Pr(a^1 \mid \bar{d}) &= 0.1\\
\Pr(a^2 \mid d) &= 0.7\\
\Pr(a^2 \mid \bar{d}) &= 0.4\\
\Pr(a^3 \mid d) &= 0.2\\
\Pr(a^3 \mid \bar{d}) &= 0.6
\end{aligned}$$



$$\Pr(d) = 0.6$$

$$\begin{aligned}
\Pr(c^1 \mid a^1 a^2) &= 0.9\\
\Pr(c^1 \mid a^1 \bar{a}^2) &= 0.3\\
\Pr(c^1 \mid \bar{a}^1 a^2) &= 0.5\\
\Pr(c^1 \mid \bar{a}^1 \bar{a}^2) &= 0.8
\end{aligned}$$

$$\begin{aligned}
\Pr(c^2 \mid c^1 a^3) &= 0.7\\
\Pr(c^2 \mid c^1 \bar{a}^3) &= 0.1\\
\Pr(c^2 \mid \bar{c}^1 a^3) &= 0.2\\
\Pr(c^2 \mid \bar{c}^1 \bar{a}^3) &= 0.4
\end{aligned}$$

Figure 9.11: A example network including a double loop with the consecutive convergence nodes C^1 and C^2 with two incoming loop arcs each.

just a single value of the convergence node and therefore is valid for multi-valued convergence nodes as well. Further note that the expression also provides for a multi-valued node D.

The expression for the prior convergence error for the network from Figure 9.9 is illustrated in the following example.

Example 9.5 *Consider the network from Figure 9.9. By exact inference the probability* $\Pr(c) = 0.317656$ *is established and upon loopy propagation the approximate probability of* $\widetilde{\Pr}(c) = 0.32035072$ *is found. The prior convergence error thus equals* $v = -0.00269472$. *Using the expression derived above, the same error is established. For the value* c *of the convergence node, the following terms are found:*

$$\begin{aligned}
(s_{a^1a^2a^3} - t_{a^1a^2a^3}) \cdot w &= -0.031776 \cdot 0.72 = -0.0228787\\
(s_{a^2a^3} - t_{a^2a^3}) \cdot w_{\bar{a}^1}(a^2a^3) &= -0.0288 \cdot -0.24 = 0.006912\\
(s_{a^1a^3} - t_{a^1a^3}) \cdot w_{\bar{a}^2}(a^1a^3) &= -0.0672 \cdot -0.40 = 0.02688\\
(s_{a^1a^2} - t_{a^1a^2}) \cdot w_{\bar{a}^3}(a^1a^2) &= 0.0504 \cdot -0.27 = -0.013608
\end{aligned}$$

from which the same prior convergence error of $v = -0.00269472$ *is computed.*

Now consider a network with a double loop with two consecutive convergence nodes, as depicted in Figure 9.11. The first convergence node C^1 effectively is the convergence node of a simple loop. For this node, therefore, the prior convergence error can be computed with the expression from Section 9.2.1. In contrast with C^1, the convergence node C^2 does not receive exact causal messages from both its parents on the loop; from its parent C^1, it receives a causal message that includes a convergence error. The formula from Section 9.2.1, therefore, does not capture the prior convergence error for node C^2. For node C^2, the approximate probability

$$\widetilde{\Pr}(c_i^2) = \sum_{C^1,A^3} \Pr(c_i^2 \mid C^1A^3) \cdot \widetilde{\pi}(C^1) \cdot \Pr(A^3)$$

is found with

$$\widetilde{\pi}(C^1) = \sum_{A^1,A^2} \Pr(C^1 \mid A^1A^2) \cdot \Pr(A^1) \cdot \Pr(A^2)$$

from which it follows that

$$\widetilde{\Pr}(c_i^2) \quad = \sum_{C^1,A^1,A^2,A^3} \Pr(c_i^2 \mid C^1A^3) \cdot \Pr(C^1 \mid A^1A^2) \cdot \Pr(A^1) \cdot \Pr(A^2) \cdot \Pr(A^3)$$

Because C^2 is independent of the nodes A^1 and A^2 given C^1 and A^3, and because C^1 is independent of A^3 given A^1 and A^2, the following property holds:

$$\sum_{C^1} \Pr(c_i^2 \mid C^1A^3) \cdot \Pr(C^1 \mid A^1A^2) = \Pr(c_i^2 \mid A^1A^2A^3)$$

Substituting the above expression for the appropriate terms in the expression for $\widetilde{\Pr}(c_i^2)$ now gives

$$\widetilde{\Pr}(c_i^2) \quad = \sum_{A^1A^2A^3} \Pr(c_i^2 \mid A^1A^2A^3) \cdot \Pr(A^1) \cdot \Pr(A^2) \cdot \Pr(A^3)$$

For the approximate probabilities for the convergence node C^2 in the network from Figure 9.11, therefore, the same expression is found as for the approximate probabilities for the convergence node C in the network from Figure 9.9. These nodes, moreover, share the same expression for their exact probabilities. As a consequence, the same expression is found for the prior convergence error for node C^2 as for node C. Note, however, that while the quantitative parental synergies in this expression were directly available from the network from Figure 9.9, they cannot be found in the specifications of the network from Figure 9.11. The quantitative synergies can, however, be computed using

$$\Pr(c_i^2 \mid A^1A^2A^3) = \sum_{C^1} \Pr(c_i^2 \mid C^1A^3) \cdot \Pr(C^1 \mid A^1A^2)$$

As noted, for the convergence error for C^1 in the network from Figure 9.11, the expressions from Section 9.2.1 can be used whereas for node C^2, the expression from this section is required. A convergence node thus has a certain degree of complexity with respect to the convergence error found in the probabilities computed for its values. In the sequel, with the complexity of a convergence node C the number of incoming arcs on the loop plus the number of convergence nodes that C has as ancestor in the loop will be indicated. The complexity degree of node C^1 in the network form Figure 9.11 thus equals two and the complexity degree of C^2 equals three.

In Section 9.2.3 a range was established for the prior error for the convergence node of a simple loop. Also, the feasible maximum and minimum values of the error were determined as functions of the probability $\Pr(c)$. To establish the general range of the prior convergence error for a double loop and to find its feasible values as a function of $\Pr(c)$, as similar analysis now is performed for the network from Figure 9.9. Again using Mathematica, it was established experimentally for a wide range of values of $\Pr(c)$ that the convergence error could attain its maximum by taking the probabilities $\Pr(d)$ and $\Pr(c \mid a^1a^2a^3)$ as variables in the optimisation problem with $\Pr(a^1 \mid d) = \Pr(a^2 \mid d) = \Pr(a^3 \mid d) = \Pr(c \mid \bar{a}^1\bar{a}^2\bar{a}^3) = 1$ and with the value zero for all other conditional probabilities specified in the network. With these probabilities it is found that $\Pr(c) = \Pr(c \mid a^1a^2a^3) \cdot \Pr(d) - \Pr(d) + 1$ and that $v = \big(\Pr(c \mid a^1a^2a^3) - 1\big) \cdot \big(\Pr(d) - \Pr(d)^3\big) + 3 \cdot \big(\Pr(d) - \Pr(d)^2\big)$. The problem of finding the maximum prior convergence error v_{max} for a given value of $\Pr(c)$ now reduces to finding the maximum of

$$v = \big(\Pr(c \mid a^1a^2a^3) - 1\big) \cdot \big(\Pr(d) - \Pr(d)^3\big) + 3 \cdot \big(\Pr(d) - \Pr(d)^2\big)$$

under the constraints

$$\begin{aligned}
&\Pr(c) = \Pr(c \mid a^1a^2a^3) \cdot \Pr(d) - \Pr(d) + 1\\
&\Pr(d) \in [0,1]\\
&\Pr(c \mid a^1a^2a^3) \in [0,1]
\end{aligned}$$

The function expressing the maximum prior convergence error v_{max} in terms of $\Pr(c)$ can now be derived in exactly the same way as for a convergence node with two parent loop nodes. As before, the constraints mentioned above can be rewritten as

$$\begin{aligned}
&\Pr(c) = \Pr(c \mid a^1a^2a^3) \cdot \Pr(d) - \Pr(d) + 1\\
&\Pr(d) \in [1 - \Pr(c), 1]
\end{aligned}$$

From the first constraint, it follows that

$$\Pr(c \mid a^1a^2a^3) = \frac{\Pr(c) - 1 + \Pr(d)}{\Pr(d)}$$

Substituting this term in the expression for the convergence error v and subsequently taking the first partial derivative of v with respect to $\Pr(d)$ gives

$$\frac{\partial v}{\partial \Pr(d)} = \big(-2 \cdot \Pr(c) - 4\big) \cdot \Pr(d) + 3$$

The partial derivative of v equals zero for $\Pr(d) = \frac{3}{2 \cdot \Pr(c) + 4}$; it is positive for smaller values of $\Pr(d)$ and negative for larger values. At $\Pr(d) = \frac{3}{2 \cdot \Pr(c) + 4}$, therefore, the convergence error v attains its maximum, provided that $\frac{3}{2 \cdot \Pr(c) + 4} \in [1 - \Pr(c), 1]$. For all values of $\Pr(c)$ it holds that $\frac{3}{2 \cdot \Pr(c) + 4} \leq 1$. The constraint $\frac{3}{2 \cdot \Pr(c) + 4} \geq 1 - \Pr(c)$, on the other hand, is only satisfied for $\Pr(c) \geq \frac{\sqrt{3}-1}{2} \approx 0.366$. For values of $\Pr(c)$ smaller than $\frac{\sqrt{3}-1}{2}$, the value for $\Pr(d)$ at which v_{max} would be found is smaller than allowed for by the constraint $\Pr(d) \geq 1 - \Pr(c)$. The maximum of v will then be found at the smallest value allowed for $\Pr(d) = 1 - \Pr(c)$. At $\Pr(d) = \frac{3}{2 \cdot \Pr(c) + 4}$ now is found that $v = \frac{(2 \cdot \Pr(c) + 1)^2}{4 \cdot \Pr(c) + 8}$ and at $\Pr(d) = 1 - \Pr(c)$ is found that $v = \Pr(c) - \Pr(c)^3$. Expressed as a function of $\Pr(c)$, the maximum prior convergence error v_{max} thus equals

$$v_{max} = \begin{cases} \Pr(c) - \Pr(c)^3 & \text{if } \Pr(c) < \frac{\sqrt{3}-1}{2} \\ \dfrac{\big(2 \cdot \Pr(c) + 1\big)^2}{4 \cdot \Pr(c) + 8} & \text{if } \Pr(c) \geq \frac{\sqrt{3}-1}{2} \end{cases}$$

At $\Pr(c) = \frac{\sqrt{3}-1}{2}$, both functions in the expression above yield the value $\frac{3-\sqrt{3}}{4}$. At $\Pr(c) = \frac{\sqrt{3}-1}{2}$, moreover, the first derivatives of both functions yield the value $\frac{3\cdot\sqrt{3}-4}{2}$. The function defined for v_{max} thus is continuously differentiable on the entire interval $[0,1]$ for $\Pr(c)$. With $\Pr(d) = \frac{3}{2\cdot\Pr(c)+4}$ the matching value of $\Pr(c \mid a^1a^2a^3)$ is $\Pr(c \mid a^1a^2a^3) = \frac{1}{3}\cdot\big(\Pr(c)-1\big)\cdot\big(2\cdot\Pr(c)+4\big)+1$ and with $\Pr(d) = 1-\Pr(c)$ the matching value is $\Pr(c \mid a^1a^2a^3) = 0$. Figure 9.12 shows, as a function of $\Pr(c)$, the combination of values of $\Pr(d)$ and $\Pr(c \mid a^1a^2a^3)$ that yield a maximum prior convergence.

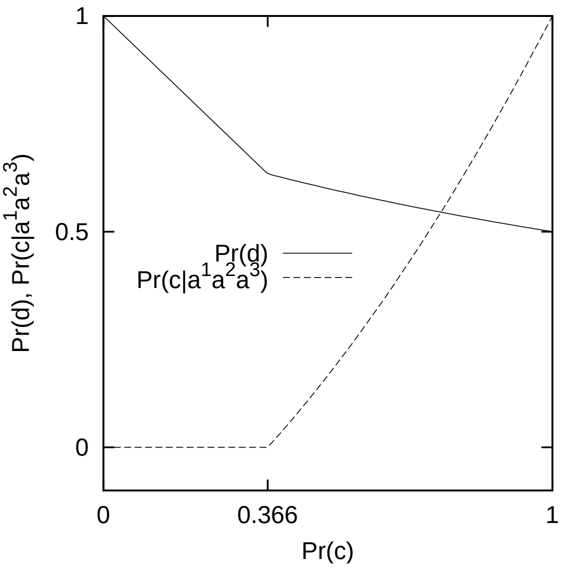


Figure 9.12: The combinations of values of $\Pr(d)$ and $\Pr(c \mid a^1a^2a^3)$, as function of $\Pr(c)$, for which the maximum prior convergence error is found.

Analogously, the relationship between the minimum convergence error v_{min} and $\Pr(c)$ is found to be

$$v_{min} = \begin{cases} \dfrac{\big(3-2\cdot\Pr(c)\big)^2}{4\cdot\Pr(c)-12} & \text{if } \Pr(c) \leq \frac{3-\sqrt{3}}{2} \\ -\big(1-\Pr(c)\big)+\big(1-\Pr(c)\big)^3 & \text{if } \Pr(c) > \frac{3-\sqrt{3}}{2} \end{cases}$$

Note that at $\Pr(c) = \frac{3-\sqrt{3}}{2}$, both functions in the expression above yield the value $v_{min} = \frac{\sqrt{3}-3}{4}$. At $\Pr(c) = \frac{3-\sqrt{3}}{2}$, moreover, the first derivatives of the two functions yield the value of $\frac{3\cdot\sqrt{3}-4}{2}$. The function defined for v_{min} thus also is continuously differentiable on the entire interval $[0,1]$ for $\Pr(c)$. Figure 9.13(a) shows the four functions which feature in the expressions for v_{max} and v_{min} given above; Figure 9.13(b) depicts the functions v_{max} and v_{min} which define the area of feasible combinations of $\Pr(c)$ and v. Note that, again, v_{max} and v_{min} are each others point reflections in the point $(\Pr(c), v) = (0, 0.5)$.

To conclude, from the functions v_{max} and v_{min} derived above, it is easily observed that the prior error for the convergence node of in a double loop adopts a value in the range $[-0.75, 0.75]$.

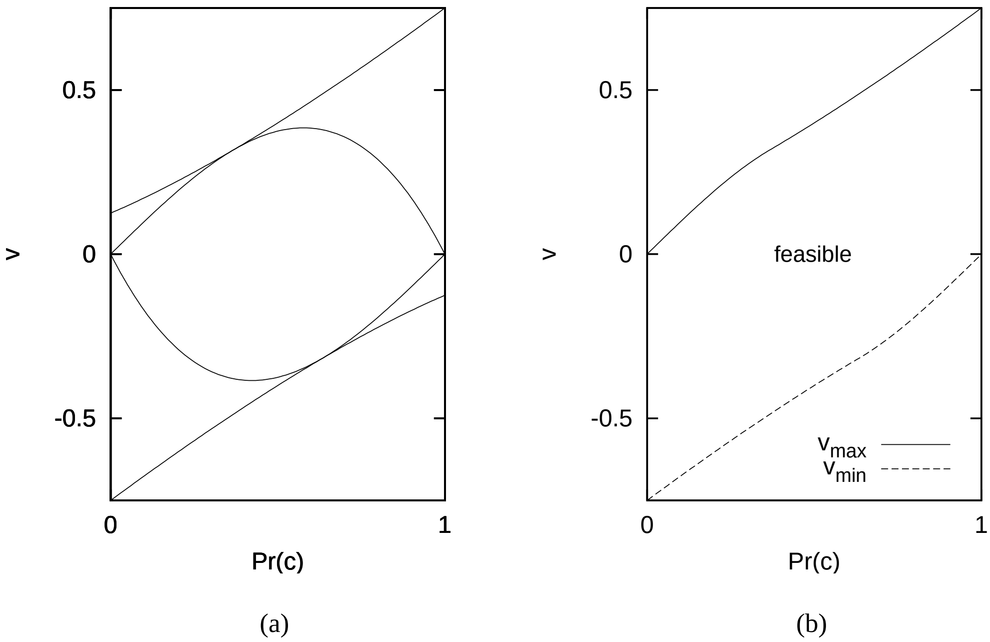


Figure 9.13: The four functions defining the feasible $\Pr(c)$,v-combinations (a), and the corresponding area of feasible $\Pr(c)$,v-combinations (b), for a convergence node with three loop parents.

The error can attain its general minimum -0.75 only for $\Pr(c) = 0$ and its general maximum 0.75 only for $\Pr(c) = 1$.

9.3.3 More Complex Binary Loops

In the previous section an expression was derived for the prior convergence error found in a convergence node with a complexity degree of three. Such a convergence node is included in a double loop. In more complex loops, convergence nodes with a higher complexity degree will be found. In this section the expression of the convergence error is generalised to convergence nodes in binary networks with a single loop of arbitrary complexity. Consider a convergence node C with the binary parents $A^1, \ldots, A^n$ and the common parent D of $A^1, \ldots, A^n$. Straightforwardly generalising the expression from the previous section gives the following expression for the convergence error v_i for the value c_i of C:

$$v_i \;=\; \sum_{\mathbf{m}} (s_{\mathbf{a^m}} - t_{\mathbf{a^m}}) \cdot w_{\bar{a}^1 \ldots \bar{a}^n \backslash \mathbf{a^m}}(\mathbf{a^m})$$

where

$$\begin{aligned}
&\mathbf{m} \in \mathcal{P}(\{1, \ldots, n\}) \\
&\mathbf{a^m} = a^x \ldots a^y \;\text{ for }\; \mathbf{m} = \{x, \ldots, y\} \\
&s_{\mathbf{a^m}} = \sum_D \prod_{i \in \mathbf{m}} \Pr(a^i \mid D) \cdot \Pr(D) \\
&t_{\mathbf{a^m}} = \prod_{i \in \mathbf{m}} \sum_D \Pr(a^i \mid D) \cdot \Pr(D) \\
&w_{\bar{a}^1 \ldots \bar{a}^n \backslash \mathbf{a^m}}(\mathbf{a^m}) = Y^{\star}_{\bar{a}^1 \ldots \bar{a}^n \backslash \mathbf{a^m}}(\mathbf{a^m}, c_i)
\end{aligned}$$

in which $\bar{a}^1 \ldots \bar{a}^n \backslash \mathbf{a^m}$ denotes the value assignment 'False' to the nodes included in the set $\{A^1, \ldots, A^n\} \backslash \{A^x, \ldots, A^y\}$. Note that the term $s_{\mathbf{a^m}}$, as before, equals $\Pr(a^x \ldots a^y)$ and thus accounts for the dependence between $A^x, \ldots, A^y$ through D. The term $t_{\mathbf{a^m}}$ equals $\Pr(a^x) \cdot \ldots \cdot \Pr(a^y)$ and again captures $\Pr(a^x \ldots a^y)$ under the assumption of independence of $A^x, \ldots, A^y$. Further note that the expression includes terms for all possible loops included in the compound loop. The term with $\mathbf{m} = 1, \ldots, n$, pertains to the entire compound loop. With $|\mathbf{m}| = n - 1$, the n compound loops with a single incoming arc of C deleted are considered, and so on. Note that, if the number of elements of $\mathbf{m}$ is smaller than two, no loop is left; the term $s_{\mathbf{a^m}}$ equals the term $t_{\mathbf{a^m}}$ and $(s_{\mathbf{a^m}} - t_{\mathbf{a^m}}) \cdot w_{\bar{a}^1 \ldots \bar{a}^n \backslash \mathbf{a^m}}(\mathbf{a^m})$ equals zero.

In the previous section on double loops it was argued that the expression derived for the prior convergence error of a double loop with a single convergence node also applies to the second convergence node of a double loop with two consecutive convergence nodes. This is because for the second convergence node the same expressions for the exact probability $\Pr(c_i)$ and the approximate probability $\widetilde{\Pr}(c_i)$ can be derived as for the convergence node in a double loop with a single convergence node. Analogously, for more complex loops, the same expressions for the exact and the approximate probabilities can be derived for convergence nodes with the same complexity, irrespective of the graphical structure. The expression for the convergence error given above, therefore, is applicable to all convergence nodes with a complexity degree n in a binary network with a single loop. For convergence nodes with a higher complexity than the number of incoming arcs on the loop, that is, for convergence node with a convergence node in the same loop as an ancestor, again, however, it may be necessary first to establish the required factors from the specification of the network.

In Sections 9.2.3 and 9.3.2 the relationships between the feasible maximum and minimum for the prior convergence error and the probability $\Pr(c)$ were determined for convergence nodes of with a complexity degree of two and three. The results in these sections now suggest a generalisation of the analysis of this relationship given a convergence node of arbitrary complexity. Recall that for convergence nodes of simple and double loops, it was found experimentally that the maximum convergence error v_{max} could be attained by taking the probabilities $\Pr(d)$ and $\Pr(c \mid a^1 \ldots a^n)$ as variables in the maximisation problem with $\Pr(a^i \mid d) = 1$ for all $i = 1, \ldots, n$, $\Pr(c \mid \bar{a}^1 \ldots \bar{a}^n) = 1$, and with the value zero for all other probabilities specified in the network. This specification readily is applicable to a convergence node with n incoming arcs on the loop. With these probabilities, as before, a reduced optimisation problem results. Analogous to before is found that $\Pr(c) = \Pr(c \mid a^1 \ldots a^n) \cdot \Pr(d) - \Pr(d) + 1$. Now below an expression for v for the reduced maximisation problem is derived. Writing x for $\Pr(d)$ and r for $\Pr(c \mid a^1 \dots a^n)$, for n is even, the term from the convergence error which applies to the entire compound loop includes

$$\begin{aligned} &w(a^1 \dots a^n) = r + 1 \\ &s_{a^1 \dots a^n} = x \\ &t_{a^1 \dots a^n} = x^n \end{aligned}$$

The terms which apply to the loops with a single incoming arc of the convergence node deleted, include

$$\begin{aligned} &w_{\bar{a}^i}(a^1 \dots a^n \backslash a^i) = -1 \\ &s_{a^1 \dots a^n \backslash a^i} = x \\ &t_{a^1 \dots a^n \backslash a^i} = x^{n-1} \end{aligned}$$

Note that this term pertains to $\binom{n}{1}$ different joint value assignments $a^1 \dots a^n \backslash a^i$. The terms which apply to the loops with two incoming arcs of the convergence node deleted, include

$$\begin{aligned} &w_{\bar{a}^i \bar{a}^j}(a^1 \dots a^n \backslash a^i, a^j) = 1 \\ &s_{a^1 \dots a^n \backslash a^i a^j} = x \\ &t_{a^1 \dots a^n \backslash a^i a^j} = x^{n-2} \end{aligned}$$

Note that this term pertains to $\binom{n}{2}$ different joint value assignments $a^1 \dots a^n \backslash a^i, a^j$. An so on. For n is even, therefore, for the reduced optimisation problem the convergence error can be written as

$$\begin{aligned} v &= (r+1) \cdot (x - x^n) - \binom{n}{1} \cdot (x - x^{n-1}) + \binom{n}{2} \cdot (x - x^{n-2}) - \ldots + \binom{n}{n-2} \cdot (x - x^2) \\ &= r \cdot (x - x^n) + \sum_{k=0}^{n-2} \binom{n}{k} (x - x^{n-k}) \cdot (-1)^k \\ &= r \cdot (x - x^n) + x \cdot \sum_{k=0}^{n-2} \binom{n}{k} \cdot (-1)^k - \sum_{k=0}^{n-2} \binom{n}{k} \cdot x^{n-k} \cdot (-1)^k \end{aligned}$$

Using the binomium of Newton $\sum_{k=0}^{n} \binom{n}{k} \cdot a^k \cdot b^{n-k} = (a+b)^n$ and n being even, it is found that

$$\begin{aligned} \sum_{k=0}^{n-2} \binom{n}{k} \cdot (-1)^k &= \sum_{k=0}^{n} \binom{n}{k} \cdot (-1)^k \cdot (1)^{n-k} - n \cdot (-1)^{n-1} - (-1)^n \\ &= (-1+1)^n - n \cdot (-1)^{n-1} - (-1)^n \\ &= n - 1 \end{aligned}$$

and that

$$\begin{aligned} \sum_{k=0}^{n-2} \binom{n}{k} \cdot x^{n-k} \cdot (-1)^k &= \sum_{k=0}^{n} \binom{n}{k} \cdot x^{n-k} \cdot (-1)^k - n \cdot x \cdot (-1)^{n-1} - (-1)^n \\ &= (x-1)^n + n \cdot x - 1 \end{aligned}$$

Substituting these two results in the expression for v above and again using n being even, gives

$$v \;=\; r \cdot (x - x^n) - (1-x)^n - x + 1$$

For an odd n, the same expression is found by a similar derivation.

