Decisiveness in Loopy Propagation

Janneke H. Bolt

Linda C. van der Gaag

Department of Information and Computing Sciences, Utrecht University

Technical Report UU-CS-2006-005

www.cs.uu.nl

ISSN: 0924-3275

Decisiveness in Loopy Propagation

Janneke H. Bolt and Linda C. van der Gaag

Department of Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands

Abstract. 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. We identify the two types of error that can arise in the probabilities yielded by the algorithm: the cycling error and the convergence error. We then focus on the cycling error and analyse its effect on the decisiveness of the approximations that are computed for the inner nodes of simple loops. More specifically, we detail the factors that induce the cycling error to push the exact probabilities towards over- or underconfident approximations.

1 Introduction

Bayesian networks [1] by now are being applied for a range of problems in a variety of domains. Successful applications are being realised for example for medical diagnosis, for traffic prediction, for technical troubleshooting and for information retrieval. In these applications, probabilistic inference plays an important role. Probabilistic inference with a Bayesian network amounts to computing (posterior) probability distributions for the variables involved. For networks without any topological restrictions, inference is known to be NP-hard [2]. For various classes of networks of restricted topology, however, efficient algorithms are available, such as Pearl's propagation algorithm for singly connected networks. The availability of these algorithms accounts to a large extent for the success of current Bayesian-network applications.

For Bayesian 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 [3]. 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 to a Bayesian network regardless of its topological structure. From an experimental point of view, Murphy et al. [4] reported good approximation behaviour of the loopy-propagation algorithm used on Bayesian networks whenever there was rapid convergence. Excellent performance has also been reported for algorithms equivalent to the loopy-propagation algorithm [5–7].

Several researchers have analysed the approximation behaviour of the loopy-propagation algorithm from a more fundamental point of view. Weiss and Freeman, more specifically, studied the performance of an equivalent algorithm on Markov networks [8,9]; their use of Markov networks was motivated by the relatively easier analysis of these networks and justified by the observation that any Bayesian network can be converted into a pairwise Markov network. For pairwise Markov networks with a single loop, Weiss in fact derived an analytical relationship between the exact probabilities and the approximate probabilities computed for the nodes in the loop [8].

In this paper we study the performance properties of the loopy-propagation algorithm on Bayesian networks directly, and thereby provide further insights in the errors that it generates. We argue that two different types of error are introduced in the computed approximate probabilities, which we term 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 emerges in a convergence node only, that is, in a node with two or more incoming arcs on the loop under study. A cycling error arises when messages are being passed on within a loop repetitively and old information is mistaken for new by the variables involved.

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 in all nodes of the loop.

Weiss notes that the approximate probabilities found upon loopy propagation are overconfident as a result of double counting of evidence [8]. We observe, however, that overconfident as well as underconfident approximations can result. We use the term *decisiveness* to refer to the over- or underconfidence of an approximation. Decisiveness is an important concept as knowledge of the over- or underconfidence of an approximate probability provides an indication of where the exact probability lies. In this paper, we study the effect of the cycling error on the decisiveness of the approximations found for the inner nodes of a simple loop in a binary Bayesian network. We show that the effect depends on the qualitative influence between the parents of the loop's convergence node and the additional intercausal influence that is induced between these parents by the entered evidence. If the two influences have equal signs, the cycling error pushes the exact probabilities to overconfident approximations; otherwise, the approximations are pushed towards underconfidence.

The paper is organised as follows. In Sect. 2, we provide some preliminaries on Bayesian networks and on Pearl's propagation algorithm. In Sect. 3, we describe the two types of error that may be introduced by loopy propagation. In Sect. 4, we derive the relationship between the exact and approximate probabilities for the inner loop nodes and in Sect. 5 we investigate the decisiveness of the approximations. The paper is rounded off with some conclusions and directions for further research in Sect. 6.

2 Preliminaries

In Sect. 2.1 we provide some preliminaries on Bayesian networks; in Sect. 2.2 we review Pearl's algorithm for probabilistic inference with singly connected networks.

2.1 Bayesian Networks

A *Bayesian network* is a model of a joint probability distribution Pr over a set of stochastic variables, consisting of a directed acyclic graph and a set of conditional probability distributions. In this paper we assume all variables of the network to be binary, taking one of the values *true* and *false*. We will write a for $A = \textit{true}$ and $\bar{a}$ for $A = \textit{false}$. We use a_i to denote any value assignment to A, that is $a_i \in \{a, \bar{a}\}$. Each variable is represented by a node in the network's digraph; from now on, we will use the terms node and variable interchangeably. The probabilistic relationships between the variables are captured by the digraph's set of arcs according to the d-separation criterion [1]. Associated with the graphical structure are numerical quantities from the modelled distribution: for each variable A, conditional probability distributions $\Pr(A \mid p(A))$ are specified, where $p(A)$ denotes the set of parents of A in the digraph. Fig. 1 depicts a small example Bayesian network.

In the sequel, we distinguish between singly connected and multiply connected Bayesian networks. A network is *singly connected* if there is at most one trail between any two variables in its digraph. If there are multiple trails between variables, then the network is *multiply connected*. A multiply connected network includes one or more loops, that is, one or more cycles in its underlying undirected graph. We say that a loop is simple if none of its nodes are shared by another loop. A node that has two or more incoming arcs on a loop will be called a *convergence node* of this loop; the other nodes of the loop will be termed *inner nodes*. The network from Fig. 1 is an example of a multiply connected network. The trail $A \rightarrow B \rightarrow C \leftarrow A$ constitutes a simple loop in the network's digraph. Node C is the only convergence node of this loop; nodes A and B are the loop's inner nodes.

2.2 Pearl's Propagation Algorithm

We briefly review Pearl's propagation algorithm [1]. This algorithm was designed for exact inference with singly connected Bayesian networks. The term *loopy propagation* used throughout the literature, refers to the application of this algorithm to networks with loops. In the algorithm, each node X is provided with a limited set of rules that enable the node to calculate its probability distribution $\Pr(X \mid \mathbf{e})$ given the available evidence $\mathbf{e}$, from messages it receives from its neighbours. These rules are applied in parallel by the various nodes at each time step. The rule used by node X for establishing the probability distribution $\Pr(X \mid \mathbf{e})$ at time t is

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

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:

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

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:

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

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

$$\lambda^{t+1}_X(U^i) = cst \cdot \sum_X \lambda^t(X) \cdot \sum_{\mathbf{U}/U^i} \Pr(X \mid \mathbf{U}) \cdot \prod_{U^k,\, k \neq i} \pi^t_X(U^k)$$

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

$$\pi^{t+1}_{Y^j}(X) = cst \cdot \pi^t(X) \cdot \prod_{Y^k,\, k \neq j} \lambda^t_{Y^k}(X)$$

where *cst* denotes a normalisation constant. Note that, in general, the number of messages that a node sends in each time step to a child equals the number of its own values; the number of messages that it sends to a parent equals the number of values of this parent. From here on, we will denote a diagnostic parameter $\lambda_X(U^i)$ and a causal parameter $\pi_{Y^j}(X)$ by the term *message vector*. For a binary parent U^i, we write the diagnostic messages as $(\lambda_X(u^i), \lambda_X(\bar{u}^i))$; for a binary child X, we write the causal messages as $(\pi_{Y^j}(x), \pi_{Y^j}(\bar{x}))$. All message vectors are initialised to contain just 1s. An observation for a node X now is entered into the network by multiplying the components of $\lambda^0(X)$ and $\pi^1_{Y^j}(X)$ by 1 for the observed value of X and by 0 for the other value(s).

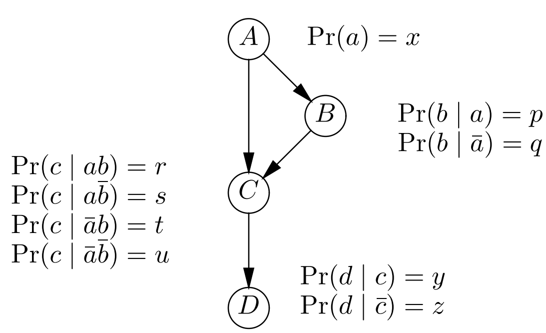

Fig. 1. A multiply connected Bayesian network with a convergence node C having the dependent parents A and B and the child D

3 Errors in Loopy Propagation

Pearl's propagation algorithm results in exact probabilities whenever it is applied to a singly connected Bayesian network. After a finite number of time steps, proportional to the diameter of the network's digraph, the probabilistic information present in the network will have been passed on to all nodes. The network then reaches an equilibrium state in which the computed probabilities and messages do no longer change upon further message passing. When applied to a multiply connected Bayesian network, that is, upon performing loopy propagation, the algorithm will often converge as well; we will consider the algorithm to have converged as soon as all causal and diagnostic messages and all computed probabilities change by less than a prespecified threshold value in the next time step. The resulting probabilities may then deviate from the probabilities of the modelled distribution, however.