The problem of finding the maximum prior convergence error v_{max} for a given value of $\Pr(c)$ thus is reduced to finding the maximum of

$$v = \Pr(c \mid a^1 \ldots a^n) \cdot (\Pr(d) - \Pr(d)^n) - (1 - \Pr(d))^n - \Pr(d) + 1$$

under the constraints

$$\begin{array}{l} \Pr(c) = \Pr(c \mid a^1 \ldots a^n) \cdot \Pr(d) - \Pr(d) + 1 \\ \Pr(d) \in [0,1] \\ \Pr(c \mid a^1 \ldots a^n) \in [0,1] \end{array}$$

These constraints can again be reformulated as

$$\begin{array}{l} \Pr(c) = \Pr(c \mid a^1 \ldots a^n) \cdot \Pr(d) - \Pr(d) + 1 \\ \Pr(d) \in [1 - \Pr(c), 1] \end{array}$$

Recall that for establishing v_{max}, for convergence nodes with a complexity degree of two and three, $\Pr(d)$ had to be set to the boundary $1 - \Pr(c)$ of its feasible interval when $\Pr(c)$ was smaller than a particular value in order to ensure that $\Pr(d) \geq 1 - \Pr(c)$ and thus that $\Pr(c \mid a^1 \ldots a^n) \geq 0$. This value of $\Pr(c)$ depended on the complexity degree of the convergence node. It is conjectured that for all n there exists such a value for $\Pr(c)$ and that this value increases with increasing n; this value will be denoted by $C_*(n)$. At $\Pr(d) = 1 - \Pr(c)$ is found that $\Pr(c \mid a^1 \ldots a^n) = 0$, resulting in

$$v_{max} = \Pr(c) - \Pr(c)^n \;\text{ for }\; \Pr(c) \leq C_*(n)$$

Likewise is found that

$$v_{min} = -(1 - \Pr(c)) + (1 - \Pr(c))^n \;\text{ for }\; \Pr(c) \geq \big(1 - C_*(n)\big)$$

Note, that for $\Pr(c) = 1$ is found that $\Pr(d) \geq \big(1 - \Pr(c)\big) \Leftrightarrow \Pr(d) \geq 0$, which implies that the restriction $\Pr(d) \geq \big(1 - \Pr(c)\big)$ is fulfilled for any $\Pr(d) \in [0,1]$, irrespective of the number of parents n, therefore $C_*(n) < 1$.

Furthermore, the results for two and three parents suggest that, given a fixed number of parents, the general maximum v_{gmax} is found at $\Pr(c) = 1$. Given that $\Pr(c) = 1$, is found that $\Pr(c \mid a^1 \ldots a^n) = 1$ and $\Pr(d) = 0.5$ (recall that $\Pr(d) = 0$ is left out of consideration since given that $\Pr(d) = 0$ effectively no loop is present and $v = 0$). Substituting $\Pr(c \mid a^1 \ldots a^n) = 1$ and $\Pr(d) = 0.5$ in the expression for $v = r \cdot (x - x^n) - (1-x)^n - x + 1$ yields for $\Pr(c) = 1$

$$v_{gmax} = 2 \cdot \left(\frac{1}{2} - \left(\frac{1}{2}\right)^n \right)$$

Likewise is found that, given a fixed number of parents, the general minimum v_{gmin}, which is found at $\Pr(c) = 0$, equals

$$v_{gmin} = -2 \cdot \left(\frac{1}{2} - \left(\frac{1}{2} \right)^n \right)$$

For $n \to \infty$ the extremes approach 1 and -1 respectively which implies that in general $v \in \langle -1, 1 \rangle$.

Note that the extremes of the prior convergence error for an arbitrary probability $\Pr(c_i)$ were conjectured to depend only on the degree of complexity of the convergence node and not on the cardinality of these nodes. Therefore, in establishing the general extremes of the convergence error for a convergence node with a complexity degree of n, the expressions for v_{gmax} and v_{gmin} as given above can be used.

9.3.4 Compound Loops in General

In the previous two sections, the expression for the convergence error was extended to simple loops with non-binary nodes and to compound loops with binary nodes. Now, supported by experimental results, it is posed as a conjecture, that these expressions combine into the following general expression for the prior convergence error for networks with a single loop of arbitrary complexity with nodes with an arbirary number of values. Given a convergence node C with the parents $A^1, \ldots, A^n$ and given the common parent D of $A^1, \ldots, A^n$, the convergence error equals

$$v_i \;=\; \sum_{\mathbf{m}} \left(\sum_{\mathbf{A^m}} \left((s_{\mathbf{A^m}} - t_{\mathbf{A^m}}) \cdot \sum_{A^1,\ldots,A^n \backslash \mathbf{A^m}} w_{A^1,\ldots,A^n \backslash \mathbf{A^m}}(\mathbf{A^m}) \right) \right) / 2^n$$

where

$$\begin{aligned}
&\mathbf{m} \in \mathcal{P}(\{1, \ldots, n\}) \\
&\mathbf{A^m} = A^x, \ldots, A^y, \ \mathbf{m} = \{x, \ldots, y\} \\
&s_{\mathbf{A^m}} = \sum_{D} \prod_{i \in \mathbf{m}} \Pr(A^i \mid D) \cdot \Pr(D) \\
&t_{\mathbf{A^m}} = \prod_{i \in \mathbf{m}} \sum_{D} \Pr(A^i \mid D) \cdot \Pr(D) \\
&w_{A^1 \ldots A^n \backslash \mathbf{A^m}}(\mathbf{A^m}) = Y^{\star}_{A^1 \ldots A^n \backslash \mathbf{A^m}}(\mathbf{A^m}, c_i)
\end{aligned}$$

Note that, as before, the term $s_{\mathbf{A^m}}$ equals $\Pr(A^x \ldots A^y)$ and thus captures the dependence between the nodes $A^x, \ldots A^y$ through D and that the term $t_{\mathbf{A^m}}$ equals $\Pr(A^x) \cdot \ldots \cdot \Pr(A^y)$ and thus captures $\Pr(A^x \ldots A^y)$ under the assumption of independence of $A^x, \ldots, A^y$. Again, if the number of elements of $\mathbf{m}$ is smaller than two, then the term $s_{\mathbf{A^m}}$ equals the term and $t_{\mathbf{A^m}}$ and thus $\sum_{\mathbf{A^m}} \left((s_{\mathbf{A^m}} - t_{\mathbf{A^m}}) \cdot \sum_{A^1,\ldots,A^n \backslash \mathbf{A^m}} w_{A^1,\ldots,A^n \backslash \mathbf{A^m}}(\mathbf{A^m}) \right)$ equals zero. Following the same line of reasoning as in the previous section, it is posed that the expression for the prior convergence error from this section is applicable to all convergence nodes with complexity n.

Note that the expression above does not provide a computational efficient way to establish the exact probility of $\Pr(c)$ from $\widetilde{\Pr}(c)$ and v, compared with the brute force computation of this probability.

$$\Pr(c) = \sum_{A^1,\ldots,A^n,D} \Pr(c \mid A^n) \cdot \Pr(A^1 \mid D) \cdot \ldots \cdot \Pr(A^1 \mid D) \cdot \Pr(D)$$

The general expression, however, does show that the basic structure of the prior convergence error is preserved for networks with a single loop of arbitrary complexity with arbitrary-valued nodes. The convergence error still is composed of factors $s - t$, that reflect the dependencies between the values of the parents of the convergence node, and factors w, that determine to what extent these dependencies can affect to computed probability.

9.4 The Posterior Convergence Error

The error that is introduced by the loopy-propagation algorithm in the probabilities computed for a convergence node of a Bayesian network in the prior state, may change in size as soon as an observation is entered for a node of which the convergence node is dependent. In a network with a single simple loop, the observation can affect the error through causal messages or through diagnostic messages to the convergence node. The former type of observation will be called a *causal observation* and the latter type will be termed a *diagnostic observation*; the observation of the convergence node itself is left out of consideration in this section since after its observation the exact probabilities of the convergence node are known.

A causal observation trivially changes the prior convergence error computed with one of the expressions from the previous sections by conditioning all probabilities involved on the entered observation. Note that causal observations, as a consequence, do not change the range of the convergence error. Consider, as an example, the network from Figure 9.14. Observations for the nodes E, F, G, H and I represent all possible types of causal observation for the convergence node C. In the network in the prior state, for example, the factor $w = \Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})$ of the convergence error for the value c of node C is computed from the conditional probabilities $\Pr(c \mid ABH)$ and the prior probabilities $\Pr(H)$. After the observation of, for example $H = h$, the factor w is computed from the conditional probabilities $\Pr(c \mid ABh)$. The observation of H thus may change the factor w. Referring to Figure 9.2, an observation whose effect enters the loop through a causal message to C thus may change the curvature of the surface. Likewise, observation of the nodes E or G may change the factor $z = \Pr(d) - \Pr(d)^2$ of the convergence error and thus may change the location of the exact probability $\Pr(c)$ on the line segment. The observation of node F may change the factor $x = \Pr(a \mid d) - \Pr(a \mid \bar{d})$, and thus may change the orientation of the line segment that expresses the exact probability $\Pr(c)$. The observation of I to conclude may change the factors $y = \Pr(b \mid d) - \Pr(b \mid \bar{d})$ and z and thus may change both the orientation of the line segment and the location of the exact probability on the segment.

A diagnostic observation, on the other hand, fundamentally changes the expression of the convergence error by including an additional factor. In the following section, the posterior con-

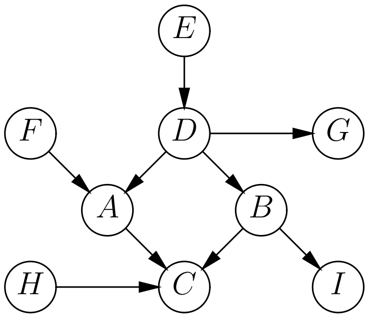


Figure 9.14: A multiply connected Bayesian network including a convergence node C with the dependent parents A and B.

vergence error given a diagnostic observation is studied for a binary convergence node; Section 9.4.2 briefly reviews the error for a convergence node with an arbitrary number of values.

9.4.1 Binary Nodes

Consider a network composed of just a binary node C and its child F, and suppose that the observation $F = f$ is entered. Pearl's algorithm computes the posterior probabilities for node C from the compound diagnostic parameter $\lambda(C) = \Pr(f \mid C)$ and the compound causal parameter $\pi(C)$, using the data-fusion rule

$$\Pr(C \mid f) = \alpha \cdot \lambda(C) \cdot \pi(C)$$

Normalisation then results in the posterior probability

$$\Pr(c \mid f) = \frac{\Pr(f \mid c) \cdot \pi(c)}{\Pr(f \mid c) \cdot \pi(c) + \Pr(f \mid \bar{c}) \cdot \big(1 - \pi(c)\big)}$$

for the value c of C.

Now consider the network from Figure 9.15; this network differs from the network from Figure 9.1, in that a child F is added for the convergence node C. Suppose that the observation $F = f$ is entered into the network. Pearl's algorithm now computes the compound causal parameter $\widetilde{\pi}(C)$ for the convergence node C using the causal messages $\widetilde{\pi}_C(A)$ and $\widetilde{\pi}_C(B)$ it receives from its parents. As argued in Chapter 8, after an observation has been entered for node F, these messages may include a cycling error. In determining the posterior convergence error, the cycling error is left out of consideration and it is assumed that node C receives the correct causal messages $\pi_C(A)$ and $\pi_C(B)$ from nodes A and B; the effect of the cycling error will be discussed in Section 9.5. From the exact causal messages $\pi_C(A)$ and $\pi_C(B)$ node C computes a compound causal parameter that includes just a convergence error. This causal parameter will be indicated with $\widetilde{\pi}_{conv}$. Note that $\widetilde{\pi}_{conv}(c)$ of the posterior network equals $\widetilde{\pi}(c)$ of the prior network and thus equals $\widetilde{\Pr}(c)$. The posterior probability of c given f that is established from $\widetilde{\pi}_{conv}(c)$ will be indicated by $\widetilde{\Pr}_{conv}(c \mid f)$.

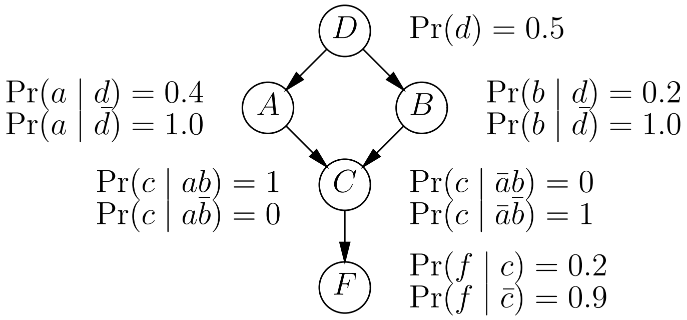

D
$\Pr(d) = 0.5$
$\Pr(a \mid d) = 0.4$
$\Pr(a \mid \bar{d}) = 1.0$
A
B
$\Pr(b \mid d) = 0.2$
$\Pr(b \mid \bar{d}) = 1.0$
$\Pr(c \mid ab) = 1$
$\Pr(c \mid a\bar{b}) = 0$
C
$\Pr(c \mid \bar{a}b) = 0$
$\Pr(c \mid \bar{a}\bar{b}) = 1$
F
$\Pr(f \mid c) = 0.2$
$\Pr(f \mid \bar{c}) = 0.9$

Figure 9.15: A multiply connected Bayesian network including a convergence node C with the dependent parents A and B and the child F.

Recall that in the graphical illustration of the prior convergence error for node C in Figure 9.2, the depicted line segment captures, for the networks from Figures 9.1 and 9.15 in their prior states alike, the exact prior probability $\Pr(c) = \pi(c)$ as a function of $\Pr(d)$, given the conditional probabilities for nodes A, B and C. The surface captures, again for the networks in their prior states, the approximate probability $\widetilde{\Pr}(c) = \tilde{\pi}(c)$ as a function of $\pi_C(a)$ and $\pi_C(b)$, given the conditional probabilities for node C. When the transformation that is defined by $\Pr(c \mid f) = \frac{\Pr(f|c)\cdot\pi(c)}{\Pr(f|c)\cdot\pi(c)+\Pr(f|\bar{c})\cdot\big(1-\pi(c)\big)}$ is applied to the surface and the line segment, Figure 9.16 results in which the curve segment captures $\Pr(c \mid f)$ and the surface captures $\widetilde{\Pr}_{conv}(c \mid f)$. The curve segment lies on the curve defined by

$$x = 0.75 \cdot y + 0.25 \quad \text{and} \quad z = \frac{2.77019 \cdot x - 3.14286}{x - 1}$$

The surface equals

$$z = \frac{0.4 \cdot x \cdot y - 0.2 \cdot x - 0.2 \cdot y + 0.2}{-1.4 \cdot x \cdot y + 0.7 \cdot x + 0.7 \cdot y + 0.2}$$

As for the prior state of the network, the approximate probability $\widetilde{\Pr}_{conv}(c \mid f)$ is found as the orthogonal projection of the exact probability $\Pr(c \mid f)$ on the surface. This because the cycling error is left out of consideration. Note that compared to Figure 9.2, the line segment has changed into a curve segment and also the surface has deformed. These changes contrast the effect of a causal observation.

By subtracting $\widetilde{\Pr}_{conv}(c \mid f)$ from $\Pr(c \mid f)$, an expression is derived for the posterior convergence error found in the probability for c given f:

$$\Pr(c \mid f) - \widetilde{\Pr}_{conv}(c \mid f) = \frac{\Pr(f \mid c) \cdot \pi(c)}{\Pr(f)} - \frac{\Pr(f \mid c) \cdot \tilde{\pi}_{conv}(c)}{\widetilde{\Pr}(f)}$$

where

$$\begin{aligned} \Pr(f) &= \Pr(f \mid c) \cdot \pi(c) + \Pr(f \mid \bar{c}) \cdot \pi(\bar{c}) \\ \widetilde{\Pr}(f) &= \Pr(f \mid c) \cdot \tilde{\pi}_{conv}(c) + \Pr(f \mid \bar{c}) \cdot \tilde{\pi}_{conv}(\bar{c}) \end{aligned}$$

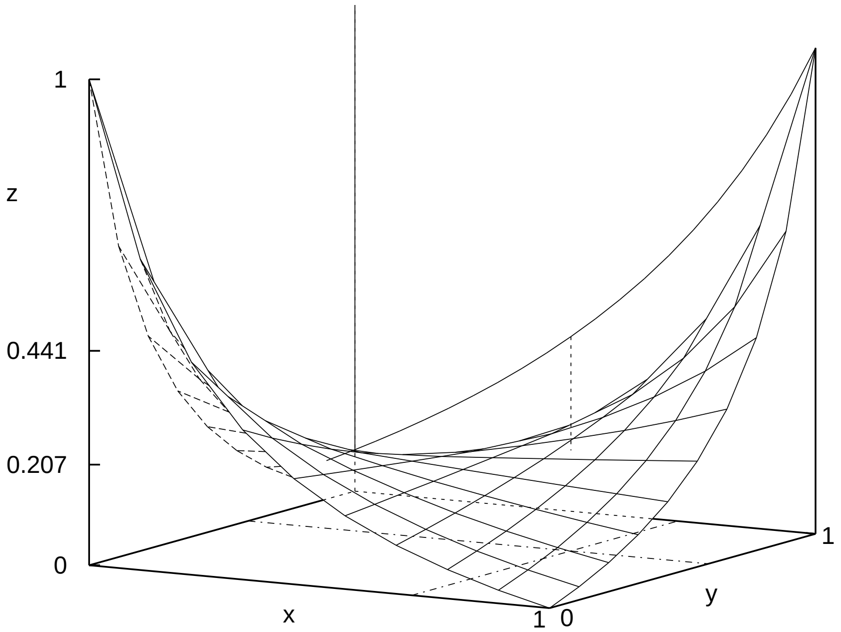


Figure 9.16: The line segment capturing $\Pr(c \mid f)$ and the surface capturing $\widetilde{\Pr}_{conv}(c \mid f)$ for the network from Figure 9.15.

Rearranging terms gives

$$u = \frac{\Pr(f \mid c) \cdot \Pr(f \mid \bar{c}) \cdot v}{\Pr(f) \cdot \Big(\Pr(f) - \big(\Pr(f \mid c) - \Pr(f \mid \bar{c}) \big) \cdot v \Big)} \tag{9.1}$$

where

$$v \;=\; \pi(c) - \widetilde{\pi}_{conv}(c)$$

denotes the prior convergence error in the probability of c computed for node C.

The expression derived above for the posterior convergence error is illustrated by the following example.

Example 9.6 *The prior convergence error v for the value c of node C in the network from Figure 9.15 equals $l \cdot m \cdot n \cdot w = -0.6 \cdot -0.8 \cdot 0.25 \cdot 2 = 0.24$; the prior probability $\Pr(f)$ is established from $\Pr(f \mid c) = 0.2$ and $\Pr(f \mid \bar{c}) = 0.9$, and equals $\Pr(f) = 0.354$. Now suppose that the observation $F = f$ is entered. The posterior convergence error $\Pr(c \mid f) - \widetilde{\Pr}_{conv}(c \mid f)$ is computed to be $\frac{0.2 \cdot 0.9 \cdot 0.24}{0.354 \cdot \big(0.354 - (0.2 - 0.9) \cdot 0.24\big)} \approx 0.233$.*

The posterior convergence error is a function of $\Pr(f \mid c)$, $\Pr(f \mid \bar{c})$, $\pi(c)$ and v. Since $\Pr(f \mid c)$, $\Pr(f \mid \bar{c})$ and $\pi(c)$ are probabilities, they are included in the interval $[0, 1]$; since the convergence node has two parents in the loop, moreover, the prior convergence error v essentially lies in $[-0.5, 0.5]$. Recall, however, from Section 9.2 that, in a network in its prior state, the possible $\Pr(c), v$-combinations are restricted. Since, given a diagnostic observation, $\pi(c)$ equals $\Pr(c)$, the same restrictions apply to the possible $\pi(c), v$-combinations. Given two parents of the convergence node C, therefore

$$v_{max} = \begin{cases} \pi(c) - \pi(c)^2 & \text{if } \pi(c) \leq 1/3 \\ \frac{1}{8} \cdot (\pi(c) + 1)^2 & \text{if } \pi(c) \geq 1/3 \end{cases}$$

and

$$v_{min} = \begin{cases} -\frac{1}{8}(\pi(c) - 2)^2 & \text{if } \pi(c) \leq 2/3 \\ -\pi(c) + \pi(c)^2 & \text{if } \pi(c) > 2/3 \end{cases}$$

Since the causal parameter $\pi(c)$ is used in the computation of $\Pr(f)$, the restrictions on the $\pi(c), v$-combinations also restrict the possible $\Pr(f), v$-combinations.

Example 9.7 *Consider a convergence node C with a child F and suppose that the observation $F = f$ has been entered. The feasible $\Pr(f)$,v-combinations now are derived by establishing the maximum v_{max} and the minimum v_{min} of the prior convergence error using $\Pr(f) = \Pr(f \mid c) \cdot \pi(c) + \Pr(f \mid \bar{c}) \cdot (1 - \pi(c))$. With, for example, $\Pr(f \mid c) = 1$ and $\Pr(f \mid \bar{c}) = 0.1$ the following functions are found*

$$v_{max} = \begin{cases} \left(\frac{\Pr(f)-0.1}{0.9}\right) - \left(\frac{\Pr(f)-0.1}{0.9}\right)^2 & \textit{if } \Pr(f) < 0.4 \\ \frac{1}{8} \cdot \left(\frac{(\Pr(f)+0.8)}{0.9}\right)^2 & \textit{if } \Pr(f) \geq 0.4 \end{cases}$$

$$v_{min} = \begin{cases} -\frac{1}{8}\left(\frac{\Pr(f)-1.9}{0.9}\right)^2 & \textit{if } \Pr(f) \leq 0.7 \\ -\left(\frac{\Pr(f)-0.1}{0.9}\right) + \left(\frac{\Pr(f)-0.1}{0.9}\right)^2 & \textit{if } \Pr(f) > 0.7 \end{cases}$$

and with, for example, $\Pr(f \mid c) = 0.1$ and $\Pr(f \mid \bar{c}) = 1$ is found that

$$v_{max} = \begin{cases} \frac{1}{8} \cdot \left(\frac{(1.9-\Pr(f))}{0.9}\right)^2 & \textit{if } \Pr(f) \leq 0.7 \\ \left(\frac{1-\Pr(f)}{0.9}\right) - \left(\frac{1-\Pr(f)}{0.9}\right)^2 & \textit{if } \Pr(f) > 0.7 \end{cases}$$

$$v_{min} = \begin{cases} -\left(\frac{1-\Pr(f)}{0.9}\right) + \left(1 - \frac{\Pr(f)}{0.9}\right)^2 & \textit{if } \Pr(f) < 0.4 \\ -\frac{1}{8}\left(\frac{-\Pr(f)-0.8}{0.9}\right)^2 & \textit{if } \Pr(f) \geq 0.4 \end{cases}$$

In the Figures 9.19(a) and 9.19(b) the areas with the feasible $\Pr(f)$,v-combinations, given $\Pr(f \mid c) = 1, \Pr(f \mid \bar{c}) = 0.1$ and $\Pr(f \mid c) = 0.1, \Pr(f \mid \bar{c}) = 1$, respectively, are indicated.