The probabilities that result from performing loopy propagation on a multiply connected network, may include two different types of error. The first type of error originates at the convergence node(s) of a loop. In Pearl's algorithm, a node with two or more parents combines the messages from its parents as if these messages come from independent variables. In a singly connected network the parents of a node indeed are always independent. The parents of a convergence node, however, may be dependent. By assuming independence upon combining the causal messages from dependent variables, an error is introduced. A convergence error may be propagated to neighbours outside the loop. Given compound loops, a convergence error may enter the loop; for simple loops, however, this error does not affect the probabilities computed for the inner nodes of the loop. In our previous work [10], we studied the convergence error in detail; in this paper, we focus on the effect of the second type of error, called the *cycling error*. This type of error arises when messages are being passed on repetitively in the loop, where old information is mistaken for new by the nodes involved. This cycling of information can occur as soon as for each convergence node of the loop either the node itself or one of its descendants is observed. A cycling error arises in all loop nodes and is propagated to nodes outside the loop. Note that at the convergence node both types of error emerge upon loopy propagation, whereas at the inner nodes of simple loops only the cycling error originates. In the sequel we will denote the probabilities that result upon loopy propagation with $\widetilde{\Pr}$ to distinguish them from the exact probabilities which are denoted by $\Pr$.

4 The Relationship Between the Exact and Approximate Probabilities

Upon performing loopy propagation on a Bayesian network with a simple loop, a cycling error may arise in the probabilities computed for all the nodes of the loop, as argued in the previous section. In this section we will derive, for the network from Fig. 1, an expression that relates the exact probabilities for the inner loop nodes to the computed approximate probabilities. Our derivation is analogous to the one constructed by Weiss [8] for an equivalent algorithm applied to binary Markov networks with a single loop. From the relationship between the exact and approximate probabilities, we then identify the factors, in terms of a network's specification, that determine whether the exact probabilities for the inner loop nodes are pushed towards overconfident or underconfident approximations. We will study these factors in Sect. 5.

We consider the Bayesian network from Fig. 1 and suppose that the evidence d has been entered. We now build upon the observation that the updating of a message vector during propagation can be captured by a transition matrix. We begin by deriving the matrices that describe the information that is included into a message vector during one clockwise cycle and during one counterclockwise cycle respectively, from node A back to itself. We will then use the eigenvalues of these matrices to express the relationship between the exact and approximate probabilities found at node A.

To derive the transition matrix that captures the information that is added during one clockwise cycle from node A back to itself, we consider the updating of the message vector $(1, 1)$ during the first cycle of the algorithm; we recall that node A initially receives this vector. In the first step of the algorithm, node A sends the vector

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

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 is

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

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 to node A:

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

After the first clockwise cycle of the algorithm, therefore, the initial message vector $(1,1)$ has been updated to the vector given above. We now derive the transition matrix

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

that captures this update. We observe that the entries l, m, n and o of this matrix should adhere to

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

from which we find that

$$l + m = \lambda_C(a)$$
$$n + o = \lambda_C(\bar{a})$$

We now split the expression for $\lambda_C(a)$ into separate terms for l and m and the expression for $\lambda_C(\bar{a})$ into separate terms for n and o. To this end, we observe that in the analysis above, the first component of the message vector from node A to node B pertains to a and the second component pertains to $\bar{a}$. Roughly speaking, now, since the first component is multiplied by l and n, these entries have to collect all information at node B concerning a. As $p = \Pr(b \mid a)$ 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. After rearranging the various terms in the expressions for $\lambda_C(a)$ and $\lambda_C(\bar{a})$ accordingly, we find that

$$\begin{aligned} l &= [(y \cdot r + z \cdot (1-r)) \cdot p + (y \cdot s + z \cdot (1-s)) \cdot (1-p)] \cdot x \\ m &= [(y \cdot r + z \cdot (1-r)) \cdot q + (y \cdot s + z \cdot (1-s)) \cdot (1-q)] \cdot (1-x) \\ n &= [(y \cdot t + z \cdot (1-t)) \cdot p + (y \cdot u + z \cdot (1-u)) \cdot (1-p)] \cdot x \\ o &= [(y \cdot t + z \cdot (1-t)) \cdot q + (y \cdot u + z \cdot (1-u)) \cdot (1-q)] \cdot (1-x) \end{aligned}$$

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

$$M^{\circlearrowleft A,d} = \begin{bmatrix} l & n \cdot \frac{1-x}{x} \\ m \cdot \frac{x}{1-x} & o \end{bmatrix}$$

Upon loopy propagation, the information captured by the above two transition matrices is included repeatedly in every cycle. We would like to observe that all message vectors are normalised, as described in Sect. 2.2. As a consequence, repeated multiplication by the transition matrices will not result in convergence to $(0, 0)$.