Figures 9.17 and 9.18 now depict the posterior convergence error u for the value c of node C as a function of $\Pr(f)$ and v; Figure 9.17 shows the error for $\Pr(f \mid c) = 1$ and $\Pr(f \mid \bar{c}) = 0.1$ and Figure 9.18 shows the error for $\Pr(f \mid c) = 0.1$ and $\Pr(f \mid \bar{c}) = 1$. Note that the Figures 9.17 and 9.18 only show u for feasible $\Pr(f)$,v-combinations.

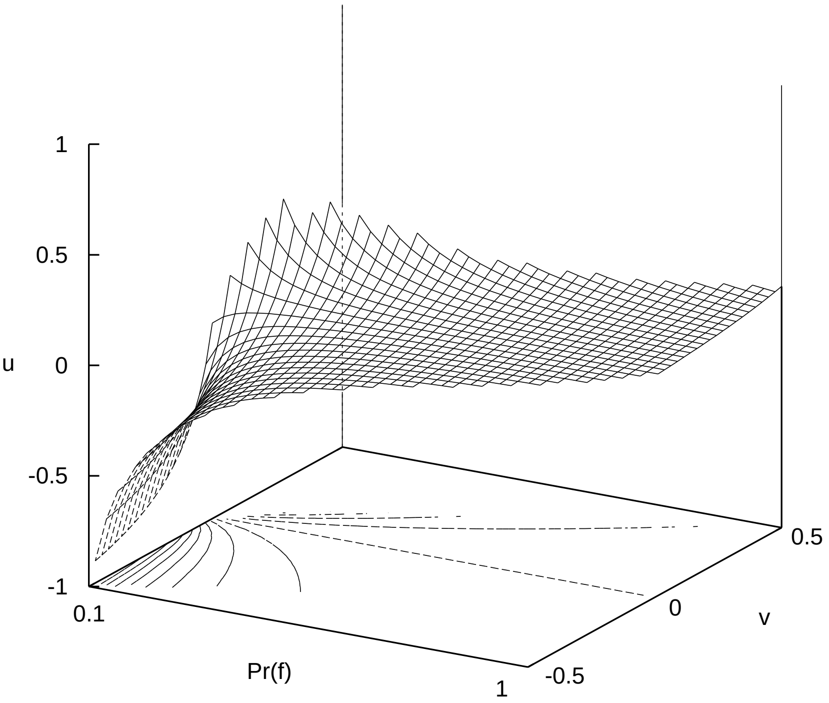


Figure 9.17: The posterior convergence error $u = \Pr(c \mid f) - \widetilde{\Pr}_{conv}(c \mid f)$ for a convergence node C with an observed child F, as a function of the prior convergence error $v = \Pr(c) - \widetilde{\Pr}(c)$ and the prior probability $\Pr(f)$ given that $\Pr(f \mid c) = 1$ and $\Pr(f \mid \bar{c}) = 0.1$.

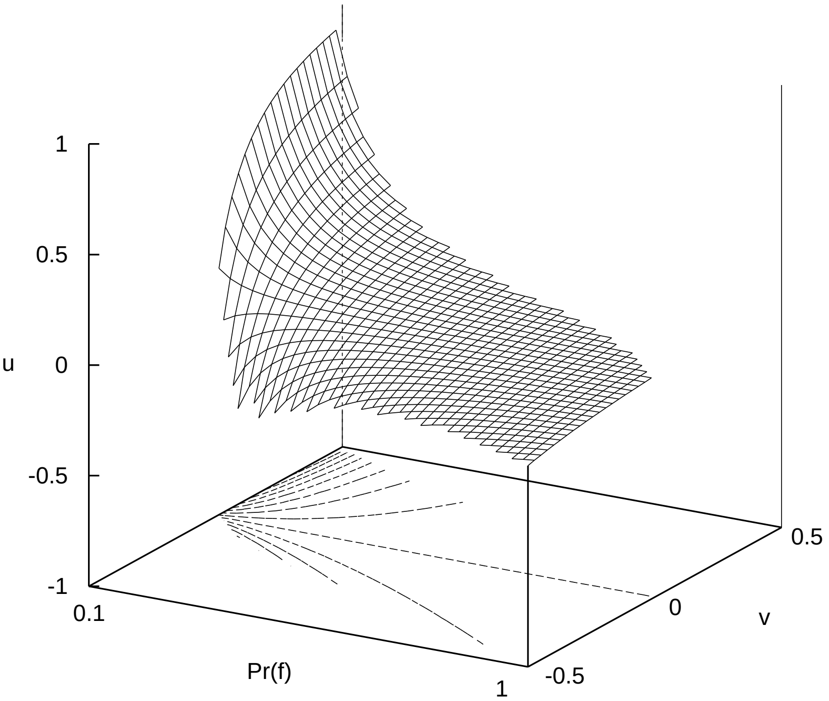


Figure 9.18: The posterior convergence error $u = \Pr(c \mid f) - \widetilde{\Pr}_{conv}(c \mid f)$ for a convergence node C with an observed child F, as a function of the prior convergence error $v = \Pr(c) - \widetilde{\Pr}(c)$ and the prior probability $\Pr(f)$ given that $\Pr(f \mid c) = 0.1$ and $\Pr(f \mid \bar{c}) = 1$.

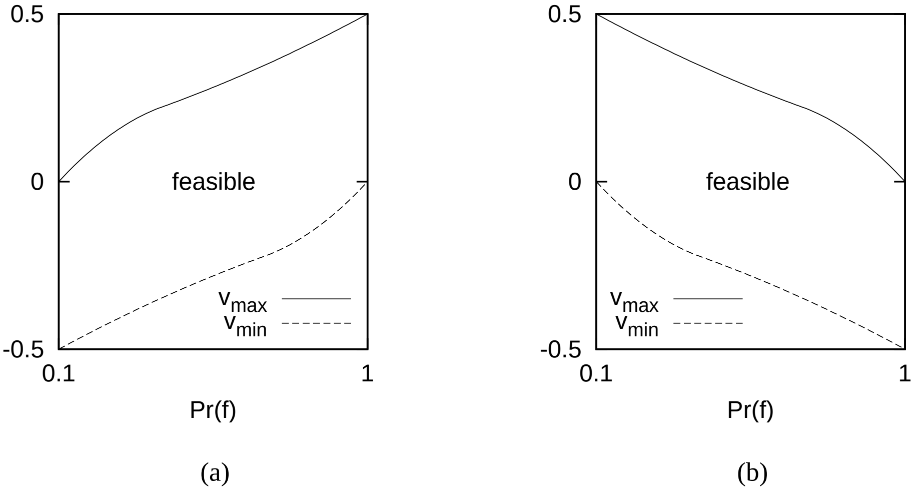


Figure 9.19: The feasible $\Pr(f)$,v-combinations given that $\Pr(f \mid c) = 1$ and $\Pr(f \mid \bar{c}) = 0.1$ (a), and given that $\Pr(f \mid c) = 0.1$ and $\Pr(f \mid \bar{c}) = 1$ (b).

In the remainder of this section some of the characteristics of the function u are discussed. First of all, the sign of the posterior convergence error u equals the sign of the prior convergence error v because the sign of the numerator of Expression 9.1 equals the sign of v and the sign of the denominator of the expression is always positive. The latter because $|\Pr(f \mid c) - \Pr(f \mid \bar{c})| \leq \Pr(f)$ and $v \in [-0.5, 0.5]$. From the numerator of the first derivative of u with respect to v

$$\frac{\partial u}{\partial v} = \frac{\Pr(f \mid c) \cdot \Pr(f \mid \bar{c})}{\Big((\pi(c) - v) \cdot \big(\Pr(f \mid c) - \Pr(f \mid \bar{c})\big) + \Pr(f \mid \bar{c})\Big)^2}$$

being positive, furthermore, it follows that, all other factors being constant, u increases with increasing v. Furthermore, obviously, given that $\Pr(f) \neq 0$, the posterior convergence error equals zero for $\Pr(f \mid c) = 0$, $\Pr(f \mid \bar{c}) = 0$ or $v = 0$. For $\Pr(f \mid c) = 0$ the computed probability $\widetilde{\Pr}(c \mid f) = \frac{\Pr(f|c) \cdot \widetilde{\pi}_{conv}(c)}{\Pr(f|c) \cdot \widetilde{\pi}_{conv}(c) + \Pr(f|\bar{c}) \cdot \widetilde{\pi}_{conv}(\bar{c})}$, is independent of the causal parameter $\pi(c)_{conv}$, and therefore, an error in $\pi_{conv}(c)$, cannot affect the computed probability $\widetilde{\Pr}(c \mid f)$. A similar observation holds for $\Pr(f \mid \bar{c}) = 0$. For $v = 0$, no error is present in $\widetilde{\pi}_{conv}(c)$, and thus obviously $\widetilde{\Pr}_{conv}(c \mid f)$ indeed equals $\Pr(c \mid f)$. Furthermore, for $\Pr(f \mid c) = \Pr(f \mid \bar{c})$, u equals v. This is not surprising because given that $\Pr(f \mid c) = \Pr(f \mid \bar{c})$, C and F are independent.

Moreover, the expression of the posterior convergence error is ill-defined for the combination $\pi(c) = 0$ and $\Pr(f \mid \bar{c}) = 0$ and for the combination $\pi(c) = 1$ and $\Pr(f \mid c) = 0$. In both cases is found that $\Pr(f) = 0$. Below the limit of u for $\pi(c) = 0$ and $\Pr(f \mid \bar{c}) \downarrow 0$ is established, assuming that $v \neq 0$. For $\pi(c) = 0$, is found that

$$u = \frac{\Pr(f \mid c) \cdot \Pr(f \mid \bar{c}) \cdot v}{\Pr(f \mid \bar{c})^2 - \Pr(f \mid \bar{c}) \cdot \big(\Pr(f \mid c) - \Pr(f \mid \bar{c})\big) \cdot v}$$

Division by $\Pr(f \mid \bar{c})$ gives

$$u = \frac{\Pr(f \mid c) \cdot v}{\Pr(f \mid \bar{c}) - \big(\Pr(f \mid c) - \Pr(f \mid \bar{c})\big) \cdot v}$$

and taking $\Pr(f \mid \bar{c}) = 0$ gives

$$\lim_{\Pr(f|\bar{c})\downarrow 0} = -1$$

The limit for $\pi(c) = 1$ and $\Pr(f \mid c) \downarrow 0$ and $v \neq 0$ can be established analogously to be equal to 1. For the range of the posterior convergence error thus is found that $u \in \langle -1, 1 \rangle$. Note that for the combinations $\pi(c) = 0$, $\Pr(f \mid \bar{c}) \downarrow 0$ and $\pi(c) = 1$, $\Pr(f \mid c) \downarrow 0$, extremely small probabilities $\Pr(f)$ are found. The posterior convergence error thus reaches its extreme values for extreme unlikely observations $F = f$.

9.4.2 Multiple-valued Nodes

In this section, the posterior convergence error given nodes with an arbitrary number of values is discussed. In the derivation of the posterior convergence error in the previous section, the cardinality of node F was not relevant; the derived expression thus is directly applicable to all values of an arbitrary-valued node F. The derived expression is not applicable to a non-binary node C however. Below the expression for the posterior convergence error given an convergence node C with an arbitrary number of values is derived. Given such a convergence node C, the posterior convergence error in the probability computed for the value c_k of C given the observation $F = f$ equals

$$\begin{aligned} u_k &= \Pr(c_k \mid f) - \widetilde{\Pr}_{conv}(c_k \mid f) \\ &= \frac{\Pr(f \mid c_k) \cdot \pi(c_k)}{\Pr(f)} - \frac{\Pr(f \mid c_k) \cdot (\pi(c_k) - v_k)}{\widetilde{\Pr}(f)} \\ &= \frac{\Pr(f \mid c_k) \cdot \pi(c_k) \cdot \widetilde{\Pr}(f) - \Pr(f \mid c_k) \cdot (\pi(c_k) - v_k) \cdot \Pr(f)}{\Pr(f) \cdot \widetilde{\Pr}(f)} \end{aligned}$$

where

$$\Pr(f) = \sum_C \Pr(f \mid C) \cdot \pi(C)$$
$$\widetilde{\Pr}(f) = \sum_C \Pr(f \mid C) \cdot (\pi(C) - v_C)$$

in which v_C denotes the prior convergence error for the relevant value of node C. The numerator of the expression above equals

$$\begin{aligned}
& \Pr(f \mid c_k) \cdot \pi(c_k) \cdot \sum_C \Pr(f \mid C) \cdot (\pi(C) - v_C) \\
& - \Pr(f \mid c_k) \cdot (\pi(c_k) - v_k) \cdot \sum_C \Pr(f \mid C) \cdot \pi(C) \\
= \; & \Pr(f \mid c_k) \cdot \pi(c_k) \cdot \Big(\Pr(f \mid c_k) \cdot (\pi(c_k) - v_k) + \sum_{C \backslash c_k} \Pr(f \mid C) \cdot (\pi(C) - v_C) \Big) \\
& - \Pr(f \mid c_k) \cdot (\pi(c_k) - v_k) \cdot \Big(\Pr(f \mid c_k) \cdot \pi(c_k) + \sum_{C \backslash c_k} \Pr(f \mid C) \cdot \pi(C) \Big) \\
= \; & \Pr(f \mid c_k) \cdot \pi(c_k) \cdot \sum_{C \backslash c_k} \Pr(f \mid C) \cdot (\pi(C) - v_C) \\
& - \Pr(f \mid c_k) \cdot (\pi(c_k) - v_k) \cdot \sum_{C \backslash c_k} \Pr(f \mid C) \cdot \pi(C) \\
= \; & \Pr(f \mid c_k) \cdot \\
& \sum_{C \backslash c_k} \Big(\pi(c_k) \cdot \Pr(f \mid C) \cdot (\pi(C) - v_C) - (\pi(c_k) - v_k) \cdot \Pr(f \mid C) \cdot \pi(C) \Big) \\
= \; & \Pr(f \mid c_k) \cdot \sum_{C \backslash c_k} \Pr(f \mid C) \cdot (\pi(C) \cdot v_k - \pi(c_k) \cdot v_C)
\end{aligned}$$

and the denominator equals

$$\begin{aligned}
& \Pr(f) \cdot \sum_C \Pr(f \mid C) \cdot (\pi(C) - v_C) = \\
& \Pr(f) \cdot (\Pr(f) - \sum_C \Pr(f \mid C) \cdot v_C)
\end{aligned}$$

which results in the following expression for the posterior convergence error

$$u_k = \frac{\Pr(f \mid c_k) \cdot \sum_{C \backslash c_k} \Pr(f \mid C) \cdot (\pi(C) \cdot v_k - \pi(c_k) \cdot v_C)}{\Pr(f) \cdot (\Pr(f) - \sum_C \Pr(f \mid C) \cdot v_C)}$$

Analogous to the binary case is found that u_k is undefined for $\Pr(f \mid c_k) = 0$ and $\pi(c_k) = 1$ and for $\sum_{C \backslash c_k} \Pr(f \mid C) = 0$ and $\sum_{C \backslash c_k} \pi(C) = 1$. Note that thus again the posterior convergence error is undefined for $\Pr(f) = 0$. Below the limit u_k for $\pi(c_k) = 1$ and $\Pr(f \mid c_k) \downarrow 0$ is established, assuming that $\sum_{C \backslash c_k} v_C \neq 0$. Using that $\forall_{C \backslash c_k} \pi(C) = 0$ is found that

$$u_k = \frac{\Pr(f \mid c_k) \cdot \sum_{C \backslash c_k} \Pr(f \mid C) \cdot (-v_C)}{\Pr(f \mid c_k) \cdot (\Pr(f \mid c_k) - \sum_C \Pr(f \mid C) \cdot v_C)}$$

division by $\Pr(f \mid c_k)$, and then taking $\Pr(f \mid c_k) = 0$ gives:

$$\lim_{\Pr(f|c_k)\downarrow 0} u_k = \frac{\sum_{C\backslash c_k} \Pr(f \mid C) \cdot (-v_C)}{-\sum_{C\backslash c_k} \Pr(f \mid C) \cdot v_C} = 1$$

For the limit of u_k for $\sum_{C\backslash c_k} \pi(C) = 1$ and $\sum_{C\backslash c_k} \Pr(f \mid C) \downarrow 0$ assuming that $v_k \neq 0$ is found that,

$$u_k = \frac{\Pr(f \mid c_k) \cdot \sum_{C\backslash c_k} \Pr(f \mid C) \cdot \pi(C) \cdot v_k}{\sum_{C\backslash c_k} \Pr(f \mid C) \cdot \pi(C) \cdot (\sum_{C\backslash c_k} \Pr(f \mid C) \cdot \pi(C) - \sum_C \Pr(f \mid C) \cdot v_C)}$$

division by $\sum_{C\backslash c_k} \Pr(f \mid C) \cdot \pi(C)$ gives

$$\begin{aligned} u_k &= \frac{\Pr(f \mid c_k) \cdot v_k}{\sum_{C\backslash c_k} \Pr(f \mid C) \cdot \pi(C) - \sum_C \Pr(f \mid C) \cdot v_C} \\ &= \frac{\Pr(f \mid c_k) \cdot v_k}{\sum_{C\backslash c_k} \Pr(f \mid C) \cdot \pi(C) - \sum_{C\backslash c_k} \Pr(f \mid C) \cdot v_C - \Pr(f \mid c_k) \cdot v_k} \end{aligned}$$

and finally taking $\sum_{C\backslash c_k} \Pr(f \mid C) = 0$ gives:

$$\lim_{\sum_{C\backslash c_k} \Pr(f|c_k)\downarrow 0} u_k = \frac{\Pr(f \mid c_k) \cdot v_k}{-\Pr(f \mid c_k) \cdot v_k} = -1$$

Thus, as in the binary case, $u_k \in \langle -1, 1 \rangle$.

9.5 The Cycling Error Entering the Convergence Node

As argued in Section 8, the approximation of a probability for a convergence node given one of its descendents may not just include a convergence error. Given that for all convergence nodes in the loop, the convergence node itself or one of its descendants is observed, information may cycle in the loop and the causal messages of the parents of the convergence nodes may include a cycling error. As a result, a cyling error is included in the compound causal parameters computed for the convergence nodes which in turn results in a cycling error in the approximations for the convergence nodes. The additional error is illustrated in Figures 9.21 and 9.22.

Figure 9.21 shows graphically the prior error $\pi(c) - \widetilde{\pi}(c)$ found in the network from in Figure 9.20. Again, the line segment captures the exact compound parameters $z = \pi(c)$ as a function of $\Pr(d)$ given the conditional probabilities for A, B and C from the network. For this line segment it is the case that

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 0.80 \\ 0.28 \end{bmatrix} \cdot \Pr(d) + \begin{bmatrix} 0 \\ 0.10 \\ 0.62 \end{bmatrix}$$

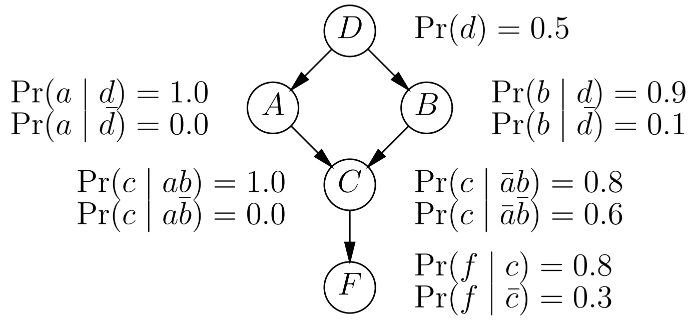

D
$\Pr(d) = 0.5$
$\Pr(a \mid d) = 1.0$
$\Pr(a \mid \bar{d}) = 0.0$
A
B
$\Pr(b \mid d) = 0.9$
$\Pr(b \mid \bar{d}) = 0.1$
$\Pr(c \mid ab) = 1.0$
$\Pr(c \mid a\bar{b}) = 0.0$
C
$\Pr(c \mid \bar{a}b) = 0.8$
$\Pr(c \mid \bar{a}\bar{b}) = 0.6$
F
$\Pr(f \mid c) = 0.8$
$\Pr(f \mid \bar{c}) = 0.3$

Figure 9.20: A multiply connected Bayesian network with a convergence node C having the dependent parents A and B and the child F.

In the example network $\Pr(d) = 0.5$ and the matching $z = \pi(c) = 0.76$. The surface captures the approximate compound parameters $z = \widetilde{\pi}(c)$ as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ given the conditional probabilities for node C. The equation of the surface is

$$z = 0.8 \cdot x \cdot y - 0.6 \cdot x + 0.2 \cdot y + 0.6$$

In the example network, $\pi_C(a) = 0.5$, $\pi_C(b) = 0.5$ and the matching $\widetilde{\pi}(c) = 0.6$. The prior convergence error $\pi(c) - \widetilde{\pi}(c) = 0.76 - 0.6$ equals 0.16.

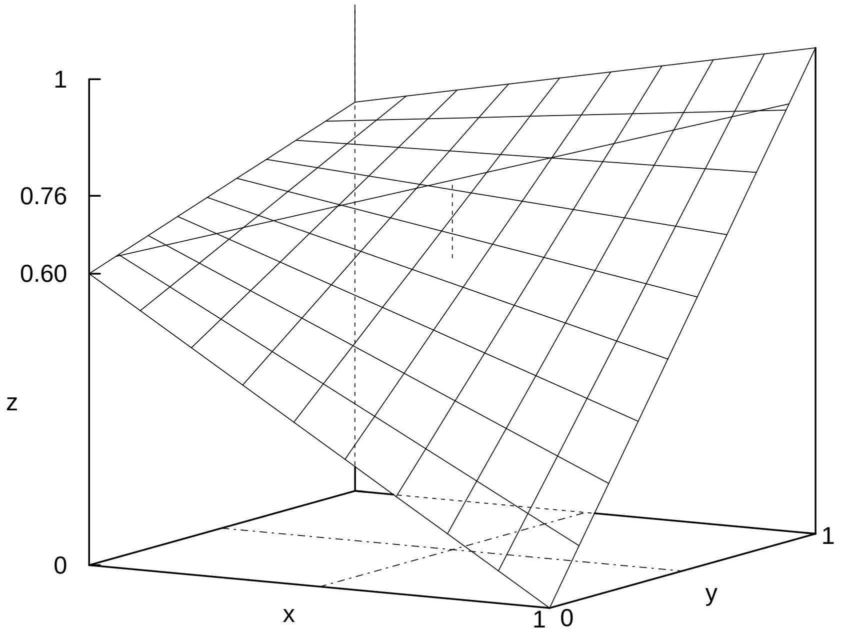


Figure 9.21: The line segment capturing $\Pr(c)$ and the surface, capturing $\widetilde{\Pr}(c)$ for the network from Figure 9.20.

Figure 9.22 shows, for the same network, the situation after the observation $F = f$ has been entered. The curve segment now captures $\Pr(c \mid f)$ as a function of $\Pr(d)$, again given the conditional probabilities for A, B and C from the example network. For this curve segment applies that

$$x = 1.25 \cdot y - 1.125 \quad \text{and} \quad z = \frac{0.224 \cdot x + 0.496}{0.14 \cdot x + 0.61}$$

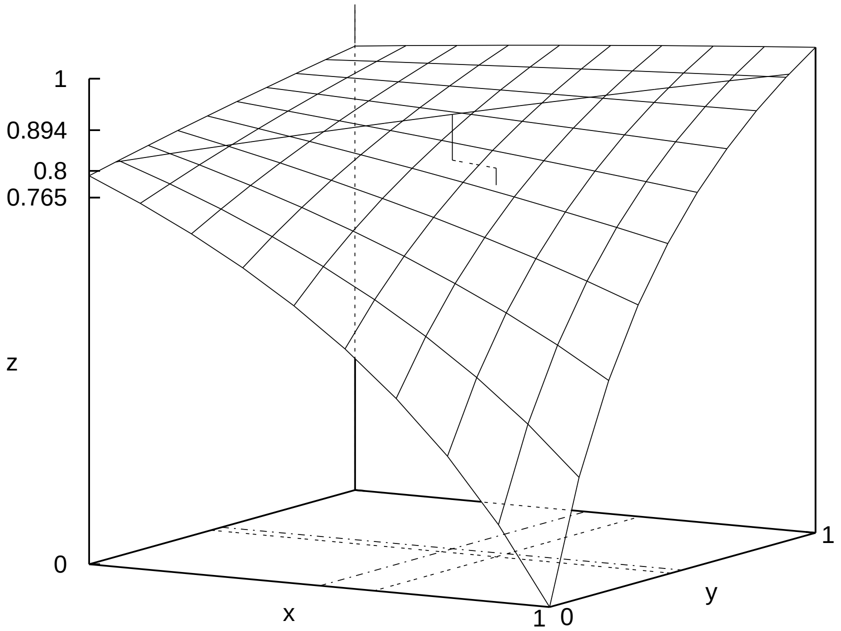


Figure 9.22: The line segment capturing $\Pr(c \mid f)$ and the surface, capturing $\widetilde{\Pr}_{conv}(c \mid f)$ and $\widetilde{\Pr}(c \mid f)$ for the network from Figure 9.20.

When $\Pr(d) = 0.5$, as in the example network, the matching $z = \Pr(c \mid f) = 0.894$. The surface captures the approximations for c given f as a function of $x = \pi_C(a)$ and $y = \pi_C(b)$ given the conditional probabilities of C from the example network. The equation of the surface is

$$z = \frac{0.64 \cdot x \cdot y - 0.48 \cdot x + 0.16 \cdot y + 0.48}{0.4 \cdot x \cdot y - 0.3 \cdot x + 0.1 \cdot y + 0.6}$$

In the example network, the exact causal messages equal $\pi_C(a) = 0.5$ and $\pi_C(b) = 0.5$, result in a compound causal parameter with just a convergence error of $\widetilde{\pi}_{conv}(c) = 0.6$ and in an approximation with just a convergence error of $\widetilde{\Pr}_{conv}(c \mid f) = 0.8$. The causal messages, however include a cycling error and equal $\widetilde{\pi}_C(a) \approx 0.6180$ and $\widetilde{\pi}_C(b) \approx 0.4607$ resulting in the actual compound causal parameter $\pi(c) = 0.5491$ and in the actual approximation $\widetilde{\Pr}(c \mid f) \approx 0.765$. In the network from Figure 9.20 the total error found in the probability computed for c given f thus equals approximately $0.894 - 0.765 = 0.129$.

Experiments showed that the point $(\widetilde{\pi}_C(a), \widetilde{\pi}_C(b))$ is located on the line that is defined by the point with the exact causal messages $(\pi_C(a), \pi_C(b))$ and the orthogonal projection of the saddle point of the surface on the base $\left(-\frac{Y^\star_{\bar{a}}(b,c)}{Y^\star(ab,c)}, -\frac{Y^\star_{\bar{b}}(a,c)}{Y^\star(ab,c)}\right)$. The equation of this line is

$$y = \frac{Y^\star_{\bar{b}}(a,c) + Y^\star(ab,c) \cdot \pi_C(b)}{Y^\star_{\bar{a}}(b,c) + Y^\star(ab,c) \cdot \pi_C(a)} \cdot x + \frac{Y^\star_{\bar{a}}(b,c) \cdot \pi_C(b) - Y^\star_{\bar{b}}(a,c) \cdot \pi_C(a)}{Y^\star_{\bar{a}}(b,c) + Y^\star(ab,c) \cdot \pi_C(a)}$$

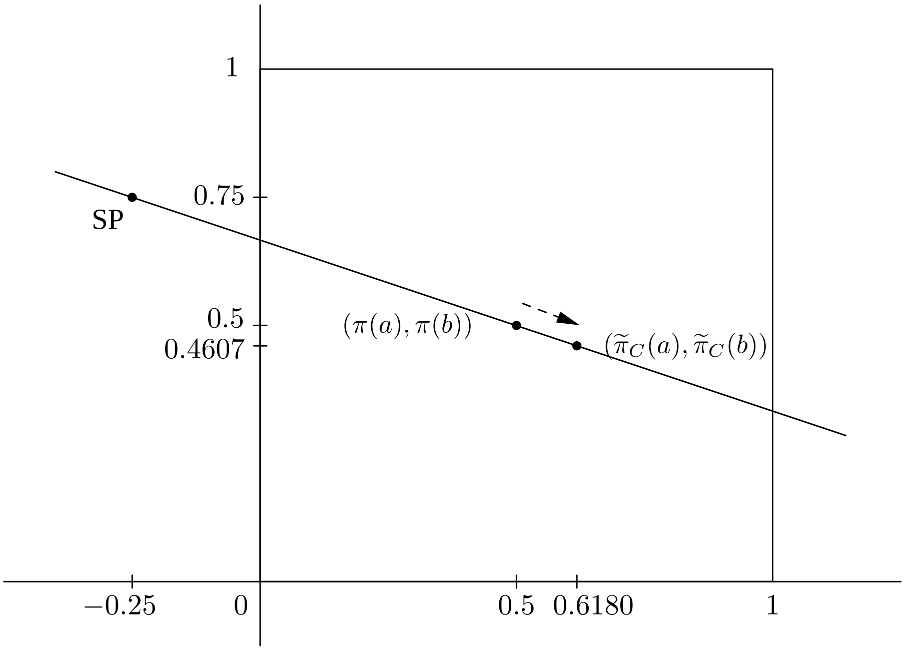


Figure 9.23: The line on which the approximate causal messages $(\widetilde{\pi}_C(a), \widetilde{\pi}_C(b))$ from nodes A and B to node C of the example network from Figure 9.20 are located given the observation $F = f$. The line is defined by the saddle point SP of the surface with the approximate probabilities of c given f and the exact causal messages $(\pi_C(a), \pi_C(b))$.

Example 9.8 *For the example network from Figure 9.20, the equation of the line on which the approximate causal messages are located equals*

$$y = -\frac{1}{3} \cdot x + \frac{2}{3}$$

The line is depicted in Figure 9.23. The approximate causal messages found upon loopy propagation $(\widetilde{\pi}_C(a), \widetilde{\pi}_C(b)) \approx (0.6180, 0.4607)$ *indeed are found on the line defined by the saddle point* $(-0.25.0.75)$ *of the surface and the exact causal messages* $(\pi_C(a), \pi_C(b)) = (0.5, 0.5)$.

Chapter 10

Inner Nodes

The previous chapter discussed the errors which are found in the probabilities computed for the convergence nodes of the loops of a Bayesian network upon loopy propagation. This chapter focusses on the inner nodes of the loops and elaborates, more specifically, on the effect of the cycling error on the decisiveness of the computed approximations. In Section 10.1, an expression is derived that relates for a binary network with just a simple loop, the exact probabilities for the inner loop nodes to the computed approximate probabilities. The derivation is analogous to the one constructed by Weiss [71] for an equivalent algorithm applied to a pairwise Markov networks with a single loop; the analysis of Weiss will be reviewed in more detail in Section 11.2. In Section 10.2, from the derived relationship between the exact and approximate probabilities the relationship between the decisiveness of the approximations on the one hand and the concepts of qualitative influence and intercausal influence from qualitative probabilistic networks on the other hand, is detailed.

10.1 The Relationship Between the Exact and the Approximate Probabilities

In relating, the exact probabilities of the inner loop nodes to the approximate probabilities found upon loopy propagation, the Bayesian network from Figure 10.1 given the observation $D = d$, is considered. The following now builds upon the observation that the updating of a message vector during propagation can be captured by a transition matrix. First, the matrices that describe the information that is included into a message vector during one clockwise cycle, and during one counterclockwise cycle respectively, from the inner loop node A back to itself will be derived. The eigenvalues of these matrices then are used to express the relationship between the exact and approximate probabilities found at node A. In the sequel a transition matrix that captures the update of a message vector during an entire cycle will be termed a *reflexive* transition matrix.

To derive the reflexive transition matrix that captures the information that is added during one clockwise cycle from node A back to itself, the updating of the initial diagnostic message $\lambda_C(A) = (1, 1)$ during the first cycle of the algorithm is considered. In the first step of the

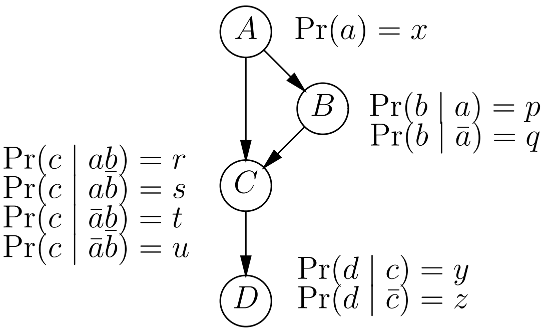

A
$\Pr(a) = x$
B
$\Pr(b \mid a) = p$
$\Pr(b \mid \bar{a}) = q$
$\Pr(c \mid ab) = r$
$\Pr(c \mid a\bar{b}) = s$
$\Pr(c \mid \bar{a}b) = t$
$\Pr(c \mid \bar{a}\bar{b}) = u$
C
$\Pr(d \mid c) = y$
$\Pr(d \mid \bar{c}) = z$
D

Figure 10.1: A multiply connected Bayesian network with a convergence node C having the dependent parents A and B, and the child D.

algorithm, node A sends the causal message

$$\pi_B(A) = \begin{bmatrix} x \\ 1 - x \end{bmatrix}$$

to node B, which subsequently sends the message vector

$$\pi_C(B) = \begin{bmatrix} p \cdot x + q \cdot (1 - x) \\ (1 - p) \cdot x + (1 - q) \cdot (1 - x) \end{bmatrix}$$

to node C. The diagnostic message that C receives from node D equals

$$\lambda_D(C) = \begin{bmatrix} y \\ z \end{bmatrix}$$

Since node C does not have any other children, its compound diagnostic parameter also equals

$$\lambda(C) = \begin{bmatrix} y \\ z \end{bmatrix}$$

This compound diagnostic parameter and the causal message that node C receives from node B are combined with the information that node C has about its own conditional probabilities, into the following diagnostic message from node C back to node A

$$\lambda_C(A) = \begin{bmatrix} \lambda(c) \cdot \big(r \cdot \pi_C(b) + s \cdot \pi_C(\bar{b})\big) + \lambda(\bar{c}) \cdot \big((1 - r) \cdot \pi_C(b) + (1 - s) \cdot \pi_C(\bar{b})\big) \\ \lambda(c) \cdot \big(t \cdot \pi_C(b) + u \cdot \pi_C(\bar{b})\big) + \lambda(\bar{c}) \cdot \big((1 - t) \cdot \pi_C(b) + (1 - u) \cdot \pi_C(\bar{b})\big) \end{bmatrix}$$

After the first clockwise cycle of the algorithm, therefore, the initial diagnostic message $(1, 1)$ has been updated to the message $\lambda_C(A)$ given above. The reflexive transition matrix

$$M^{\circlearrowright A,d} = \begin{bmatrix} l & m \\ n & o \end{bmatrix}$$

that captures this update message now is found as follows. First of all, the entries l, m, n and o of the matrix should adhere to

$$\begin{bmatrix} l & m \\ n & o \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \lambda_C(A)$$

from which it is found that

$$\begin{array}{rcl} l + m & = & \lambda_C(a) \\ n + o & = & \lambda_C(\bar{a}) \end{array}$$

The expression for $\lambda_C(a)$ thus has to split into separate terms for l and m and the expression for $\lambda_C(\bar{a})$ has to split into separate terms for n and o. To this end it is observed that in the analysis above the first component of the causal message from node A to node B pertains to a and the second component pertains to $\bar{a}$. Roughly speaking, since the first component is multiplied by l and n, these two matrix entries have to collect all information at node B concerning a. Since the conditional probability $p = \Pr(b \mid a)$ pertains to a, therefore, all terms containing p are assigned to l and n. Likewise, all terms containing q are assigned to m and o. By rearranging the various terms in the expressions for $\lambda_C(a)$ and $\lambda_C(\bar{a})$ accordingly, it follows that

$$\begin{array}{rcl} l & = & \Big(\big(y \cdot r + z \cdot (1-r)\big) \cdot p + \big(y \cdot s + z \cdot (1-s)\big) \cdot (1-p)\Big) \cdot x \\ m & = & \Big(\big(y \cdot r + z \cdot (1-r)\big) \cdot q + \big(y \cdot s + z \cdot (1-s)\big) \cdot (1-q)\Big) \cdot (1-x) \\ n & = & \Big(\big(y \cdot t + z \cdot (1-t)\big) \cdot p + \big(y \cdot u + z \cdot (1-u)\big) \cdot (1-p)\Big) \cdot x \\ o & = & \Big(\big(y \cdot t + z \cdot (1-t)\big) \cdot q + \big(y \cdot u + z \cdot (1-u)\big) \cdot (1-q)\Big) \cdot (1-x) \end{array}$$

Note that from $x, p, q, r, s, t, u, y, z \in [0, 1]$ it follows that $l, m, n, o \in [0, 1]$.

Analogously, the matrix that captures the information that is included during a single counterclockwise cycle into the messages from node A back to itself is found to be the transpose of the matrix above:

$$M^{\circlearrowleft A,d} = \begin{bmatrix} l & n \\ m & o \end{bmatrix}$$

Upon loopy propagation, the two transition matrices are applied repeatedly to the clockwise and counterclockwise messages from node A back to itself. Note that since all message vectors are normalised in Pearl's algorithm, the repeated multiplication by the transition matrices will not result in convergence to $(0, 0)$.

The eigenvalues of the transition matrices now can be exploited to relate the approximate probabilities found in the equilibrium state of the algorithm for the inner loop node A to its exact probabilities. The eigenvalues λ_1 and λ_2 of the matrix $M^{\circlearrowleft A,d}$ equal

$$\lambda_{1,2} = \frac{1}{2} \cdot \Big((l + o) \pm \sqrt{(l+o)^2 - 4 \cdot (l \cdot o - m \cdot n)}\Big)$$

where λ_1 is the largest of the two values; since $M^{\circlearrowleft A,d} = (M^{\circlearrowright A,d})^T$ for $M^{\circlearrowright A,d}$ the same eigenvalues are found. Note that $(l+o)^2 - 4 \cdot (l \cdot o - m \cdot n)$ can also be written as $(l-o)^2 + 4 \cdot m \cdot n$.

Since the entries of the two matrices are positive this expression is positive. The eigenvalues λ_1 and λ_2 thus are real numbers and λ_1, $\lambda_1 + \lambda_2$ and $\lambda_1 - \lambda_2$ are positive. The relationship between the exact and the approximate probabilities of A given d now is expressed by

$$\Pr(a_i \mid d) = \widetilde{\Pr}(a_i \mid d) - \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot \big(2 \cdot \widetilde{\Pr}(a_i \mid d) - 1\big)$$

The eigenvectors of the reflexive matrices are used to prove this property. Note that for each eigenvalue of $M^{\circlearrowright A,d}$ and for each eigenvalue of $M^{\circlearrowleft A,d}$, an eigenvector direction is found. For $M^{\circlearrowright A,d}$, the normalised principal eigenvector will be denoted by (α_1, β_1); (γ_1, δ_1) is a fixed arbitrary vector in the second eigenvector direction. For $M^{\circlearrowleft A,d}$, the normalised principal eigenvector will be denoted by (α_2, β_2). The reflexive matrices are applied repeatedly to the cycling messages within the loop under study. According to the power lemma of Strang [61], if $|\lambda_1| > |\lambda_2|$, that is, if λ_1 is positive and unique, the messages will converge to the normalised principal eigenvectors of the reflexive matrices, that is, to the eigenvectors with the largest eigenvalue. In the equilibrium state of the network, the approximate marginal probability distribution computed for the inner loop node A thus equals

$$\widetilde{\Pr}(A \mid d) = cst_1 \cdot \begin{bmatrix} \alpha_1 \cdot \alpha_2 \\ \beta_1 \cdot \beta_2 \end{bmatrix}$$

where cst_1 is a normalisation constant.

The computed approximate probabilities $\widetilde{\Pr}(a_i \mid d)$ now are related to the exact probabilities $\Pr(a_i \mid d)$ as follows. The entries l and o of the transition matrices $M^{\circlearrowright A,d}$ and $M^{\circlearrowleft A,d}$, are equal to $l = \Pr(d \mid a) \cdot \Pr(a)$ and $o = \Pr(d \mid \bar{a}) \cdot \Pr(\bar{a})$. For the exact probabilities $\Pr(a_i \mid d)$ it thus holds that $\Pr(a \mid d) = l/(l+o)$ and $\Pr(\bar{a} \mid d) = o/(l+o)$. To express $\Pr(a_i \mid d)$ and $\widetilde{\Pr}(a_i \mid d)$ in similar terms, the entries l and o now are related to the expressions $\alpha_1 \cdot \alpha_2$ and $\beta_1 \cdot \beta_2$. To this end, the matrix $M^{\circlearrowright A,d}$ is diagonalised into

$$\begin{aligned} M^{\circlearrowright A,d} &= \begin{bmatrix} \alpha_1 & \gamma_1 \\ \beta_1 & \delta_1 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \cdot \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} \\ &= \begin{bmatrix} \alpha_1 \cdot \mathcal{A} \cdot \lambda_1 + \gamma_1 \cdot \mathcal{C} \cdot \lambda_2 & \alpha_1 \cdot \mathcal{B} \cdot \lambda_1 + \gamma_1 \cdot \mathcal{D} \cdot \lambda_2 \\ \beta_1 \cdot \mathcal{A} \cdot \lambda_1 + \delta_1 \cdot \mathcal{C} \cdot \lambda_2 & \beta_1 \cdot \mathcal{B} \cdot \lambda_1 + \delta_1 \cdot \mathcal{D} \cdot \lambda_2 \end{bmatrix} \end{aligned}$$

where $\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} = \begin{bmatrix} \alpha_1 & \gamma_1 \\ \beta_1 & \delta_1 \end{bmatrix}^{-1}$. So,

$$\begin{aligned} l &= \alpha_1 \cdot \mathcal{A} \cdot \lambda_1 + \gamma_1 \cdot \mathcal{C} \cdot \lambda_2 \\ o &= \beta_1 \cdot \mathcal{B} \cdot \lambda_1 + \delta_1 \cdot \mathcal{D} \cdot \lambda_2 \end{aligned}$$

From $\begin{bmatrix} \alpha_1 & \gamma_1 \\ \beta_1 & \delta_1 \end{bmatrix} \cdot \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ it now follows that $\gamma_1 \cdot \mathcal{C} = \beta_1 \cdot \mathcal{B}$ and $\delta_1 \cdot \mathcal{D} = \alpha_1 \cdot \mathcal{A}$. The entries l and o can therefore also be written as

$$\begin{array}{rcl} l & = & \alpha_1 \cdot \mathcal{A} \cdot \lambda_1 + \beta_1 \cdot \mathcal{B} \cdot \lambda_2 \\ o & = & \beta_1 \cdot \mathcal{B} \cdot \lambda_1 + \alpha_1 \cdot \mathcal{A} \cdot \lambda_2 \end{array}$$

To express $\mathcal{A}$ and $\mathcal{B}$ in terms of α_2 and β_2, the matrix $M^{\circlearrowleft A,d}$ is rewritten as

$$M^{\circlearrowleft A,d} = (M^{\circlearrowright A,d})^T = \begin{bmatrix} \mathcal{A} & \mathcal{C} \\ \mathcal{B} & \mathcal{D} \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 & \beta_1 \\ \gamma_1 & \delta_1 \end{bmatrix}$$

The first column of the matrix $\begin{bmatrix} \mathcal{A} & \mathcal{C} \\ \mathcal{B} & \mathcal{D} \end{bmatrix}$ is a vector in the direction of the principal eigenvector (α_2, β_2) of $M^{\circlearrowleft A,d}$. So, $\mathcal{A} = cst_2 \cdot \alpha_2$ and $\mathcal{B} = cst_2 \cdot \beta_2$, where cst_2 is a normalisation constant. Using these expressions, it now follows that

$$\begin{array}{rcl} l & = & cst_2 \cdot (\alpha_1 \cdot \alpha_2 \cdot \lambda_1 + \beta_1 \cdot \beta_2 \cdot \lambda_2) \\ & = & cst_2/cst_1 \cdot \big(\widetilde{\Pr}(a \mid d) \cdot \lambda_1 + \widetilde{\Pr}(\bar{a} \mid d) \cdot \lambda_2\big) \\ o & = & cst_2 \cdot (\beta_1 \cdot \beta_2 \cdot \lambda_1 + \alpha_1 \cdot \alpha_2 \cdot \lambda_2) \\ & = & cst_2/cst_1 \cdot \big(\widetilde{\Pr}(\bar{a} \mid d) \cdot \lambda_1 + \widetilde{\Pr}(a \mid d) \cdot \lambda_2\big) \end{array}$$

With $\Pr(a \mid d) = l/(l+o)$ and $\Pr(\bar{a} \mid d) = o/(l+o)$, now is found that

$$\Pr(a_i \mid d) = \widetilde{\Pr}(a_i \mid d) - \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot \big(2 \cdot \widetilde{\Pr}(a_i \mid d) - 1\big)$$

For $\bar{d}$, the derivation is analogous. For node the inner loop node B similar expressions are found. The transition matrix for an entire cycle may, more or less, be viewed as the result of multiplication of the transition matrices between the neighbouring nodes in the cycle. The clockwise matrices for nodes A and B now result from the multiplication of the same matrices, in the same order yet with a different ‘starting point’. As a consequence, the clockwise reflexive matrices for nodes A and B have the same eigenvalues. The same applies to the counterclockwise matrices.

The following example illustrates the iterative updating of the approximate probabilities during loopy propagation.

Example 10.1 *Consider the example Bayesian network from Figure 10.2. After entering the evidence $D = d$ into the network, the following reflexive matrices for the inner loop node A are found*

$$M^{\circlearrowright A,d} = \begin{bmatrix} 0.0540 & 0.6192 \\ 0.1240 & 0.5408 \end{bmatrix}$$

and

$$M^{\circlearrowleft A,d} = \begin{bmatrix} 0.0540 & 0.1240 \\ 0.6192 & 0.5408 \end{bmatrix}$$

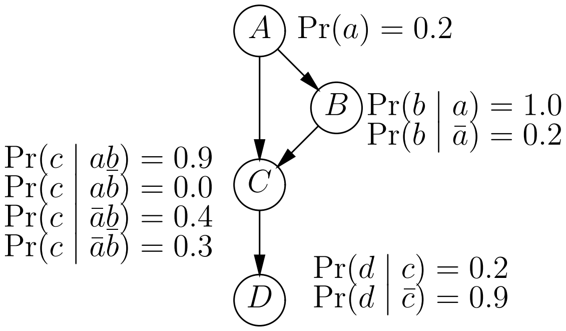


Figure 10.2: An example Bayesian network

The approximate probabilities $\widetilde{\Pr}^{it^1}(A \mid d)$ *found after the first cycle equal the vector that results after the component-wise multiplication of the vectors*

$$\begin{bmatrix} 0.0540 & 0.6192 \\ 0.1240 & 0.5408 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.6732 \\ 0.6648 \end{bmatrix}$$

and

$$\begin{bmatrix} 0.0540 & 0.1240 \\ 0.6192 & 0.5408 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.1780 \\ 1.1600 \end{bmatrix}$$

after normalisation, and equal $\widetilde{\Pr}^{it^1}(A \mid d) \approx (0.1345, 0.8655)$*. Likewise, the approximate probabilities established for node* A *after in the second cycle equal the vector that results after the component-wise multiplication of the vectors*

$$\begin{bmatrix} 0.0540 & 0.6192 \\ 0.1240 & 0.5408 \end{bmatrix} \cdot \left(\begin{bmatrix} 0.0540 & 0.6192 \\ 0.1240 & 0.5408 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0.4480 \\ 0.4430 \end{bmatrix}$$

and

$$\begin{bmatrix} 0.0540 & 0.1240 \\ 0.6192 & 0.5408 \end{bmatrix} \cdot \left(\begin{bmatrix} 0.0540 & 0.1240 \\ 0.6192 & 0.5408 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0.1535 \\ 0.7375 \end{bmatrix}$$

after normalisation, and equal $\widetilde{\Pr}^{it^2}(A \mid d) \approx (0.1738, 0.8262)$*, and so on.*

Table 10.1 lists the approximate probability distributions $\widetilde{\Pr}^{it^i}(a \mid d)$ *and* $\widetilde{\Pr}^{it^i}(a \mid \bar{d})$ *for node* A *for the first five cycles. Given the observation* $D = d$*, the algorithm had converged within a bound of* 0.0001 *in six cycles. Given the observation* $D = \bar{d}$*, the algorithm had converged within the same bound in seven cycles. The consecutive approximations are shown graphically in Figures 10.3 and 10.4; for comparison, the exact probabilities are depicted as well. The two figures shows that the approximate probabilities asymptotically approach their final value. The approximate probabilities given* d *oscillate around the final value, whereas given* $\bar{d}$ *the approximations go steadily towards the final value. In Section 10.2, this difference in approximation behaviour will be discussed.*

Table 10.1: The approximate probability distributions of $\widetilde{\Pr}^{it^i}(a \mid d)$ and $\widetilde{\Pr}^{it^i}(a \mid \bar{d})$ for node A in the network from Figure 10.2 in the first five cycles of the loopy-propagation algorithm.

i	1	2	3	4	5
$\widetilde{\Pr}^{it^i}(a \mid d)$	0.1345	0.1738	0.1696	0.1701	0.1700
$\widetilde{\Pr}^{it^i}(a \mid \bar{d})$	0.3297	0.2928	0.2848	0.2831	0.2827

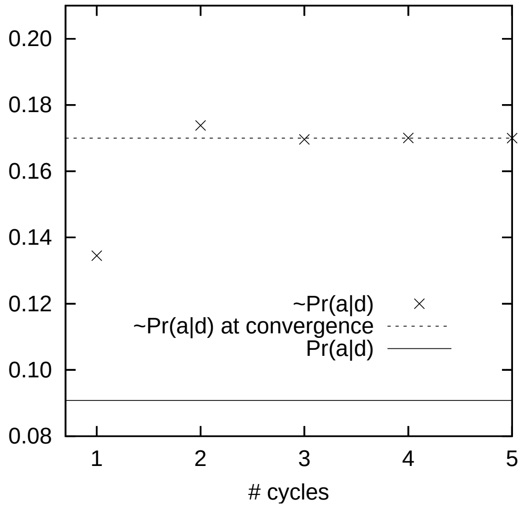


Figure 10.3: The approximate probabilities $\widetilde{\Pr}^{it^i}(a \mid d)$ found from the network of Figure 10.2 during the first five cycles of the loopy propagation algorithm.