Example We consider the example Bayesian network from Fig. 2 and address the approximate probabilities found for the inner node A upon performing loopy propagation. We consider the probabilities after the evidence d and $\bar{d}$ have been entered, respectively, during the first five cycles of the algorithm. The subsequent approximations are shown in Figs. 3 and 4; for comparison, the exact probabilities are depicted as well. We observe that the approximate probabilities asymptotically approach a particular value. The approximate probabilities given d oscillate around the final value, whereas given $\bar{d}$ the approximations go steadily towards the final value. In Sect. 5, we will return to this difference in approximation behaviour. □

The example demonstrates that upon loopy propagation the computed probabilities converge towards an equilibrium value. We now will exploit the eigenvalues of the transition matrices to relate the approximate probabilities in the equilibrium state to the exact probabilities. The eigenvalues λ_1 and λ_2 of the matrix $M^{\circlearrowright A,d}$ are the solutions of

$$\lambda = \frac{1}{2} \cdot [(l + o) \pm \sqrt{(l + o)^2 - 4 \cdot (l \cdot o - m \cdot n)}]$$

where λ_1 is the largest of the two values. For $M^{\circlearrowleft A,d}$ the same eigenvalues are found. Note that since the entries of the two matrices are positive, the eigenvalues λ_1 and λ_2 are real numbers and both λ_1 and $\lambda_1 + \lambda_2$ are positive.

The relationship between the exact and the approximate probability of a_i given d now is expressed by

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

To prove this property, we begin by observing that for each eigenvalue of $M^{\circlearrowright A,d}$ and for each eigenvalue of $M^{\circlearrowleft A,d}$, an eigenvector direction is found. For $M^{\circlearrowright A,d}$, we denote the normalised principal eigenvector by (α_1, β_1) and we denote a fixed arbitrary vector in the second eigenvector direction by (γ_1, δ_1); for $M^{\circlearrowleft A,d}$ we denote the normalised principal eigenvector by (α_2, β_2). In

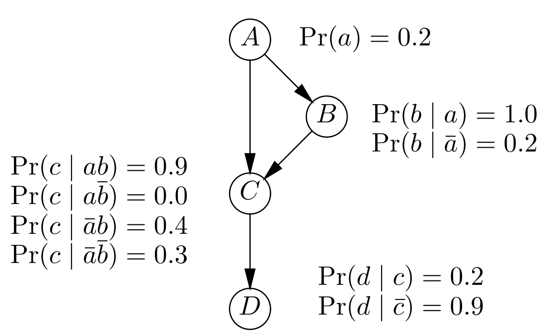

Fig. 2. An example Bayesian network

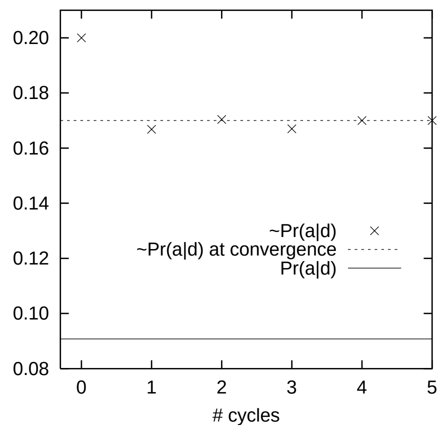

Fig. 3. The approximate probabilities given d found for node A in the network of Fig. 2 during the first five cycles of the loopy propagation algorithm