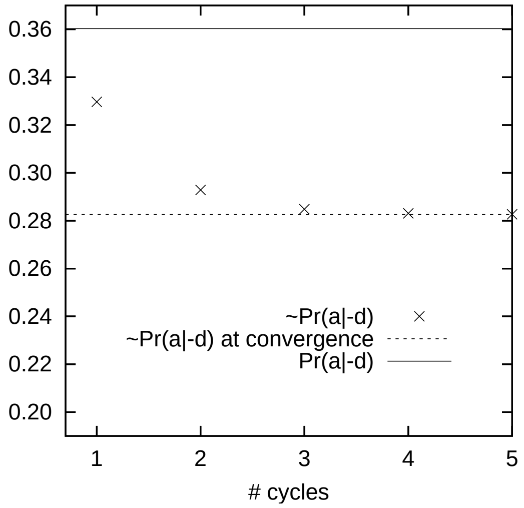


Figure 10.4: The approximate probabilities $\widetilde{\Pr}^{it^i}(a \mid \bar{d})$ found from the network of Figure 10.2 during the first five cycles of the loopy propagation algorithm.

The eigenvalues of the reflexive $M^{\circlearrowright A,d}$ *are* $\lambda_1 \approx 0.6662$ *and* $\lambda_2 \approx -0.0714$*; its normalised principal eigenvector is*

$$\begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} \approx \begin{bmatrix} 0.5028 \\ 0.4972 \end{bmatrix}$$

The eigenvalues of $M^{\circlearrowleft A,d}$ *are again* $\lambda_1 \approx 0.6662$ *and* $\lambda_2 \approx -0.0714$*; its normalised principal eigenvector equals*

$$\begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} \approx \begin{bmatrix} 0.1685 \\ 0.8315 \end{bmatrix}$$

The approximate probabilities found for node A *upon loopy propagation given* d *now are equal to*

$$cst_1 \cdot \begin{bmatrix} 0.5028 \cdot 0.1685 \\ 0.4972 \cdot 0.8315 \end{bmatrix} \approx \begin{bmatrix} 0.1700 \\ 0.8300 \end{bmatrix}$$

At convergence, thus, $\widetilde{\Pr}(a \mid d) \approx 0.1700$ *and* $\widetilde{\Pr}(\bar{a} \mid d) \approx 0.8300$*. Recall that the exact probability* $\Pr(a \mid d)$ *equals* $l/(l+o)$*. It can be read from the diagonal of either of the two transition matrices to be* $0.0540/0.5948 \approx 0.0908$*. For the relationship between* $\Pr(a \mid d)$ *and* $\widetilde{\Pr}(a \mid d)$ *now indeed is observed that*

$$\begin{aligned} \Pr(a \mid d) &= \widetilde{\Pr}(a \mid\ d) - \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot \big(2 \cdot \widetilde{\Pr}(a \mid d) - 1\big) \\ &\approx 0.1700 + \frac{0.0714}{0.5948} \cdot (2 \cdot 0.1700 - 1) \approx 0.098 \end{aligned}$$

To conclude this section, it is noted that the above derivation of the relationship between the exact and approximate probabilities for an inner loop node in terms of reflexive matrices applies to simple loops only. For non-simple loops, unfortunately, the changes in the various messages in the loop cannot be described anymore by a reflexive matrix. In more complex loops at least one of the loop nodes is incident on more than two loop arcs and merges the incoming messages from other loop nodes before passing them on. The changes included in the messages that cycle in the loop thus are non-linear. Consider as an example the graph from Figure 10.5, which includes a double loop. A message from node D to node B will, return to D from the direction of both C and A. Node D then merges the two messages by component wise multiplication before sending them on to B again.

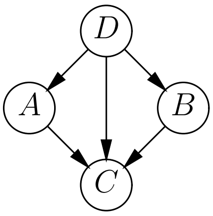


Figure 10.5: An example Bayesian network including a double loop.

10.2 The Decisiveness of the Approximations

In the previous section, the error that may arise in the probabilities computed upon loopy propagation for an inner loop node was discussed for a binary Bayesian network with a simple loop. More specifically, an expression relating the exact probabilities for the inner loop node to the computed approximate probabilities was derived. The current section builds upon this expression to state some properties of the approximations in terms of the specification of the network. An approximate probability is *overconfident* if it is closer to one of the extremes, that is to 0 or 1, than the exact probability; the approximation is *underconfident* if it is closer to 0.5 [47]. The term *decisiveness* will be used to refer to either the over- or the underconfidence of an approximate probability. As an example, Figure 10.6 depicts, for the network from Figure 10.2, the exact and approximate probabilities $\Pr(a \mid d)$ and $\widetilde{\Pr}(a \mid d)$ as a function of $\Pr(a)$; Figure 10.7 depicts $\Pr(a \mid \bar{d})$ and $\widetilde{\Pr}(a \mid \bar{d})$. From the figures, it is readily seen that the evidence d results in underconfident approximations, while the evidence $\bar{d}$ gives overconfident approximate probabilities for all possible values of $\Pr(a)$. We will argue that the approximations for the inner nodes of a loop are either all pushed towards overconfidence or all pushed towards underconfidence as soon as the convergence node of the loop has an observed descendant. We further argue that the decisiveness of the approximations depends on the sign of the qualitative influence between the parents of the convergence node and the sign of the intercausal influence that is induced between these parents by the entered evidence.

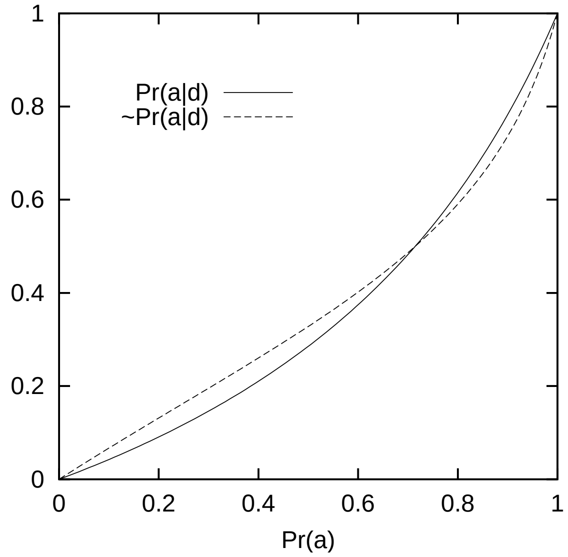


Figure 10.6: $\Pr(a \mid d)$ and $\widetilde{\Pr}(a \mid d)$ as a function of $\Pr(a)$ for the network from Figure 10.2.

In Chapter 2, the notions of qualitative influence and intercausal influence were introduced. The sign of the qualitative influence between the nodes A and B in the example network from Figure 10.1 can straightforwardly be inferred from the specification of the network and equals $\Pr(b \mid a) - \Pr(b \mid \bar{a}) = p - q$. The intercausal influence that is induced between the two nodes by the evidence d is derived below. The exact probability distributions for nodes A and B are

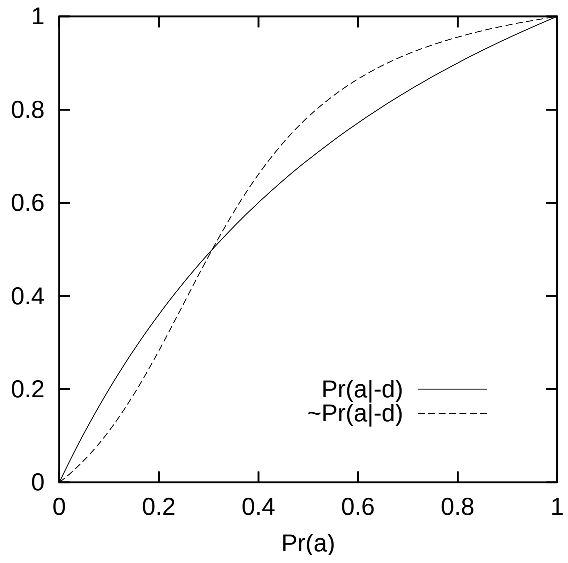


Figure 10.7: $\Pr(a \mid \bar{d})$ and $\widetilde{\Pr}(a \mid \bar{d})$ as a function of $\Pr(a)$ for the network from Figure 10.2.

assumed to be non-degenerate, that is, all marginal probabilities for A and B are assumed to be elements of $\langle 0, 1\rangle$; note that for degenerate distributions in fact no loop is present and the algorithm will yield exact probabilities. To separate the intercausal influence from the direct qualitative influence in the relationship between A and B, the network from Figure 10.1 without the arc (A, B) is considered. The intercausal influence then is captured by

$$\begin{aligned}
\Pr(a \mid bd) - \Pr(a \mid \bar{b}d) &= \frac{\Pr(abd)}{\Pr(bd)} - \frac{\Pr(a\bar{b}d)}{\Pr(\bar{b}d)} \\
&= \frac{x \cdot e}{x \cdot e + (1-x) \cdot g} - \frac{x \cdot f}{x \cdot f + (1-x) \cdot h} \\
&= \frac{(x - x^2) \cdot (e \cdot h - f \cdot g)}{\big(x \cdot e + (1-x) \cdot g\big) \cdot \big(x \cdot f + (1-x) \cdot h\big)}
\end{aligned}$$

where

$$\begin{aligned}
e &= r \cdot y + (1-r) \cdot z \\
f &= s \cdot y + (1-s) \cdot z \\
g &= t \cdot y + (1-t) \cdot z \\
h &= u \cdot y + (1-u) \cdot z
\end{aligned}$$

Because the denominator of the expression for $\Pr(a \mid bd) - \Pr(a \mid \bar{b}d)$ is positive, the sign of the intercausal influence that is induced between A and B is equal to the sign of the numerator $(x - x^2) \cdot (e \cdot h - f \cdot g)$. Because $x = \Pr(a) \in \langle 0, 1\rangle$, moreover, the sign of the intercausal influence equals the sign of $e \cdot h - f \cdot g$, that is, the sign of

$$(y - z)^2 \cdot (r \cdot u - s \cdot t) + z \cdot (y - z) \cdot (r + u - s - t)$$

Similarly, the sign of the intercausal influence induced by the evidence $\bar{d}$ is found to be equal to the sign of

$$(z-y)^2 \cdot (r \cdot u - s \cdot t) + (1-z) \cdot (z-y) \cdot (r+u-s-t)$$

In the sequel, these two expressions will be denoted by $I\!I(A, B \mid d_j)$, $d_j \in \{d, \bar{d}\}$.

We now relate the qualitative influence between the nodes A and B and the sign of the additional influence between those nodes, induced by the observation of D to the over- or underconfidence of the approximate probabilities computed for the inner loop nodes by the loopy-propagation algorithm. Recall that, since all entries of the transition matrix $M^{\circlearrowleft A,d}$ are positive, its eigenvalues λ_1 and λ_2 are real numbers andλ_1, $\lambda_1 + \lambda_2$ and $\lambda_1 - \lambda_2$ are positive. From the relationship

$$\Pr(a_i \mid d_j) = \widetilde{\Pr}(a_i \mid d_j) - \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot \big(2 \cdot \widetilde{\Pr}(a_i \mid d_j) - 1\big)$$

established in the previous section, therefore, it follows that overconfident approximations will be computed for the probability $\Pr(a_i \mid d_j)$ if $\lambda_2 > 0$ and underconfident approximations will be found if $\lambda_2 < 0$. This is substantiated as follows. Given that $\lambda_2 < 0$ it is readily seen from the formula above that for $\widetilde{\Pr}(a_i \mid d_j) < 0.5$ is found that $\Pr(a_i \mid d_j) < \widetilde{\Pr}(a_i \mid d_j)$ and that for $\widetilde{\Pr}(a_i \mid d_j) > 0.5$ is found that $\Pr(a_i \mid d_j) > \widetilde{\Pr}(a_i \mid d_j)$. The approximation is closer to 0.5 than the actual probability and is thus underconfident. In order to substantiate that for $\lambda_2 > 0$ overconfident approximations are found the equation above is rewritten as

$$\widetilde{\Pr}(a_i \mid d_j) = \Pr(a_i \mid d_j) + \frac{\lambda_2}{\lambda_1 - \lambda_2} \cdot (2 \cdot \Pr(a_i \mid d_j) - 1)$$

Now given that $\lambda_2 > 0$, it is readily seen from the formula above that for $\Pr(a_i \mid d_j) < 0.5$ is found that $\widetilde{\Pr}(a_i \mid d_j) < \Pr(a_i \mid d_j)$ and that for $\Pr(a_i \mid d_j) > 0.5$ is found that $\widetilde{\Pr}(a_i \mid d_j) > \Pr(a_i \mid d_j)$. The approximations are closer to the extremes 0 and 1 than the actual probabilities and thus are overconfident. Note that for $\Pr(a_i \mid d_j) = 0.5$, loopy propagation will result in the exact probability, irrespective of the eigenvalues of the reflexive matrices of the inner loop nodes. This observation is easily verified by computing $\widetilde{\Pr}(a_i \mid d_j)$ from the relationship $\Pr(a_i \mid d_j) = \widetilde{\Pr}(a_i \mid d_j) - \lambda_2/(\lambda_1 + \lambda_2) \cdot (2 \cdot \widetilde{\Pr}(a_i \mid d_j) - 1)$ for $\Pr(a_i \mid d_j) = 0.5$.

$$\begin{aligned}
0.5 &= \widetilde{\Pr}(a_i \mid d_j) - \frac{\lambda_2}{\lambda_1 + \lambda_2} \cdot (2 \cdot \widetilde{\Pr}(a_i \mid d_j) - 1) \Leftrightarrow \\
0.5 &= \frac{\widetilde{\Pr}(a_i \mid d_j) \cdot (\lambda_1 - \lambda_2) + \lambda_2}{\lambda_1 + \lambda_2} \Leftrightarrow \\
0.5 \cdot (\lambda_1 + \lambda_2) &= \widetilde{\Pr}(a_i \mid d_j)(\lambda_1 - \lambda_2) + \lambda_2 \Leftrightarrow \\
0.5 \cdot (\lambda_1 - \lambda_2) &= \widetilde{\Pr}(a_i \mid d_j)(\lambda_1 - \lambda_2) \Leftrightarrow \\
0.5 &= \widetilde{\Pr}(a_i \mid d_j)
\end{aligned}$$

From $\lambda_2 = \frac{1}{2} \cdot \Big((l+o) - \sqrt{(l+o)^2 - 4 \cdot (l \cdot o - m \cdot n)}\Big)$, it follows that the sign of the eigenvalue λ_2 equals the sign of the expression $l \cdot o - m \cdot n$. Simple manipulation of the terms involved, shows that, the sign of this expression is equal to the sign of

$$(p-q) \cdot I\!I(A, B \mid d_j)$$

The approximate probabilities given the evidence d_j, established for node A thus are overconfident if the sign of $p - q$ is equal to the sign of $II(A, B \mid d_j)$, that is, if the sign of the qualitative influence between nodes A and B is equal to the sign of the additional influence that is induced between A and B by the evidence d_j; the approximations are underconfident if these signs are opposite.

Example 10.2 *Consider again the network from Figure 10.2. The sign of the intercausal influence between the nodes A and B equals the sign of* $1.0 - 0.2 = 0.8$ *and hence is positive. The sign of the additional influence that is induced between nodes A and B by the evidence d, equals the sign of* $(0.2-0.9)^2 \cdot (0.9 \cdot 0.3 - 0.0 \cdot 0.4) + 0.9 \cdot (0.2-0.9) \cdot (0.9+0.3-0.0-0.4) \approx -0.37$ *and thus is negative. The qualitative and intercausal influences between the nodes A and B therefore have opposite signs and the approximate probabilities established for the two inner loop nodes upon loopy propagation will be underconfident. Indeed, the underconfident approximations* $\widetilde{\Pr}(a \mid d) \approx 0.17$ *and* $\widetilde{\Pr}(b \mid d) \approx 0.30$ *are found for the exact probabilities* $\Pr(a \mid d) \approx 0.09$ *and* $\Pr(b \mid d) \approx 0.26$*. Given the evidence $\bar{d}$, on the other hand, the sign of the intercausal influence between A and B equals the sign of* $(0.9 - 0.2)^2 \cdot (0.9 \cdot 0.3 - 0.0 \cdot 0.4) + (1.0-0.9) \cdot (0.9-0.2) \cdot (0.9+0.3-0.0-0.4) \approx 0.19$ *and hence is positive. The qualitative and intercausal influences now have equal signs and overconfident approximate probabilities will result upon loopy propagation. Indeed, the overconfident approximations* $\widetilde{\Pr}(a \mid \bar{d}) \approx 0.28$ *and* $\widetilde{\Pr}(b \mid \bar{d}) \approx 0.52$ *are found for the exact probabilities* $\Pr(a \mid \bar{d}) \approx 0.36$ *and* $\Pr(b \mid \bar{d}) \approx 0.51$.

Recall that the reflexive matrices of all inner loop nodes of a simple loop have identical eigenvalues. All inner loop nodes therefore will have an identical decisiveness. As mentioned above, exact probabilities will be found for the inner loop nodes upon loopy propagation whenever $\lambda_2 = 0$, that is, whenever $(p - q) \cdot II(A, B \mid d_j) = 0$. If the factor $p - q$ equals zero, then the nodes A and B are independent in the network in its prior state and in fact no loop is present. If the factor $II(A, B \mid d_j)$ equals zero, then the message that node C sends to node B is independent of the probabilities found at A, and vice versa. Nodes A and B will then receive the correct messages from node C and the algorithm will converge to the correct marginal distributions for A and B in just a single cycle.

Example 10.3 *Consider the network from Figure 10.2 with the following specification:* $\Pr(a) = 0.5$, $\Pr(b \mid d) = 0.9$, $\Pr(b \mid \bar{d}) = 0.1$, $\Pr(c \mid ab) = 0.8$, $\Pr(c \mid a\bar{b}) = 0.2$, $\Pr(c \mid \bar{a}b) = 0.4$, $\Pr(c \mid \bar{a}\bar{b}) = 0.1$, $\Pr(d \mid c) = 1$ *and* $\Pr(d \mid \bar{c}) = 0$*. Given this specification, the observation of $D = d$ induces an intercausal influence with a zero sign between nodes A and B. Indeed, given $D = d$, the correct posterior probabilities* $\Pr(a \mid c) \approx 0.8506$ *and* $\Pr(b \mid c) \approx 0.8736$ *are found.*

At this point also the difference in approximation behaviour of the loopy-propagation algorithm demonstrated after entering, respectively, the observations $D = d$ and $D = \bar{d}$ in the network from Figure 10.2 can be explained. Recall that Figure 10.4 pertains to the algorithm's approximation behaviour for node A after entering the observation $\bar{d}$ in the network from Figure 10.2. The qualitative influence between the nodes A and B in this network is positive. The observation $\bar{d}$, moreover, induces a positive intercausal influence between A and B, that is, a higher probability b induces the influence of the observation of $\bar{d}$ on A along the trail $D \leftarrow C \leftarrow A$

to be ‘more positive’. Now consider the behaviour of the loopy propagation algorithm during a clockwise cycle. If, in a particular cycle, the influence of the observation $\bar{d}$ on A along the trail $D \leftarrow C \leftarrow A$ is positive, then, because of the positive qualitative influence between A and B, the influence on B along the trail $D \leftarrow C \leftarrow A \rightarrow B$ will be positive as well; the clockwise process thus increases b. Because of the positive intercausal, this will strengthen the positive influence of the observation of $\bar{d}$ on A along the trail $D \leftarrow C \leftarrow A$ and thus also strengthen the positive influence on B along the trail $D \leftarrow C \leftarrow A \rightarrow B$. This will again strengthen the positive influence of the observation of $\bar{d}$ on A along the trail $D \leftarrow C \leftarrow A$, and so on, until convergence of the algorithm. In the counterclockwise cycles of the algorithm a similar behaviour is observed. The approximate probabilities thus steadily go towards their final values, as shown in Figure 10.4. Alternatively, if the qualitative influence and the intercausal influence between the nodes A and B have opposite signs, the successive approximate probabilities will oscillate towards the final approximation as shown in Figure 10.3. In the example network from Figure 10.2, the observation of d, induces a negative intercausal influence between A and B, that is, a higher probability b induces the influence of the observation d on A along the trail $D \leftarrow C \leftarrow A$ to be ‘less positive’. Now consider again the behaviour of the loopy propagation algorithm during a clockwise cycle. If, in a particular cycle, the influence of the observation d on A along the trail $D \leftarrow C \leftarrow A$ is positive, then, as in the case of the observation $\bar{d}$, the clockwise process will increase the probability b. Now, however, because of the negative intercausal influence between nodes A and B, this increase of b will result in a ‘less positive’ influence of the observation of d on the node A along the trail $D \leftarrow C \leftarrow A$ and thus also in a ‘less positive’ influence on B along the trail $D \leftarrow C \leftarrow A \rightarrow B$. The clockwise process thus now results in a decrease of b compared to the previous cycle. As a result, in the next cycle, the influence of the observation d on A will be ‘more positive’ than in the previous cycle and so on, until convergence of the algorithm. In the counterclockwise cycles of the algorithm a similar behaviour is observed. Both processes add up to the oscillating behaviour depicted in Figure 10.3 for the example network. Recall from Section 10.1 that the cycling process convergences if the largest eigenvalue of the reflexive matrices is positive and unique. Intuitively, the convergence of the algorithm can be explained by the fact that the influence of an observation weakens when it is transported along a trail. Each cycle of the algorithm now can be considered to add to the length of the trail along which the influence of the observation is transported.