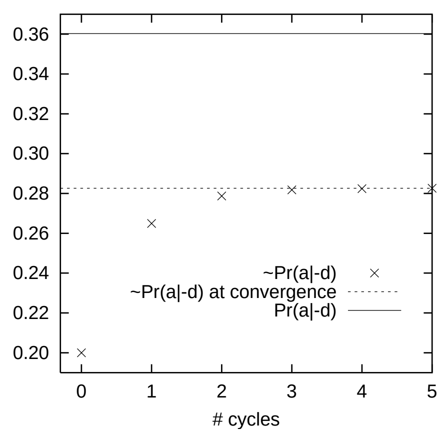

Fig. 4. The approximate probabilities given $\bar{d}$ found for node A in the network of Fig. 2 during the first five cycles of the loopy propagation algorithm

the equilibrium state, now, node A receives the message $\lambda_C(A) = (\alpha_1, \beta_1)$ from node C and the message $\lambda_B(A) = (\alpha_2, \beta_2)$ from node B, cf. [11]. These two messages are combined by node A with its own knowledge $\pi(A) = (x, 1-x)$ of the prior distribution over its values, which results in the approximate probabilities

$$\widetilde{\Pr}(A \mid d) = cst_1 \cdot \begin{bmatrix} \alpha_1 \cdot \alpha_2 \cdot x \\ \beta_1 \cdot \beta_2 \cdot (1-x) \end{bmatrix}$$

where cst_1 is a normalisation constant.

We now relate the computed approximate probability $\widetilde{\Pr}(a_i \mid d)$ to the exact probability $\Pr(a_i \mid d)$. We begin by observing that for the entries l and o of the transition matrices $M^{\circlearrowleft A,d}$ and $M^{\circlearrowright A,d}$, we have that $l = \Pr(d \mid a) \cdot \Pr(a)$ and $o = \Pr(d \mid \bar{a}) \cdot \Pr(\bar{a})$. For the exact probabilities $\Pr(a_i \mid d)$ we thus find 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, we now relate the entries l and o to the expressions $\alpha_1 \cdot \alpha_2 \cdot x$ and $\beta_1 \cdot \beta_2 \cdot (1-x)$. To this end, we diagonalise the matrix $M^{\circlearrowleft A,d}$ into

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

where $\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} = \begin{bmatrix} \alpha_1 & \gamma_1 \\ \beta_1 & \delta_1 \end{bmatrix}^{-1}$. We thus have that

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

From $\begin{bmatrix} \alpha_1 & \gamma_1 \\ \beta_1 & \delta_1 \end{bmatrix} \cdot \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ we find 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

$$l = \alpha_1 \cdot \mathcal{A} \cdot \lambda_1 + \beta_1 \cdot \mathcal{B} \cdot \lambda_2$$
$$o = \beta_1 \cdot \mathcal{B} \cdot \lambda_1 + \alpha_1 \cdot \mathcal{A} \cdot \lambda_2$$

To express $\mathcal{A}$ and $\mathcal{B}$ in terms of α_2, β_2 and x, we now rewrite the matrix $M^{\circlearrowleft A,d}$ as

$$M^{\circlearrowleft A,d} = \begin{bmatrix} \frac{1}{x} & 0 \\ 0 & \frac{1}{1-x} \end{bmatrix} \cdot (M^{\circlearrowleft A,d})^T \cdot \begin{bmatrix} x & 0 \\ 0 & 1-x \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{x} & 0 \\ 0 & \frac{1}{1-x} \end{bmatrix} \cdot \begin{bmatrix} \mathcal{A} & \mathcal{C} \\ \mathcal{B} & \mathcal{D} \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 & \beta_1 \\ \gamma_1 & \delta_1 \end{bmatrix} \cdot \begin{bmatrix} x & 0 \\ 0 & 1-x \end{bmatrix}$$

The first column of the product

$$\begin{bmatrix} \frac{1}{x} & 0 \\ 0 & \frac{1}{1-x} \end{bmatrix} \cdot \begin{bmatrix} \mathcal{A} & \mathcal{C} \\ \mathcal{B} & \mathcal{D} \end{bmatrix} = \begin{bmatrix} \frac{1}{x} \cdot \mathcal{A} & \frac{1}{x} \cdot \mathcal{C} \\ \frac{1}{1-x} \cdot \mathcal{B} & \frac{1}{1-x} \cdot \mathcal{D} \end{bmatrix}$$

now is a vector in the direction of the principal eigenvector (α_2, β_2) of $M^{\circlearrowleft A,d}$. We thus find that $\mathcal{A} = cst_2 \cdot \alpha_2 \cdot x$ and $\mathcal{B} = cst_2 \cdot \beta_2 \cdot (1-x)$, where $1/cst_2$ is a normalisation constant. We conclude that

$$l = cst_2 \cdot (\alpha_1 \cdot \alpha_2 \cdot x \cdot \lambda_1 + \beta_1 \cdot \beta_2 \cdot (1-x) \cdot \lambda_2)$$
$$= cst_2/cst_1 \cdot (\widetilde{\Pr}(a \mid d) \cdot \lambda_1 + \widetilde{\Pr}(\bar{a} \mid d) \cdot \lambda_2)$$
$$o = cst_2 \cdot (\beta_1 \cdot \beta_2 \cdot (1-x) \cdot \lambda_1 + \alpha_1 \cdot \alpha_2 \cdot x \cdot \lambda_2)$$
$$= cst_2/cst_1 \cdot (\widetilde{\Pr}(\bar{a} \mid d) \cdot \lambda_1 + \widetilde{\Pr}(a \mid d) \cdot \lambda_2)$$

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

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

For $\bar{d}$, the derivation is analogous. For node B similar expressions are found. We would like to note that the transition matrices for nodes A and B have the same eigenvalues. 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. Eventually, the clockwise matrices for nodes A and B result from the multiplication of the same two 2 by 2 matrices, yet in different order. The same applies to the counterclockwise matrices. As a consequence, the transition matrices per cycle for nodes A and B have the same eigenvalues.

Example We consider again the Bayesian network from Fig. 2. Upon performing loopy propagation on this network, a single clockwise cycle serves to include the information

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

into the message vector of node A back to itself. The eigenvalues of this matrix are $\lambda_1 \approx 0.6662$ and $\lambda_2 \approx -0.0714$. Its normalised principal eigenvector is

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

A single counterclockwise cycle serves to include the information

$$M^{\circlearrowleft A,d} = \begin{bmatrix} 0.0540 & 0.4960 \\ 0.1548 & 0.5408 \end{bmatrix}$$

into the message vector of node A back to itself. The eigenvalues of this matrix are again $\lambda_1 \approx 0.6662$ and $\lambda_2 \approx -0.0714$. Its normalised the principal eigenvector is

$$\begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} \approx \begin{bmatrix} 0.4476 \\ 0.5524 \end{bmatrix}$$

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

$$cst_1 \cdot \begin{bmatrix} 0.5028 \cdot 0.4476 \cdot 0.2 \\ 0.4972 \cdot 0.5524 \cdot 0.8 \end{bmatrix} \approx \begin{bmatrix} 0.17 \\ 0.83 \end{bmatrix}$$

We thus find that $\widetilde{\Pr}(a \mid d) \approx 0.17$ and $\widetilde{\Pr}(\bar{a} \mid d) \approx 0.83$. We recall that the exact probability $\Pr(a \mid d)$ equals $l/(l+o)$ and can be read from the diagonal of the transition matrices to be $0.0540/0.5948 \approx 0.09$. For the relationship between $\Pr(a \mid d)$ and $\widetilde{\Pr}(a \mid d)$ we indeed observe that $\Pr(a \mid d) = \widetilde{\Pr}(a \mid d) - \lambda_2/(\lambda_1 + \lambda_2) \cdot (2 \cdot \widetilde{\Pr}(a \mid d) - 1)$ holds as $0.09 \approx 0.17 + 0.0714/0.5948 \cdot (2 \cdot 0.17 - 1)$. Note that the error in $\widetilde{\Pr}(a \mid d)$ computed from the network equals $0.09 - 0.17 = -0.06$ and hence is small. Moreover, given a threshold value of 0.0001, the algorithm had converged within just five cycles. □

5 The Decisiveness of the Approximations

In the previous section we studied, for a Bayesian network with a simple loop, the cycling error that may arise in the probabilities computed upon loopy propagation for the inner loop nodes. More specifically, we derived an expression that relates the exact probabilities for the inner loop nodes to the computed approximate probabilities. We now build upon this analysis to state some properties of the approximations in terms of the specification of the network. We say that an approximation 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. We use the term *decisiveness* to refer to the over- or underconfidence of an approximation. As an example, Fig. 5 depicts, for the network from Fig. 2, the exact and approximate probabilities $\Pr(a \mid d)$ and $\widetilde{\Pr}(a \mid d)$ as a function of $\Pr(a)$; Fig. 6 depicts $\Pr(a \mid \bar{d})$ and $\widetilde{\Pr}(a \mid \bar{d})$. We observe that the evidence d results in underconfident approximations, while the evidence $\bar{d}$ gives overconfident approximations for all possible values of $\Pr(a)$. We 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 show 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.