In the analysis in this chapter, an observation was entered for node D in the network from Figure 10.1. A similar analysis holds, however, if the convergence node C itself is observed. The expressions for the intercausal influence between the nodes A and B then reduce to $r \cdot u - s \cdot t$ and $r \cdot u - s \cdot t - (r + u - s - t)$ after the observation of c and $\bar{c}$, respectively. Furthermore, the influence of the cycling error was analysed only for the simple network from Figure 10.1. The analysis, however, extends directly to networks with a single simple loop with more than two inner nodes and to networks with a single simple loop in which the loop nodes have additional neighbours outside the loop. For such networks, however, the computation of the qualitative influence and of the intercausal influence induced between the parents of the convergence nodes may be more involved. The essence of the analysis also extends to networks with multiple simple loops. In such networks, however, the effect of the cycling error within a loop may be obscured by errors entering the loop from outside.

To conclude, a network with a single simple loop having two or more convergence nodes is considered. In such a network, a cycling error occurs only if for each of the loop's convergence node, either the node itself or one of its descendants is observed. Note that an intercausal influence is then is induced between the parent nodes of each convergence node. Now consider an arbitrary convergence node C with the parents A and B on the loop; C is called the primary convergence node of the loop. Because the loop is simple, there are exactly two trails between the nodes A and B on the loop. The sign of the indirect influence between A and B along the trail not containing C can then be considered the qualitative influence between A and B; the intercausal influence between the nodes A and B that is induced by observation for node C or for one of its descendants acts as the intercausal influence. The decisiveness of the approximate probabilities established for the inner loop nodes can now be derived, as before, from the signs of these two influences. Because, effectively this procedure comes down to $\otimes$-combining the involved signs, the convergence node C, used for establishing the decisiveness, indeed can be chosen arbitrarily.

Example 10.4 *Consider the example network from Figure 10.8, including a simple loop with two convergence nodes. Given the evidence d and f, the loopy-propagation algorithm will compute approximate probabilities for the inner loop nodes A and B. To establish the decisiveness of these approximations, the signs of the two influences between A and B are established. Taking node C for the primary convergence node of the loop, the sign of the intercausal influence between A and B given d equals the sign of $(0.2-0.9)^2 \cdot (0.9 \cdot 0.3 - 0.0 \cdot 0.4) + 0.9 \cdot (0.2-0.9) \cdot (0.9+0.3-0.0-0.4) \approx -0.37$ and hence is negative. The sign of the qualitative influence between A and B along the trail $A \rightarrow E \leftarrow B$ now equals the sign of the intercausal influence between A and B given f. This sign equals the sign of $(0.2-0.8)^2 \cdot (0.7 \cdot 1.0 - 0.2 \cdot 0.5) + 0.8 \cdot (0.2-0.8) \cdot (0.7+1.0-0.2-0.5) \approx -0.26$ and hence also is negative. Since the two signs are equal, it follows that the approximations for the inner loop nodes A and B will be overconfident. Indeed the overconfident approximations $\widetilde{\Pr}(a \mid d) \approx 0.36$ and $\widetilde{\Pr}(b \mid d) \approx 0.31$ for the probabilities $\Pr(a \mid d) \approx 0.38$ and $\Pr(b \mid d) \approx 0.40$ are found. It is easily seen that exactly the same computations are made with E as the primary convergence node.*

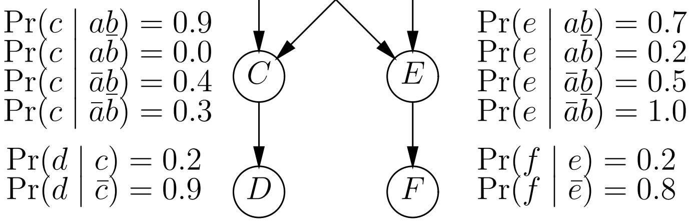

$\Pr(a) = 0.2$
A
B
$\Pr(b) = 0.3$
$\Pr(c \mid a\underline{b}) = 0.9$
$\Pr(c \mid ab) = 0.0$
$\Pr(c \mid \bar{a}\underline{b}) = 0.4$
$\Pr(c \mid \bar{a}b) = 0.3$
C
E
$\Pr(e \mid a\underline{b}) = 0.7$
$\Pr(e \mid ab) = 0.2$
$\Pr(e \mid \bar{a}\underline{b}) = 0.5$
$\Pr(e \mid \bar{a}b) = 1.0$
$\Pr(d \mid c) = 0.2$
$\Pr(d \mid \bar{c}) = 0.9$
D
F
$\Pr(f \mid e) = 0.2$
$\Pr(f \mid \bar{e}) = 0.8$

Figure 10.8: An example Bayesian network, containing a loop with two convergence nodes

Chapter 11

The Convergence Error in Markov Networks

In Chapter 8, was argued that upon loopy propagation in Bayesian networks two different types of error may arise: the cycling error and the convergence error. Now, any Bayesian network can be converted into an equivalent pairwise Markov network, and for such networks, an algorithm equivalent to the loopy propagation algorithm for Bayesian networks exists [71, 77]. In [71] Weiss studied this equivalent algorithm for pairwise Markov networks and he established the relationship between the exact and the approximate probabilities for such networks with just a simple loop. In Markov networks, upon loopy propagation, all errors in the computed probabilities arise as a result of the cycling of information; there appeared to be no equivalent for the convergence error. This raised the question how the convergence error is embedded in loopy propagation in Markov networks. This chapter argues that the convergence error in a Bayesian network is converted to a cycling error in the equivalent Markov network. It shows furthermore, that the prior convergence error in Markov networks is characterised by the fact that the relationship between the exact and the approximate probabilities as established in [71] does not exist for the node in which this error arises. In Section 11.1 the conversion of a Bayesian network into a Markov network is described, in Section 11.2 the analysis of Weiss of loopy propagation in Markov networks is discussed and in Section 11.3 the convergence error in a Markov network is traced.

11.1 The Conversion of a Bayesian Network into a Markov Network

In the conversion of a Bayesian network into a Markov network, for any node with multiple parents, an auxiliary node is constructed into which the common parents are clustered. This auxiliary node is connected to the child and its parents and the original arcs between child and parents are removed. Thereafter, all arcs are replaced by edges. The clusters are all pairs of connected nodes; for a cluster with an auxiliary node and a former parent node, the potential is set to 1 if the nodes have the same value for the former parent node and to 0 otherwise. For the other clusters, the potentials are equal to the conditional probabilities of the former child given the former parent. The prior probability of a former root node is incorporated by multiplication into one of the potentials of the clusters in which it takes part.

Example 11.1 *The Bayesian network from Figure 11.1 can be converted into the pairwise Markov network from Figure 11.2. This Markov network has the clusters AB, AX, BX and XC. The values of node X are composed of the value combinations of nodes A and B; node X has the values x_{ab}, $x_{a\bar{b}}$, $x_{\bar{a}b}$ and $x_{\bar{a}\bar{b}}$. Given that the prior probability of root node A is incorporated in the potential of cluster AB, the network has the following potentials: $\psi(AB) = \Pr(B \mid A) \cdot \Pr(A)$; $\psi(XC) = \Pr(C \mid AB)$; $\psi(AX) = 1$ if the value of X for A equals the value of A and is zero otherwise and $\psi(BX) = 1$ the value of X for B equals the value of B and is zero otherwise. The joint probability distribution of the Bayesian network from Figure 11.1 is given by*

$$\Pr(ABC) = \Pr(B \mid A) \cdot \Pr(A) \cdot \Pr(C \mid AB)$$

and the joint probability distribution over A, B and C of the Markov network in Figure 11.2 is given by

$$\begin{aligned} \Pr(ABC) &= \sum_X \psi(AB) \cdot \psi(AX) \cdot \psi(BX) \cdot \psi(XC) \\ &= \Pr(B \mid A) \cdot \Pr(A) \cdot \Pr(C \mid AB) \end{aligned}$$

These two joint probability distributions are equivalent.

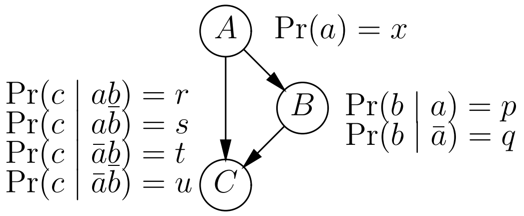

A
$\Pr(a) = x$
B
$\Pr(b \mid a) = p$
$\Pr(b \mid \bar{a}) = q$
$\Pr(c \mid ab) = r$
$\Pr(c \mid a\bar{b}) = s$
$\Pr(c \mid \bar{a}b) = t$
$\Pr(c \mid \bar{a}\bar{b}) = u$
C

Figure 11.1: A multiply connected Bayesian network with a convergence node C having the dependent parents A and B.

11.2 An Analysis of Loopy Propagation in Markov Networks

Weiss [71] analysed the performance of the loopy-propagation algorithm for Markov networks with a single loop and related the approximate probabilities found for the nodes in the loop to their exact probabilities. He noted that upon application of the algorithm, messages cycle in the

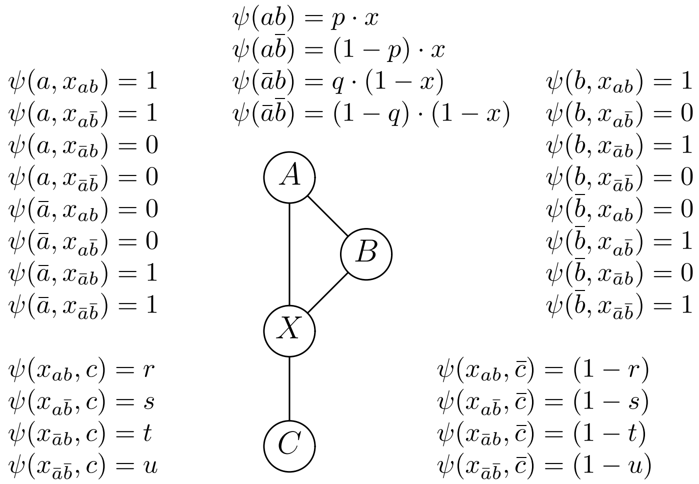

$\psi(ab) = p \cdot x$
$\psi(a\bar{b}) = (1-p) \cdot x$
$\psi(\bar{a}b) = q \cdot (1-x)$
$\psi(\bar{a}\bar{b}) = (1-q) \cdot (1-x)$
$\psi(a, x_{ab}) = 1$
$\psi(a, x_{a\bar{b}}) = 1$
$\psi(a, x_{\bar{a}b}) = 0$
$\psi(a, x_{\bar{a}\bar{b}}) = 0$
$\psi(\bar{a}, x_{ab}) = 0$
$\psi(\bar{a}, x_{a\bar{b}}) = 0$
$\psi(\bar{a}, x_{\bar{a}b}) = 1$
$\psi(\bar{a}, x_{\bar{a}\bar{b}}) = 1$
$\psi(b, x_{ab}) = 1$
$\psi(b, x_{a\bar{b}}) = 0$
$\psi(b, x_{\bar{a}b}) = 1$
$\psi(b, x_{\bar{a}\bar{b}}) = 0$
$\psi(\bar{b}, x_{ab}) = 0$
$\psi(\bar{b}, x_{a\bar{b}}) = 1$
$\psi(\bar{b}, x_{\bar{a}b}) = 0$
$\psi(\bar{b}, x_{\bar{a}\bar{b}}) = 1$
A
B
X
C
$\psi(x_{ab}, c) = r$
$\psi(x_{a\bar{b}}, c) = s$
$\psi(x_{\bar{a}b}, c) = t$
$\psi(x_{\bar{a}\bar{b}}, c) = u$
$\psi(x_{ab}, \bar{c}) = (1-r)$
$\psi(x_{a\bar{b}}, \bar{c}) = (1-s)$
$\psi(x_{\bar{a}b}, \bar{c}) = (1-t)$
$\psi(x_{\bar{a}\bar{b}}, \bar{c}) = (1-u)$

Figure 11.2: A pairwise Markov network that represents the same joint probability distribution over the nodes A, B and C as the Bayesian network from Figure 11.1.

loop and errors emerge as a result of the double counting of information. The main idea of his analysis is that for a node in the loop, two reflexive matrices can be derived; one capturing the change of a message vector cycling clockwise back to the node it started and one capturing the change of a message vector cycling counterclockwise. The probability distribution computed by the loopy-propagation algorithm for the loop node in the steady state, can now be inferred from the principal eigenvectors of the reflexive matrices plus the incoming vectors from outside the loop. Subsequently, he showed that the reflexive matrices also include the exact probability distribution and he used those two observations to derive an analytical relationship between the approximate and the exact probabilities.

Below the derivation of this relationship in Markov networks is described in more detail. Consider a pairwise Markov network with a single simple loop with the loop nodes $L^1 ... L^n, n \geq 3$ and with connected to each node L^i in the loop, an observed node O^i as shown in Figure 11.3; the transition matrices $M^{L^iL^{i+1}}$ are denoted by M^i and the transition matrices $M^{L^{i+1}L^i}$ are denoted by M^{i^T}. During propagation, a node O^i will repeatedly send the same message to the loop node L^i. This vector is one of the columns of the transition matrix $M^{O^iL^i}$ In order to enable the incorporation of this message into the reflexive loop matrices, this vector is transformed

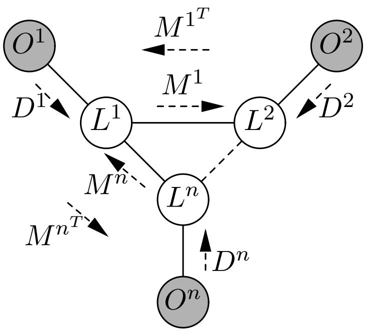


Figure 11.3: An example pairwise Markov network including a single simple loop.

into a diagonal matrix D^i, with the vector elements on the diagonal. For example, suppose that $M^{O^iL^i} = \begin{bmatrix} p & r \\ q & s \end{bmatrix}$ and suppose that the observation $O^i = o_1^i$ is entered into the network, then $D^i = \begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}$. The reflexive matrix $M^{\circlearrowleft L^1}$ for the transition of a counterclockwise message from node L^1 back to itself is defined as $M^{1^T} D^2 ... M^{n-1^T} D^n M^{n^T} D^1$. The message that L^2 sends to L^1 in the steady state now is in the direction of the principal eigenvector of $M^{\circlearrowleft L^1}$. The reflexive matrix $M^{\circlearrowright L^1}$ for the transition of a clockwise message from node L^1 back to itself is defined as $M^n D^n M^{n-1} D^{n-1} ... M^1 D^1$. The message that node L^n sends to L^1 in the steady state is in the direction of the principal eigenvector of $M^{\circlearrowright L^1}$. Component wise multiplication of the two principal eigenvectors and the message from O^1 to L^1, and normalisation of the resulting vector, yields a vector of which the components equal the approximate probabilities of L^1 in the steady state [1]. Furthermore, Weiss proved that the elements on the diagonals of the reflexive matrices equal the correct probabilities of the relevant value of L^1 and the evidence, for example, $M_{1,1}^{\circlearrowleft L^1}$ equals $\Pr(l_1^1, \mathbf{o})$. Subsequently, he related the exact probabilities of a node L in the loop to its approximate probabilities by

$$\Pr(l_i) = \frac{\lambda_1 \widetilde{\Pr}(l_i) + \sum_{j=2} P_{ij} \lambda_j P_{ji}^{-1}}{\sum_j \lambda_j}$$

where P is a matrix that is composed of the eigenvectors of C, with the principal eigenvector in the first column, and $\lambda_1 ... \lambda_j$ are the eigenvalues of the reflexive matrices, with λ_1 the maximum eigenvalue. Note that from this formula it follows that correct probabilities will be found if λ_1 equals 1 and all other eigenvalues equal 0.

In this analysis, all nodes O^i are considered observed. Note that given unobserved nodes outside the loop, the analysis is essentially the same. In that case, however, a transition matrix $M^{O^iL^i} = \begin{bmatrix} p & r \\ q & s \end{bmatrix}$ will result in the diagonal matrix $D^i = \begin{bmatrix} p+r & 0 \\ 0 & q+s \end{bmatrix}$.

11.3 Tracing the Convergence Error

As showed in Section 8, in the analysis of loopy propagation in Bayesian networks a distinction can be made between the cycling error and the convergence error. For Markov networks, however, such a distinction does not exist. All errors result from the cycling of information and, on first sight, there is no equivalent for the convergence error. Yet, any Bayesian network can be converted into an equivalent pairwise Markov network for which an algorithm equivalent to the loopy-propagation algorithm in Bayesian networks exists. This section investigates how the convergence error is embedded in loopy propagation in Markov networks. To do so, the Markov

[1] Note that in the definitions of the matrices $M^{\circlearrowleft L^1}$ and $M^{\circlearrowright L^1}$, the matrix D^1 is the last element. As a result these two matrices are not each other transposes. In the analysis in Section 10.1 the reflexive matrices for loop node A were defined in such a way that they were each other transposes. As a result, the probabilities of A could be computed from just the eigenvectors of the reflexive loop matrices. In the option chosen by Weiss, the computation of approximate probabilities of node L^1, involves the component wise multiplication of the two eigenvectors and of the incoming vector from node O^1.

network equivalent to the simplest Bayesian network in which a convergence error may arise is studied. The focus thereby is on the node that replaces the convergence node in the loop.

The simplest Bayesian network in which a convergence error may arise is the network from Figure 11.1 in its prior state. In its prior state, there is no cycling of information, and exact probabilities are computed for nodes A and B. In node C, however, a convergence error may arise. The network can be converted into the equivalent pairwise Markov network from Figure 11.2. This network has the following transition matrices

$$M^{XA} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M^{XB} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$M^{XC} = \begin{bmatrix} r & s & t & u \\ 1-r & 1-s & 1-t & 1-u \end{bmatrix}$$

$$M^{AB} = \begin{bmatrix} p \cdot x & q \cdot (1-x) \\ (1-p) \cdot x & (1-q) \cdot (1-x) \end{bmatrix}$$

and their transposes. In the prior state of the network, C will send the message $M^{CX} \cdot (1,1) = (1,1,1,1)$ to X. In order to enable the incorporation of this message into the reflexive loop matrices it is transformed into D^{CX} which results in the 4x4-identity matrix.

Now first consider the performance of the equivalent loopy propagation algorithm for the inner loop node A. This node has the following reflexive matrices for its clockwise and counter-clockwise messages respectively:

$$M^{\circlearrowright A} = M^{XA} \cdot D^{CX} \cdot M^{BX} \cdot M^{AB} = \begin{bmatrix} x & 1-x \\ x & 1-x \end{bmatrix}$$

with eigenvalues 1 and 0 and principal eigenvector (1,1) and

$$M^{\circlearrowleft A} = M^{BA} \cdot M^{XB} \cdot D^{CX} \cdot M^{AX} = \begin{bmatrix} x & x \\ 1-x & 1-x \end{bmatrix}$$

with eigenvalues 1 and 0 and principal eigenvector $(x, 1-x)$. Note that the correct probabilities for node A indeed are found on the diagonal of the reflexive matrices. Furthermore, $\lambda_1 = 1$ and $\lambda_2 = 0$ and therefore correct approximations are expected. Indeed is found that the approximations $(1 \cdot x, 1 \cdot (1-x))$ equal the exact probabilities. Note also that,

$$\begin{bmatrix} x & 1-x \\ x & 1-x \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} x & x \\ 1-x & 1-x \end{bmatrix} \cdot \left(\begin{bmatrix} x & x \\ 1-x & 1-x \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \alpha \cdot \begin{bmatrix} x \\ 1-x \end{bmatrix}$$

Thus, as expected, the messages from node A back to itself do not change any more after the first cycle. As in the Bayesian network, for node A, no cycling of information occurs in the Markov network. For node B a similar evaluation can be made.

Now consider the convergence node C. In the Bayesian network in its prior state a convergence error may arise in this node. In the conversion of the Bayesian network into the Markov network, the convergence node C is placed outside the loop and the auxiliary node X is added. For X, the following reflexive matrices are computed for the clockwise and counterclockwise messages from node X back to itself respectively

$$\begin{aligned} M^{\circlearrowright X} &= M^{BX} \cdot M^{AB} \cdot M^{XA} \cdot D^{CX} \\ &= \begin{bmatrix} p \cdot x & p \cdot x & q \cdot (1-x) & q \cdot (1-x) \\ (1-p) \cdot x & (1-p) \cdot x & (1-q) \cdot (1-x) & (1-q) \cdot (1-x) \\ p \cdot x & p \cdot x & q \cdot (1-x) & q \cdot (1-x) \\ (1-p) \cdot x & (1-p) \cdot x & (1-q) \cdot (1-x) & (1-q) \cdot (1-x) \end{bmatrix} \end{aligned}$$

with eigenvalues 1, 0, 0, 0, principal eigenvector $\left(\frac{p \cdot x + q \cdot (1-x)}{(1-p) \cdot x + (1-q) \cdot (1-x)}, 1, \frac{p \cdot x + q \cdot (1-x)}{(1-p) \cdot x + (1-q) \cdot (1-x)}, 1\right)$ and with other eigenvectors $(0, 0, -1, 1)$, $(-1, 1, 0, 0)$ and $(0, 0, 0, 0)$.

$$\begin{aligned} M^{\circlearrowleft X} &= M^{AX} \cdot M^{BA} \cdot M^{XB} \cdot D^{CX} \\ &= \begin{bmatrix} p \cdot x & (1-p) \cdot x & p \cdot x & (1-p) \cdot x \\ p \cdot x & (1-p) \cdot x & p \cdot x & (1-p) \cdot x \\ q \cdot (1-x) & (1-q) \cdot (1-x) & q \cdot (1-x) & (1-q) \cdot (1-x) \\ q \cdot (1-x) & (1-q) \cdot (1-x) & q \cdot (1-x) & (1-q) \cdot (1-x) \end{bmatrix} \end{aligned}$$

with eigenvalues 1, 0, 0, 0, principal eigenvector $\left(\frac{x}{1-x}, \frac{x}{1-x}, 1, 1\right)$ and with other eigenvectors $(0, -1, 0, 1)$, $(-1, 0, 1, 0)$ and $(0, 0, 0, 0)$.

On the diagonal of the reflexive matrices of X the probabilities $\Pr(AB)$ are found. Since in a network in its prior state, the elements of the diagonal of its reflexive matrices equal the correct probabilities for a loop, the probabilities $\Pr(AB)$ can be considered the exact probabilities for node X. The approximate probabilities yielded upon loopy propagation for node X equal the normalised vector of the component wise multiplication of the principal eigenvectors of the two reflexive matrices of X and the incoming vector from node C, which equals $(1, 1, 1, 1)$. The components of this vector equal the normalised probabilities $\Pr(A) \cdot \Pr(B)$.

A first observation is that λ_1 equals one and the other eigenvalues equal zero. According to the equation

$$\Pr(l_i) = \frac{\lambda_1 \widetilde{\Pr}(l_i) + \sum_{j=2} P_{ij} \lambda_j P_{ji}^{-1}}{\sum_j \lambda_j}$$

therefore, exact probabilities should be computed for node X upon loopy-propagation. However, the computed probabilities are $\alpha \cdot \Pr(A) \cdot \Pr(B)$ whereas the exact probabilities are $\Pr(AB)$ and these probabilities are not equal in general. This apparent contradiction in results now can be explained by the fact that the relationship between the exact and the approximate probabilities as given above, does not exist for node X in the prior state of the network. For node X in the prior network, the matrix P in the expression, includes a column $(0,0,0,0)$. This matrix thus is singular and therefore, the matrix P^{-1}, which is needed in the expression of relationship between the exact and approximate probabilities, does not exist. Further note that the messages from node X back to itself, in contrast with the messages from nodes A and B back to itself, may still change after the first cycle. Thus, although in the Bayesian network there is no cycling of information, in the Markov network, for node X information cycles, resulting in errors computed for its probabilities.

The probabilities computed by the loopy-propagation algorithm for node C in the Markov network from Figure 11.2 equal the normalised product $M^{XC} \cdot m$, where m is the message vector with the approximate probabilities found at node X, that is, these probabilities equal

$$\alpha \cdot \begin{bmatrix} r & s & t & u \\ 1-r & 1-s & 1-t & 1-u \end{bmatrix} \cdot \begin{bmatrix} \Pr(a)\cdot\Pr(b) \\ \Pr(a)\cdot\Pr(\bar{b}) \\ \Pr(\bar{a})\cdot\Pr(b) \\ \Pr(\bar{a})\cdot\Pr(\bar{b}) \end{bmatrix}$$

These approximate probabilities indeed equal the approximate probabilities found in the equivalent Bayesian network. Note that if node X would send its exact probabilities, that is, if node X would send the vector $\Pr(AB)$, exact probabilities for node C would be computed. In the Markov network the convergence error thus is founded in the cycling of information for the auxiliary node X.

In Section 9.2.1 an expression for the size of the prior convergence error in the approximate probability computed for the value c_i of convergence node C in the network from figure 11.1 was given. It was argued that part of the factors of this expression capture the degree of dependency between the parents of the convergence node and that one of its factors determines the extent to which this dependency can affect the computed probability. This last factor is composed of the conditional probabilities of the value c_i of node C. In the current analysis the effect of the degree of dependence between A and B is reflected in the difference between the exact and the approximate probabilities found for node X. The effect of the conditional probabilities of c_i emerges in the transition of the message vector from X to C.

In the analysis so far, the small example network from Figure 11.1 was considered. Note, however, that for any prior binary Bayesian networks with only simple loops with a single convergence node, the situation for any convergence node can be ‘summarised’ to the situation in Figure 11.1 by marginalisation over the relevant variables. The results of the analysis for the network from Figure 11.1, therefore, apply to any simple loop in a binary Bayesian network in its prior state. Note that two loops in sequence may result in incorrect probabilities entering the second loop. The reflexive matrices, however, will have a similar structure as the reflexive matrices derived in this section, which implies that for all the auxiliary nodes of the convergence nodes the expression that relates the approximate to the exact probabilities does not exist. Note also that, given a loop with multiple convergence nodes, in the prior state of the network, the parents of the convergence nodes are independent and effectively no loop is present.

In general, as soon as a diagnostic observation is made for a convergence node, the matrix P is not singular any more and the exact and approximate probabilities for node X indeed can be related by the expression as established by Weiss. There is an exception, however. As discussed in Section 10.2, if in a binary network a diagnostic observation for a convergence node results in an intercausal influence of zero between the parents of this convergence node, than the second eigenvalue of the reflexive matrices of the inner loop nodes equals zero and exact probabilities are computed for the inner nodes. In that case, as in the prior network, also for the auxiliary node only one eigenvalue is not equal to zero and thus according to the expression derived by Weiss, exact probabilities should be computed for this node, whereas again, this is not the case. It was expected therefore that also given a diagnostic observation that induces a intercausal influence of zero between the parents of a convergence node, the matrix P with the eigenvectors of the auxiliary node X will be singular. Several examples confirmed this assumption

The reflexive matrices appeared to have a feature that may indicate an interesting area for future research. For a reflexive matrices of the example network from Figure 11.1 is found is found that, given the observation of $C = c$, the product of its eigenvalues unequal to zero equals $(x-x^2)\cdot(p-q)\cdot(r\cdot u-s\cdot t)$. The factor $(r\cdot u-s\cdot t)$, in this expression is equal to the expression which is used to compute the sign of the intercausal influence or product synergy, between A and B that is induced by the observation of c. The expression $(x - x^2) \cdot (p - q) \cdot (r \cdot u - s \cdot t)$ shows an analogy to the expression $(x - x^2) \cdot (p - q) \cdot (r - s - t + u)$ with which the size of the prior convergence error in the example network can be established. In Section 9.1, a new notion was defined that quantified and generalised the notion of the additive synergy as given for binary networks. This notion was used in the general expression for the prior convergence error. An interesting question now is whether an meaningful analogous extension of the idea of product synergy exists.

Chapter 12

Conclusions

The loopy-propagation algorithm for computing approximate probabilities from multiply connected Bayesian networks has been reported to show good experimental results on real-life networks. The basic idea of the algorithm is to apply Pearl's standard propagation algorithm for singly connected networks to Bayesian networks with loops. While for singly connected networks the algorithm yields exact probability distributions, it includes errors in the distributions established for multiply connected networks. Two different types of error were identified that may arise in the probabilities computed by the algorithm: the convergence error and the cycling error. Convergence errors arise at each convergence node and are found both in a network's prior and posterior state. Cycling errors arise in loop nodes as soon as for each convergence node of the loop, either the node itself, or one of its descendents is observed.

The various factors that govern the size of the prior convergence error found in the probability computed for some value of the convergence node were detailed. In this respect, first the notion *quantitative parental synergy* was defined. This notion is related to the notion of additive synergy in binary networks, and can be computed from the conditional probabilities specified for the convergence node. The prior convergence error was found to be composed of factors that captures the degree of the dependency between the parents of the convergence node and of the quantitative parental synergies. These synergies determine the degree to which the dependency between the parent nodes of the convergence node can affect the computed probabilities. Informally speaking, the more dependent the parents of the convergence node and the larger the quantitative parental synergies, the larger the convergence error will be. It was found that the minimum and the maximum value of a convergence error is dependent on the number of parents of the convergence node. The minimum and maximum equal, respectively, -0.5 and 0.5 if there are two parents, and approach -1 and 1 as the number of parents approaches infinity.

When observations are entered into a network, loopy propagation results in posterior errors at the convergence nodes that may differ in size from the prior ones. Such a posterior error may include a posterior convergence error and may include a cycling error that has entered the node from inside the loop. The posterior convergence error was separated from the cycling error and studied in isolation. It was found that, for causal observations, the posterior convergence error is governed by the same factors as the prior convergence error. The maximum posterior convergence error given a causal observation therefore is, like the prior convergence error, dependent on the number of parents of the convergence node. A diagnostic observation, on the other hand, fundamentally changes the expression of the convergence error. The minimum and the maximum of the posterior convergence error can, given such an observation, even if the convergence node has only two parents, approach -1 and 1. With respect to the cycling error entering the convergence nodes from inside the loop was found that, given binary parents of the convergence nodes, as a result of the cycling error the approximations were shifted along a line, defined by the exact causal messages from the parents of the convergence node, and the saddle point of surface that captures the approximate probabilities found for the convergence node.

For binary Bayesian networks with simple loops, moreover, the factors that determine the effect of the cycling error on the decisiveness of the approximations computed for the inner loop nodes were identified. It was found that this effect depends on the sign of the qualitative influence between the parents of the convergence node of the loop and the sign of the intercausal influence that is induced between these parents; the approximations are overconfident if these signs are equal and underconfident if these signs are opposite. So far, the effect of the cycling error on the decisiveness of the approximations computed for the inner nodes of a loop was studied in isolation. An overall analysis, involving multiple compound loops, unfortunately, is much more complicated. In a network with multiple loops, for example, approximate probabilities may enter a loop as a result of errors introduced in other parts of the network and will have their own effect on the resulting approximations.

Loopy propagation has also been studied by the analysis of the performance of an equivalent algorithm in pairwise Markov networks with just a simple loop. In these networks all errors result from the cycling of information and at first sight there is no equivalent for the convergence error. How the convergence error is embedded in loopy propagation in Markov networks was investigated by studying the pairwise Markov equivalent to the simplest Bayesian network in which a convergence error may occur. The convergence error was found to be converted to a cycling error in the equivalent Markov network. Furthermore, it was found that the prior convergence error is characterised by the fact that the relationship between the exact probabilities and the approximate probabilities yielded by loopy propagation, as established by Weiss, does not exist for the loop node in which this error occurs. The same observation was made for posterior networks in which the observation of the convergence node or one of its descendants induced a zero intercausal influence between the parents of the convergence node.

To conclude, for a simple loop, there appeared to be an analogy between the expression for the prior convergence error and the product of the eigenvalues unequal to zero of the reflexive matrices of the inner loop nodes given an observation for the convergence node. The expressions for both consist of a part that reflects the dependency between the parents of the convergence node; the expression for the prior convergence error moreover includes the formula that is used to establish the sign of the additive synergy for the convergence node. The expression for the product of the eigenvalues is supplemented with the formula that is used to establish the sign of the product synergy with respect to the observed value of the convergence node. In Section 9.1, the notion of quantitative parental synergy was defined that quantified and generalised the notion of the additive synergy as given for binary networks. This notion was used in the general expression for the prior convergence error. An interesting question is whether a meaningful analogous extension of the product synergy exists.

Bibliography

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: turbo-codes. In *IEEE International Conference on Communications*, pages 1064–1070, 1993.

[2] J.H. Bolt. Loopy propagation: the convergence error in markov networks. In M. Studeny and J. Vomlel, editors, *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, pages 43 – 50, 2006.

[3] J.H. Bolt, S. Renooij, and L.C. van der Gaag. Upgrading ambiguous signs in QPNs. In C. Meek and U. Kjaerulff, editors, *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 73 – 80. Morgan Kaufmann Publishers, San Francisco, California, 2003.

[4] J.H. Bolt and L.C. van der Gaag. On the convergence error in loopy propagation. In N. Taatgen R. Verbrugge and L. Schomaker, editors, *Proceedings of the Sixteenth Belgium-Netherlands Conference on Artificial Intelligence*, pages 267 – 274, 2004.

[5] J.H. Bolt and L.C. van der Gaag. *Advances in Probabilistic Graphical Models, Studies in Fuzziness and Soft Computing*, volume 213, chapter Decisiveness in loopy propagation, pages 153 – 173. Springer, Berlin, 2007.

[6] J.H. Bolt, L.C. van der Gaag, and S. Renooij. Introducing situational influences in QPNs. *International Journal of Approximate Reasoning*, 38:333 – 354, 2005.

[7] J. Cano, L. Hern'andez, and S. Moral. Importance sampling algorithms for the propagation of probabilities in belief networks, 1996.

[8] G. Casella and C. Robert. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.

[9] A.S. Cofiño, R. Cano, C. Sordo, and J.M. Gutiérrez. Bayesian networks for probabilistic weather prediction. In F. van Harmelen, editor, *Proceedings of the Fifteenth Eureopean Conference on Artificial Intelligence*, pages 695 – 699. IOS Press, 2002.

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

[11] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, 1999.

[12] P. Dagum and M. Luby. Approximate inference in Bayesian networks is NP hard. *Artificial Intelligence*, 60:141–153, 1993.

[13] M.J. Druzdzel. *Probabilistic Reasoning in Decision Support Systems: From Computation to Common Sense*. PhD thesis, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1993.

[14] M.J. Druzdzel and M. Henrion. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 548–553. AAAI Press, Menlo Park, California, 1993.

[15] M.J. Druzdzel and M. Henrion. Intercausal reasoning with uninstantiated ancestor nodes. In D. Heckerman and A. Mamdani, editors, *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pages 317–325. Morgan Kaufmann Publishers, San Mateo, California, 1993.

[16] M.J. Druzdzel and L.C. van der Gaag. Elicitation of probabilities for belief networks: combining qualitative and quantitative information. In Ph. Besnard and S. Hank, editors, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 141–148. Morgan Kaufmann Publishers, San Francisco, 1995.

[17] G. Elidan, I. McGraw, and D. Koller. Residual belief propagation: informed scheduling for asynchronous message passing. In *The Twenty-second Conference on Uncertainty in Artificial Intelligenc*, pages 165–173, 2006.

[18] W.T. Freeman and E.C. Pasztor. Learning to estimate scenes from images. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, volume 11, pages 775 – 781. MIT Press Cambridge, MA, 1999.

[19] B.J. Frey. *Bayesian Networks for Pattern Classification, Data Compression and Channel Coding*. PhD thesis, University of Toronto, Canada, 1997.

[20] S.L. Lauritzen F.V. Jensen and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quaterly*, 4:269–282, 1990.

[21] P.L. Geenen, A.R.W. Elbers, L.C. van der Gaag, and W.L.A. van der Loeffen. Development of a probabilistic network for clinical detection of classical swine fever. In *Proceedings of the Eleventh Symposium of the International Society for Veterinary Epidemiology and Economics*, pages 667–669, 2006.

[22] A. Globerson and T. Jaakkola. Convergent propagation algorithms via oriented trees. In *Proceedings of the twentythird Conference on Uncertainty in Artificial Intelligence*, 2007.

[23] F. Harary. *Graph Theory*. Addison-Wesley, Reading, Massachusetts, 1969.

[24] T. Heskes. Stable fixed points of loopy belief propagation are local minima of the Bethe free energy. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 343–350, Cambridge, MA, 2003. MIT Press.

[25] T.M. Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 26:153–190, 2006.

[26] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.

[27] A.T. Ihler. Accuracy bounds for belief propagation. In *Proceedings of the twentythird Conference on Uncertainty in Artificial Intelligence*, pages 183–190, 2007.

[28] E.T. Jaynes and G.L Bretthorst. *Probability Theory: the Logic of Science*. Cambridge, 2003.

[29] F.V. Jensen. *An Introduction to Bayesian Networks*. Springer Verlag, New York, 1996.

[30] F.V. Jensen, U. Kjaerulff, B. Kristiansen, H. Langseth, C. Skaanning, J. Vomlel, and M. Vomlelova. The sacso methodology for troubleshooting complex systems. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 15:321–333, 2001.

[31] M.I. Jordan, Z. Ghahramani, T. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[32] M.I. Jordan and Y. Weiss. *The Handbook of Brain Theory and Neural Networks, 2nd edition*, chapter Graphical models: Probabilistic inference. MIT Press, Cambridge MA, 2002.

[33] R.J. Kennett, K.B. Korb, and A.E. Nicholson. Seabreeze prediction using Bayesian networks. In *Proceedings of the Fifth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, volume 2035 of *Series Lecture Notes in Computer Science*, pages 1611–3349. Springer Berlin/Heidelberg, 2001.

[34] A. N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. 1933.

[35] F.R. Kschischang and B.J. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal of Selected Areas in Communications*, 16(2):219–230, 1998.

[36] S.L. Lauritzen. *Graphical Models*, volume 17 of *Oxford Statistical Science Series*. Clarendon Press, Oxford, 1996.

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

[38] C-L. Liu and M.P. Wellman. Incremental tradeoff resolution in qualitative probabilistic networks. In *Proceedings of the Fourteenth International Conference on Uncertainty in Artificial Intelligence*, pages 338–345, Madison, WI, 1998.

[39] P.J.F. Lucas, N. de Bruijn, K. Schurink, and A. Hoepelman. A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU. *Artificial Intelligence in Medicine*, 19(3):251–279, 2000.

[40] T. Jaakkola M. J. Wainwright and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005.

[41] D. Mackay and R.M. Neal. Good codes based on very sparse matrices. In C. Boyd, editor, *Cryptography and Coding: 5th SIAM Conference*, volume 1025 of *Lecture Notes in Computer Science*, pages 100–111. Springer-Verlag, 1995.

[42] R.J. McEliece, D.J.C. MacKay, and J-F. Cheng. Turbo decoding as an instance of Pearl's ”belief propagation” algorithm. *IEEE Journal on Selected Area's in Communications*, 16(2):140–152, 1998.

[43] T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT Media Lab, 2001.

[44] T. Minka and Y. Qi. Tree-structured approximations by expectation propagation. In L. Saul S. Thrun and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, Cambridge MA, 2004.

[45] A. Montanari and T. Rizzo. How to compute loop corrections to the bethe approximation. *Journal of Statistical Mechanics: Theory and Experiment*, 10, October 2005.

[46] J.M. Mooij and H.J. Kappen. Loop corrections for approximate inference on factor graphs. *Journal of Machine Learning Research*, 8:1113–1143, May 2007.

[47] M.G. Morgan and M. Henrion. *Uncertainty, a Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge University Press, Cambridge, 1990.

[48] M.P.Wellman and M. Henrion. Explaining ”explaining away”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):287 – 292, 1993.

[49] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In K.B. Laskey and H. Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 467–475. Morgan Kaufmann Publishers, San Francisco, 1999.

[50] S. Parsons. Order of magnitude reasoning and qualitative probability. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(3):373 – 390, 2003.

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

[52] S. Renooij. *Qualitative Approaches to Quantifying Probabilistic Networks*. PhD thesis, Institute for Information and Computing Sciences, Utrecht University, 2001.

[53] S. Renooij and L.C. van der Gaag. Enhancing QPNs for trade-off resolution. In K. Laskey and H. Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 480–487. Morgan Kaufmann Publishers, San Francisco, 1999.

[54] S. Renooij and L.C. van der Gaag. From qualitative to quantitative probabilistic networks. In A. Darwiche and N. Friedman, editors, *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 422–429. Morgan Kaufmann Publishers, San Francisco, 2002.

[55] S. Renooij, L.C. van der Gaag, S. Green, and S. Parsons. Zooming in on trade-offs in qualitative probabilistic networks. In J. Etheredge and B. Manaris, editors, *Proceedings of the Thirteenth International FLAIRS Conference*, pages 303 – 307. AAAI Press, Menlo Park, California, 2000.

[56] S. Renooij, L.C. van der Gaag, and S. Parsons. Context-specific sign-propagation in qualitative probabilistic networks. *Artificial intelligence*, 140:207 – 230, 2002.

[57] S. Renooij, L.C. van der Gaag, and S. Parsons. Propagation of multiple observations in QPNs revisited. In F. van Harmelen, editor, *Proceedings of the Fifteenth European Conference on Artificial Intelligence*, pages 665–669. IOS Press, Amsterdam, 2002.

[58] J.A. Rice. *Matematical Statistics and Data Analysis*. Duxbury Press, California, 1995.

[59] T. Tanaka S. Ikeda and S. Amari. Information geometry of loopy bp. In *Supplementary Proceedings of ICANN/ICONIP*, pages 54 – 57, 2003.

[60] P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In J.F. Lemmer R.D. Shachter, T. Levitt and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence*, volume 4, pages 169–198. Elsevier, Amsterdam, 1990.

[61] G. Strang. *Linear Algebra and Its Applications*. Academic Press, 1980.

[62] Ch. Sutton and A. McCallum. Improved dynamic schedules for belief propagation. In *Proceedings of the twentythird Conference on Uncertainty in Artificial Intelligence*, 2007.

[63] Y.W. Teh and M. Welling. The unified propagation and scaling algorithm. In S. Becker T.G. Dietterich and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, Cambridge MA, 2002.

[64] L.C. van der Gaag. Probabilistic reasoning. Lecture Notes, 2002.

[65] L.C. van der Gaag, S. Renooij, C.L.M. Witteman, B.M.P. Aleman, and B.G. Taal. How to elicit many probabilities. In K.B. Laskey and H. Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 647–654, Morgan Kaufmann Publishers, San Francisco, California, 1999.

[66] L.C. van der Gaag, S. Renooij, C.L.M. Witteman, B.M.P. Aleman, and B.G. Taal. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25(2):123–148, 2002.

[67] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory*, 49(5):1120–1146, 2003.

[68] M.J. Wainwright. *Stochastic Processes on Graphs: Geometric and Variational Approaches*. PhD thesis, Department of EECS, Massachusetts Institute of Technology, 2002.

[69] H. Wasyluk, A. Onisko, and M.J. Druzdzel. Support of diagnosis of liver disorders based on a causal bayesian network model. *Medical Science Monitor*, 7:327–332, 2001.

[70] Y. Weiss. Interpreting images by propagating Bayesian beliefs. In M.I. Jordan M.C. Mozer and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 908–915, 1997.

[71] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[72] Y. Weiss and W.T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[73] M. Welling and Y. W. Teh. Belief optimization for binary networks: a stable alternative to loopy belief propagation. In J. Breese and D. Koller, editors, *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence,* pages 554 – 561. Morgan Kaufmann Publishers, San Francisco, 2001.

[74] M.P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–303, 1990.

[75] J.S Yedidia. *Advanced Mean Field Methods: Theory and Practice*, chapter An idiosyncratic journey beyond mean field theory. MIT Press, Cambridge, MA, 2001.

[76] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Bethe free energy, Kikuchi approximations and belief propagation algorithms. Technical Report TR-2001-16, MERL, 2001.

[77] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. International Joint Conference on Artificial Intelligence, Distinguished Lecture Track, 2001.

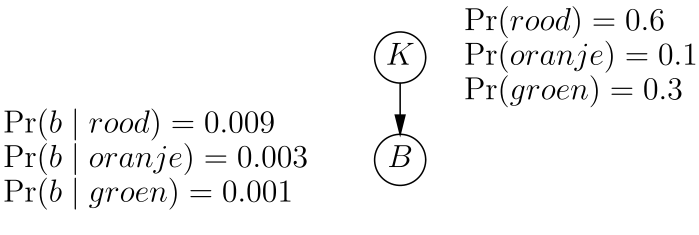
[78] A. Yuille. CCCP algorithms to minimise the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. In *Neural Comp.*, volume 14, pages 1691 – 1722, 2002.

[79] C. Zhang, S. Sun, and G. Yu. A Bayesian network approach to time series forecasting of short-term traffic flows. In *Proceedings of the Seventh International IEEE Conference on Intelligent Transportation Systems*, pages 216 – 221, 2004.

Samenvatting

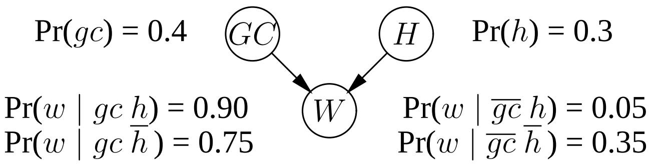
Informeel geformuleerd is een statistische variabele een kenmerk van iets dat met een bepaalde kans een bepaalde waarde kan aannemen. Zo is bijvoorbeeld, voor een bepaald stoplicht, de variabele $Kleur$ een kenmerk dat de waarden $rood$, $oranje$ en $groen$ kan aannemen. De waarden van een statistische variabele sluiten elkaar uit en bij elkaar opgeteld komen de kansen op de waarden uit op 1. De kansen op de waarden worden aangegeven met '$\Pr$'. Verder wordt bijvoorbeeld $Kleur = rood$ vaak afgekort tot $rood$. Voor het stoplicht zou bijvoorbeeld kunnen gelden dat $\Pr(rood) = 0.6$, $\Pr(oranje) = 0.1$ en $\Pr(groen) = 0.3$. Of een auto die het stoplicht passeert op de kruising erna een botsing heeft is ook een variabele. Deze variabele $Botsing$ kan de waarde ja of de waarde nee aannemen. Bij een variabele met de waarden ja en nee wordt het aannemen van de waarde ja vaak aangegeven met een kleine letter en het aannemen van de waarde nee met een kleine letter met een streepje erboven; dus $Botsing = ja$ wordt geschreven als b en $Botsing = nee$ als $\bar{b}$. Er zou bijvoorbeeld kunnen gelden dat $\Pr(b) = 0.006$ en $\Pr(\bar{b}) = 0.994$. Variabelen kunnen elkaar beïnvloeden. Zo beïnvloedt de kleur van het stoplicht de kans op een botsing. De kans op een botsing *gegeven dat het stoplicht op rood staat* zou bijvoorbeeld gelijk kunnen zijn aan 0.009 en de kans op een botsing *gegeven dat het stoplicht op groen staat* aan 0.001. Dit zijn voorwaardelijke kansen en die worden genoteerd als $\Pr(b \mid rood) = 0.009$ en $\Pr(b \mid groen) = 0.001$. Ook voorwaardelijke kansen tellen op tot 1; een auto heeft al dan niet een botsing gegeven een bepaalde kleur van het stoplicht. Met $\Pr(b \mid rood) = 0.009$ ligt dus bijvoorbeeld vast dat $\Pr(\bar{b} \mid rood) = 0.991$.

Met een Bayesiaans netwerk kan de kansverdeling over een verzameling statistische variabelen compact worden vastgelegd. Zo'n netwerk bestaat uit een gerichte graaf waarin de knopen de variabelen representeren en waarin de pijlen ruwweg de directe afhankelijkheden tussen de variabelen aangeven. Verder geeft een Bayesiaans netwerk voor alle variabelen de kansen op de waarden gegeven alle mogelijke waardencombinaties van de ouders. Voor het invullen van de kansen in een netwerk kan bijvoorbeeld gebruik worden gemaakt van schattingen van deskundigen of van bestaande gegevens. Hieronder staat een Bayesiaans netwerkje dat is opgebouwd uit de besproken variabelen $Kleur$ (K) en $Botsing$ (B).


$\Pr(rood) = 0.6$
$\Pr(oranje) = 0.1$
$\Pr(groen) = 0.3$
K
$\Pr(b \mid rood) = 0.009$
$\Pr(b \mid oranje) = 0.003$
$\Pr(b \mid groen) = 0.001$
B

In dit netwerkje is $Kleur$ een ouder van $Botsing$ en is $Botsing$ een kind van $Kleur$. Gegeven een Bayesiaans netwerk kunnen in principe alle kansen van de gerepresenteerde kansverdeling worden uitgerekend. Het netwerkje geeft bijvoorbeeld niet expliciet de kans op een botsing ($\Pr(b)$), maar deze kans kan wel worden berekend. Het netwerkje geeft bijvoorbeeld ook niet expliciet wat de kans is dat iemand door rood gereden is, gegeven dat er een botsing heeft plaatsgevonden ($\Pr(\mathrm{rood} \mid b)$), maar ook deze kans kan met behulp van het netwerk worden bepaald.

Hieronder staat nog een voorbeeld van een Bayesiaans netwerkje. Het netwerkje bevat de

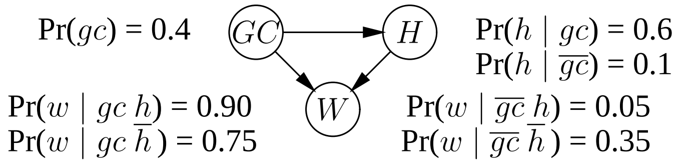


Pr(gc) = 0.4
GC
H
Pr(h) = 0.3
Pr($w \mid gc\ h$) = 0.90
Pr($w \mid gc\ \overline{h}$) = 0.75
W
Pr($w \mid \overline{gc}\ h$) = 0.05
Pr($w \mid \overline{gc}\ \overline{h}$) = 0.35

variabele $Goede\ Conditie$ (GC) die aangeeft of iemand ja of nee een goede conditie heeft, de variabele $Hardlopen$ (H) die aangeeft of iemand ja of nee aan het hardlopen is en de variabele $Welbevinden$ (W) die aangeeft of iemand ja of nee een gevoel van welbevinden ervaart. Volgens het netwerkje is de kans dat iemand een goede conditie heeft gelijk aan 0.4 en dus de kans dat iemand geen goede conditie heeft gelijk aan 0.6. Het netwerkje geeft ook de kansen op de waarden van $Hardlopen$. Voor $Welbevinden$ zijn alleen de kansen, gegeven de waardencombinaties van de ouders, gespecificeerd. Zo is bijvoorbeeld de kans dat iemand een gevoel van welbevinden ervaart, gegeven dat deze persoon een goede conditie heeft en gegeven dat deze persoon aan het hardlopen is ($\Pr(w \mid gc\ h)$), volgens het netwerkje gelijk aan 0.9. Weer geldt dat slechts en deel van de kansverdeling expliciet gegeven is; alle andere kansen kunnen worden uitgerekend.

Een kwalitatief probabilistisch netwerk is een wat grovere variant op een Bayesiaans netwerk. In een kwalitatief probabilistisch netwerk worden de invloeden tussen met elkaar verbonden variabelen met behulp van plussen, minnen en vraagtekens aangegeven. In het tweede voorbeeldnetwerkje kan er een plus worden gezet op de pijl tussen $Goede\ Conditie$ en $Welbevinden$. Een goede conditie heeft hoe dan ook een positieve invloed op het gevoel van welbevinden. De invloed van hardlopen op het gevoel van welbevinden hangt echter af de conditie. Met een goede conditie is harlopen plezierig maar met een slechte conditie niet. Op de pijl tussen de variabelen $Hardlopen$ en $Welbevinden$ komt daarom een vraagteken te staan. Met behulp van een kwalitatief netwerk kan de invloed van het observeren van de waarde van een bepaalde variabele op de kans op de waarde ja van de andere variabelen worden berekend. In het voorbeeldnetwerkje vergroot bijvoorbeeld de observatie dat iemand een goede condititie heeft de kans op de observatie dat deze persoon een gevoel van welbevinden ervaart. De aanwezigheid van vraagtekens in een kwalitatief netwerk heeft tot gevolg dat er voor sommige variabelen geen betekenisvolle uitspraken gedaan kunnen worden over de invloed van een bepaalde observatie. Wanneer je ziet dat iemand hardloopt weet je niet of dat de kans op de observatie van een gevoel van welbevinden verhoogt of verlaagt. In het eerste deel van dit proefschrift wordt beargumenteerd dat er aan een vraagteken in een kwalitatief netwerk een zogenaamd ‘situationeel teken’ kan worden toegevoegd. Ruwweg komt het erop neer dat er gebruik wordt gemaakt van de kans op de waarde ja van de ene ouder om de invloed tussen de andere ouder en het gezamelijke kind te bepalen. In het voorbeeldnetwerkje is de kans dat iemand een goede conditie heeft gelijk aan 0.4 en gegeven die kans is de invloed van $Hardlopen$ op $Welbevinden$ negatief. De observatie dat iemand hardloopt verlaagt dus de kans op de observatie dat iemand een gevoel van welbevinden ervaart en het vraagteken kan worden aangevuld met een minteken. Wanneer observaties beschikbaar komen kan een situationeel teken veranderen. In het proefschrift wordt ook beschreven wanneer je zeker weet dat een situationeel teken gelijk blijft en wanneer je een situationeel teken in een vraagteken zult moeten veranderen.

Zoals gezegd kan met behulp van een Bayesiaans netwerk in principe iedere kans van de gerepresenteerde kansverdeling bepaald worden. Voor sommige netwerken zijn de exacte kansen echter niet meer binnen redelijke tijd te berekenen. Problemen onstaan vooral wanneer een netwerk grote gecompliceerde lussen heeft. Het netwerkje op de vorige bladzijde heeft geen lus. In dit netwerkje wordt ervan uitgegaan dat $Hardlopen$ en $Goede\ Conditie$ geen directe invloed op elkaar uitoefenen. Je zou echter ook kunnen zeggen dat dit niet klopt en dat er nog een pijl tussen deze twee variabelen hoort te staan. In het onderstaande netwerkje is deze pijl wel aanwezig. Nu heeft het netwerkje een lus. In het proefschrift worden variabelen met twee of meer


Pr(gc) = 0.4
GC
H
Pr($h \mid gc$) = 0.6
Pr($h \mid \overline{gc}$) = 0.1
Pr($w \mid gc\ h$) = 0.90
Pr($w \mid gc\ \overline{h}$) = 0.75
W
Pr($w \mid \overline{gc}\ h$) = 0.05
Pr($w \mid \overline{gc}\ \overline{h}$) = 0.35

inkomende pijlen uit dezelfde lus convergentievariabelen genoemd, de andere variabelen in een lus zijn binnenvariabelen. In het laatste netwerkje is $Welbevinden$ dus een convergentievariabele en $Goede\ Conditie$ en $Hardlopen$ zijn binnenvariabelen. Bij netwerken waarvoor exacte kansen niet meer binnen redelijke tijd te berekenen zijn kunnen benaderingsalgoritmen worden gebruikt. Een van deze algoritmen is het zogenaamde geluste propagatie-algoritme. Een kenmerk van dit algoritme is dat er alleen informatie wordt uitgewisseld tussen naast elkaar gelegen variabelen. Het voordeel hiervan is dat ook in complexe netwerken berekeningen binnen redelijke tijd kunnen worden uitgevoerd; het nadeel hiervan is dat er fouten in de berekende kansen kunnen ontstaan. Het proefschrift geeft aan dat er twee soorten fouten kunnen worden gevonden in de kansen die met het geluste propagatie-algoritme berekend worden: convergentiefouten en cirkelfouten. Beide soorten fouten ontstaan in de lussen in het netwerk. Convergentiefouten onstaan in convergentievariabelen doordat het algoritme de informatie die een variabele van zijn ouders ontvangt combineert alsof de ouders onafhankelijk van elkaar zijn terwijl dat niet het geval is. Het proefschrift geeft een formule voor de fout die gevonden wordt in de nietvoorwaardelijke kansen die berekend worden voor de convergentievariabelen. De formule geeft dus bijvoorbeeld de fout in de benadering voor $\Pr(w)$. In de formule wordt gebruik gemaakt van een weegfactor, de zogenaamde ‘kwantitatieve ouderlijke synergie’, een begrip dat in dit proefschrift wordt geïntroduceerd. Deze weegfactor wordt berekend uit de voorwaardelijke kansen die in het netwerk voor de convergentieknoop gespecificeerd zijn. Het ontstaan van cirkelfouten hangt samen met het cirkelen van informatie in een lus. Wanneer een netwerk een enkele lus heeft leidt de cirkelfout óf tot ‘overmoedige’ benaderingen óf tot ‘voorzichtige’ benaderingen voor de binnenknopen. Een overmoedige benadering ligt dichter bij 0 of bij 1 dan de werkelijk kans; een voorzichtige benadering ligt juist dichter bij 0.5. Het proefschrift laat zien dat twee kwalitatieve kenmerken van de lus bepalen of de benaderingen door de cirkelfout richting overmoedig of juist richting voorzichtig worden gestuurd.

Dankwoord

Alleerst wil ik mijn promotor en dagelijks begeleider Linda van der Gaag hartelijk danken. Haar inzet, betrokkenheid en haar vertrouwen in mijn werk waren erg belangrijk bij de voltooiing van dit proefschrift. Verder wil ik Martijn Schrage hartelijk danken voor de implementaties die hij voor mij heeft gemaakt. Zijn snelheid van begrip bij mijn misschien niet altijd even heldere uitleg was erg plezierig. Ook met Silja Renooij en Peter de Waal heb ik van gedachten gewisseld over mijn onderzoek; het was fijn dat dat kon. Peter wil ik bovendien bedanken voor het lezen en het becommentariëren van delen van Hoofdstuk 9. De leden van de leescommissie, Frans Groen, Jan van Leeuwen, Peter Lucas, Simon Parsons en Arno Siebes dank ik hartelijk voor de tijd die zij hebben vrijgemaakt voor de beoordeling van het manuscript. Tot slot gaat mijn dank uit naar familie en vrienden die hebben meegeleefd tijdens mijn promotietraject en die mij zo nu en dan van de broodnodige afleiding hebben voorzien.

Curriculum Vitae

Ik ben op 26 september 1962 te Roermond geboren. In dezelfde plaats volgde ik van 1974 tot 1980 het Atheneum B aan de Rijksscholengemeenschap. Na de middelbare school ging ik studeren aan de Landbouwuniversiteit te Wageningen waar ik, in 1987, de studie Humane Voeding afrondde. Daarna werd mijn carrièrepad wat kronkeliger. Eerst werkte ik als administratief medewerker bij het Centraal Museum te Utrecht en daarna als redacteur van de personeelskrant bij Melkunie Holland. In 1990 vond ik een baan in de richting van mijn studie en werd ik projectmedewerker bij het Adviescentrum Toxicologie van het RIVM te Bilthoven. In de tijd dat ik bij het RIVM werkte volgde ik ook enkele jaren de avondopleiding Grafische Vormgeving aan de Hogeschool voor de Kunsten te Utrecht. Mede dankzij die opleidingsjaren kon ik in 1995 als medewerker presentatie aan de slag bij een klein landschapsarchitectenbureau. Op een gegeven moment wilde ik echter graag weer wat analytischer aan de slag en dat was de aanleiding om in 1998 te beginnen aan de opleiding Cognitieve Kunstmatige Intelligentie aan de Universiteit Utrecht. Die studie voltooide ik (cum laude) in 2002. Daarna volgde van februari 2003 tot februari 2008 een aanstelling als assistent in opleiding bij de groep Beslissings Ondersteunende Systemen van het Departement voor Informatica en Informatiekunde van de Universiteit Utrecht. Bij dezelfde groep vervul ik op dit moment de functie van onderzoeker.

SIKS Dissertation Series

1998-1 Johan van den Akker (CWI), *DEGAS - An Active, Temporal Database of Autonomous Objects*
1998-2 Floris Wiesman (UM), *Information Retrieval by Graphically Browsing Meta-Information*
1998-3 Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*
1998-4 Dennis Breuker (UM), *Memory versus Search in Games*
1998-5 E.W.Oskamp (RUL), *Computerondersteuning bij Straftoemeting*
1999-1 Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*
1999-2 Rob Potharst (EUR), *Classification using decision trees and neural nets*
1999-3 Don Beal (UM), *The Nature of Minimax Search*
1999-4 Jacques Penders (UM), *The practical Art of Moving Physical Objects*
1999-5 Aldo de Moor (KUB), *Empowering Communities:*
A Method for the Legitimate User-Driven Specification of Network Information Systems
1999-6 Niek J.E. Wijngaards (VU), *Re-design of compositional systems*
1999-7 David Spelt (UT), *Verification support for object database design*
1999-8 Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation*
2000-1 Frank Niessink (VU), *Perspectives on Improving Software Maintenance*
2000-2 Koen Holtman (TUE), *Prototyping of CMS Storage Management*
2000-3 Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie;*
een procesbenadering en actorperspectief
2000-4 Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*
2000-5 Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval*
2000-6 Rogier van Eijk (UU), *Programming Languages for Agent Communication*
2000-7 Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*
2000-8 Veerle Coupe (EUR), *Sensitivity Analyis of Decision-Theoretic Networks*
2000-9 Florian Waas (CWI), *Principles ofProbabilistic Query Optimization*
2000-10 Niels Nes (CWI), *Image Database Management System Design Considerations Algorithms and Architecture*
2000-11 Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*
2001-1 Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*
2001-2 Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*
2001-3 Maarten van Someren (UvA), *Learning as problem solving*
2001-4 Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*
2001-5 Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*
2001-6 Martijn van Welie (VU), *Task-based User Interface Design*
2001-7 Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*
2001-8 Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*
2001-9 Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*
2001-10 Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS:*
a multiagent modeling and simulation language forwork practice analysis and design
2001-11 Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*
2002-01 Nico Lassing (VU), *Architecture-Level Modifiability Analysis*
2002-02 Roelof van Zwol (UT), *Modelling and searching web-based document collections*
2002-03 Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*
2002-04 Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*
2002-05 Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*
2002-06 Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*
2002-07 Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*
2002-08 Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*
2002-09 Willem-Jan van den Heuvel(KUB), *Integrating Modern Business Applications with Objectified LegacySystems*
2002-10 Brian Sheppard (UM), *Towards Perfect Play of Scrabble*
2002-11 Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*
2002-12 Albrecht Schmidt (Uva), *Processing XML in Database Systems*
2002-13 Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*
2002-14 Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*
2002-15 Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*
2002-16 Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*
2002-17 Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*
2003-01 Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*
2003-02 Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*
2003-03 Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*
2003-04 Milan Petkovic (UT), *Content-Based Video Retrieval Supported by Database Technology*
2003-05 Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law - A modelling approach*

2003-06 Boris van Schooten (UT), *Development and specification of virtual environments*
2003-07 Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*
2003-08 Yongping Ran (UM), *Repair Based Scheduling*
2003-09 Rens Kortmann (UM), *The resolution of visually guided behaviour*
2003-10 Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*
2003-11 Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*
2003-12 Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*
2003-13 Jeroen Donkers (UM), *Nosce Hostem - Searching with Opponent Models*
2003-14 Stijn Hoppenbrouwers (KUN), *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*
2003-15 Mathijs de Weerdt (TUD), *Plan Merging in Multi-Agent Systems*
2003-16 Menzo Windhouwer (CWI), *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*
2003-17 David Jansen (UT), *Extensions of Statecharts with Probability, Time, and Stochastic Timing*
2003-18 Levente Kocsis (UM), *Learning Search Decisions*
2004-01 Virginia Dignum (UU), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*
2004-02 Lai Xu (UvT), *Monitoring Multi-party Contracts for E-business*
2004-03 Perry Groot (VU), *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*
2004-04 Chris van Aart (UVA), *Organizational Principles for Multi-Agent Architectures*
2004-05 Viara Popova (EUR), *Knowledge discovery and monotonicity*
2004-06 Bart-Jan Hommes (TUD), *The Evaluation of Business Process Modeling Techniques*
2004-07 Elise Boltjes (UM), *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*
2004-08 Joop Verbeek(UM), *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*
2004-09 Martin Caminada (VU), *For the Sake of the Argument; explorations into argument-based reasoning*
2004-10 Suzanne Kabel (UVA), *Knowledge-rich indexing of learning-objects*
2004-11 Michel Klein (VU), *Change Management for Distributed Ontologies*
2004-12 The Duy Bui (UT), *Creating emotions and facial expressions for embodied agents*
2004-13 Wojciech Jamroga (UT), *Using Multiple Models of Reality: On Agents who Know how to Play*
2004-14 Paul Harrenstein (UU), *Logic in Conflict. Logical Explorations in Strategic Equilibrium*
2004-15 Arno Knobbe (UU), *Multi-Relational Data Mining*
2004-16 Federico Divina (VU), *Hybrid Genetic Relational Search for Inductive Learning*
2004-17 Mark Winands (UM), *Informed Search in Complex Games*
2004-18 Vania Bessa Machado (UvA), *Supporting the Construction of Qualitative Knowledge Models*
2004-19 Thijs Westerveld (UT), *Using generative probabilistic models for multimedia retrieval*
2004-20 Madelon Evers (Nyenrode), *Learning from Design: facilitating multidisciplinary design teams*
2005-01 Floor Verdenius (UVA), *Methodological Aspects of Designing Induction-Based Applications*
2005-02 Erik van der Werf (UM), *AI techniques for the game of Go*
2005-03 Franc Grootjen (RUN), *A Pragmatic Approach to the Conceptualisation of Language*
2005-04 Nirvana Meratnia (UT), *Towards Database Support for Moving Object data*
2005-05 Gabriel Infante-Lopez (UVA), *Two-Level Probabilistic Grammars for Natural Language Parsing*
2005-06 Pieter Spronck (UM), *Adaptive Game AI*
2005-07 Flavius Frasincar (TUE), *Hypermedia Presentation Generation for Semantic Web Information Systems*
2005-08 Richard Vdovjak (TUE), *A Model-driven Approach for Building Distributed Ontology-based Web Applications*
2005-09 Jeen Broekstra (VU), *Storage, Querying and Inferencing for Semantic Web Languages*
2005-10 Anders Bouwer (UVA), *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
2005-11 Elth Ogston (VU), *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*
2005-12 Csaba Boer (EUR), *Distributed Simulation in Industry*
2005-13 Fred Hamburg (UL), *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*
2005-14 Borys Omelayenko (VU), *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*
2005-15 Tibor Bosse (VU), *Analysis of the Dynamics of Cognitive Processes*
2005-16 Joris Graaumans (UU), *Usability of XML Query Languages*
2005-17 Boris Shishkov (TUD), *Software Specification Based on Re-usable Business Components*
2005-18 Danielle Sent (UU), *Test-selection strategies for probabilistic networks*
2005-19 Michel van Dartel (UM), *Situated Representation*
2005-20 Cristina Coteanu (UL), *Cyber Consumer Law, State of the Art and Perspectives*
2005-21 Wijnand Derks (UT), *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*
2006-01 Samuil Angelov (TUE), *Foundations of B2B Electronic Contracting*
2006-02 Cristina Chisalita (VU), *Contextual issues in the design and use of information technology in organizations*
2006-03 Noor Christoph (UVA), *The role of metacognitive skills in learning to solve problems*
2006-04 Marta Sabou (VU), *Building Web Service Ontologies*
2006-05 Cees Pierik (UU), *Validation Techniques for Object-Oriented Proof Outlines*
2006-06 Ziv Baida (VU), *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*
2006-07 Marko Smiljanic (UT), *XML schema matching – balancing efficiency and effectiveness by means of clustering*
2006-08 Eelco Herder (UT), *Forward, Back and Home Again - Analyzing User Behavior on the Web*
2006-09 Mohamed Wahdan (UM), *Automatic Formulation of the Auditor’s Opinion*
2006-10 Ronny Siebes (VU), *Semantic Routing in Peer-to-Peer Systems*

2006-11 Joeri van Ruth (UT), *Flattening Queries over Nested Data Types*
2006-12 Bert Bongers (VU), *Interactivation - Towards an e-cology of people, our technological environment, and the arts*
2006-13 Henk-Jan Lebbink (UU), *Dialogue and Decision Games for Information Exchanging Agents*
2006-14 Johan Hoorn (VU), *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*
2006-15 Rainer Malik (UU), *CONAN: Text Mining in the Biomedical Domain*
2006-16 Carsten Riggelsen (UU), *Approximation Methods for Efficient Learning of Bayesian Networks*
2006-17 Stacey Nagata (UU), *User Assistance for Multitasking with Interruptions on a Mobile Device*
2006-18 Valentin Zhizhkun (UVA), *Graph transformation for Natural Language Processing*
2006-19 Birna van Riemsdijk (UU), *Cognitive Agent Programming: A Semantic Approach*
2006-20 Marina Velikova (UvT), *Monotone models for prediction in data mining*
2006-21 Bas van Gils (RUN), *Aptness on the Web*
2006-22 Paul de Vrieze (RUN), *Fundaments of Adaptive Personalisation*
2006-23 Ion Juvina (UU), *Development of Cognitive Model for Navigating on the Web*
2006-24 Laura Hollink (VU), *Semantic Annotation for Retrieval of Visual Resources*
2006-25 Madalina Drugan (UU), *Conditional log-likelihood MDL and Evolutionary MCMC*
2006-26 Vojkan Mihajlovic' (UT), *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
2006-27 Stefano Bocconi (CWI), *Vox Populi: generating video documentaries from semantically annotated media repositories*
2006-28 Borkur Sigurbjornsson (UVA), *Focused Information Access using XML Element Retrieval*
2007-01 Kees Leune (UvT), *Access Control and Service-Oriented Architectures*
2007-02 Wouter Teepe (RUG), *Reconciling Information Exchange and Confidentiality: A Formal Approach*
2007-03 Peter Mika (VU), *Social Networks and the Semantic Web*
2007-04 Jurriaan van Diggelen (UU), *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*
2007-05 Bart Schermer (UL), *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*
2007-06 Gilad Mishne (UVA), *Applied Text Analytics for Blogs*
2007-07 Natasa Jovanovic' (UT), *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*
2007-08 Mark Hoogendoorn (VU), *Modeling of Change in Multi-Agent Organizations*
2007-09 David Mobach (VU), *Agent-Based Mediated Service Negotiation*
2007-10 Huib Aldewereld (UU), *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*
2007-11 Natalia Stash (TUE), *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*
2007-12 Marcel van Gerven (RUN), *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
2007-13 Rutger Rienks (UT), *Meetings in Smart Environments; Implications of Progressing Technology*
2007-14 Niek Bergboer (UM), *Context-Based Image Analysis*
2007-15 Joyca Lacroix (UM), *NIM: a situated computational memory model*
2007-16 Davide Grossi (UU), *Designing Invisible Handcuffs: Formal Investigations in Institutions and Organizations for Multi-agent Systems*
2007-17 Theodore Charitos (UU), *Reasoning with Dynamic Networks in Practice*
2007-18 Bart Orriens (UvT), *On the development an management of adaptive business collaborations*
2007-19 David Levy (UM), *Intimate relationships with artificial partners*
2007-20 Slinger Jansen (UU), *Customer Configuration Updating in a Software Supply Network*
2007-21 Karianne Vermaas (UU), *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*
2007-22 Zlatko Zlatev (UT), *Goal-oriented design of value and process models from patterns*
2007-23 Peter Barna (TUE), *Specification of Application Logic in Web Information Systems*
2007-24 Georgina Ramrez Camps (CWI), *Structural Features in XML Retrieval*
2007-25 Joost Schalken (VU), *Empirical Investigations in Software Process Improvement*
2008-01 Katalin Boer-Sorbn (EUR), *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*
2008-02 Alexei Sharpanskykh (VU), *On Computer-Aided Methods for Modeling and Analysis of Organizations*
2008-03 Vera Hollink (UVA), *Optimizing hierarchical menus: a usage-based approach*
2008-04 Ander de Keijzer (UT), *Management of Uncertain Data - towards unattended integration*
2008-05 Bela Mutschler (UT), *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*
2008-06 Arjen Hommersom (RUN), *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*
2008-07 Peter van Rosmalen (OU), *Supporting the tutor in the design and support of adaptive e-learning*
2008-08 Janneke Bolt (UU), *Bayesian Networks: Aspects of Approximate Inference*
2008-09 Christof van Nimwegen (UU), *The paradox of the guided user: assistance can be counter-effective*