We begin by introducing the two types of influence that we will exploit in the sequel. A *qualitative influence* [12] between two neighbouring nodes expresses how the values of the one

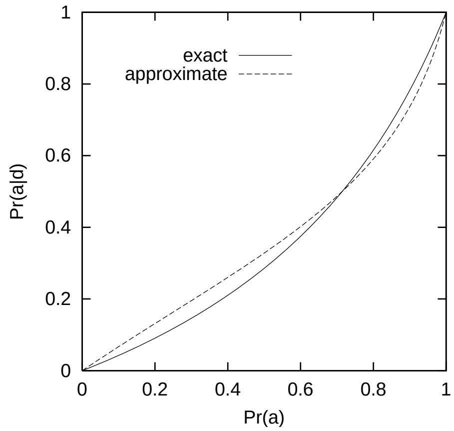

Fig. 5. $\Pr(a \mid d)$ and $\widetilde{\Pr}(a \mid d)$ as a function of $\Pr(a)$ for the network from Fig. 2

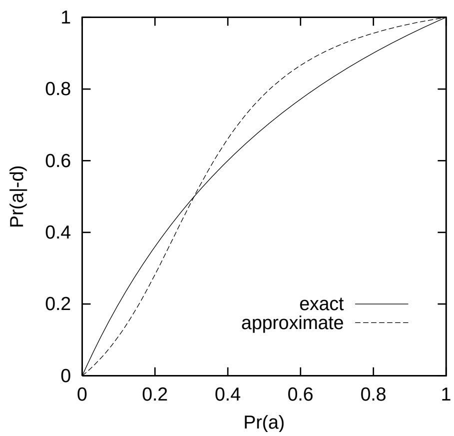

Fig. 6. $\Pr(a \mid \bar{d})$ and $\widetilde{\Pr}(a \mid \bar{d})$ as a function of $\Pr(a)$ for the network from Fig. 2

node influence the probabilities of the values of the other node along their common arc. A positive qualitative influence of a parent B on its child C is found if

$$\Pr(c \mid bx) - \Pr(c \mid \bar{b}x) \geq 0$$

for any combination of values x for the set $p(C) \setminus \{B\}$ of parents of C other than B. In Fig. 1, for example, a positive influence of B on C is found if both $r - s \geq 0$ and $t - u \geq 0$. A negative and a zero qualitative influence are defined analogously. When no consistent sign can be found for the influence given the different combinations of values x, we say that the influence is ambiguous. Positive, negative, zero and ambiguous qualitative influences are indicated by the signs $+$, $-$, 0 and $?$, respectively. Qualitative influences are symmetric, that is, a positive qualitative influence of B on C implies a positive influence of C on B. Qualitative influences further adhere to the property of transitivity, that is, influences along a trail with at most one incoming arc for each variable combine into a net influence whose sign is defined by the $\otimes$-operator from Table 1.

Intercausal influences [13] are dynamic in nature and can only arise after evidence has been entered into the network. In the prior state of a network, that is, when no evidence has been entered as yet, the parents of a node are d-separated from one another along the trail that includes their common child. As soon as evidence is entered for this child or for one of its descendants, however, the two parents may become dependent along this trail. The influence that is thus induced, is

Table 1. The ⊗- operator for combining signs

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

termed an intercausal influence. For example, for a node C with the independent parents A and B we find a positive intercausal influence of node B on node A with respect to c, if

$$\Pr(a \mid bc) - \Pr(a \mid \bar{b}c) \geq 0$$

Intercausal influences, like qualitative influences, adhere to the properties of symmetry and transitivity. We informally review the effect of the intercausal influence. For a node C with the parents A and B, entering evidence will influence the probability distribution for node A along the trail $A \to C$ and the probability distribution for node B along the trail $B \to C$. The influence of node C on the one parent now typically may change with a subsequent change of the probability distribution for the other parent. A positive intercausal influence between nodes A and B with respect to c implies that, given the evidence c, an increase in the probability of b will result in an increase, in terms of positivity, of the influence of node C on node A along the trail $A \to C$. 'An increase in positivity' will say that a negative influence becomes weaker and a positive influence becomes stronger. Given a positive intercausal influence, moreover, an increase of a, will result in an increase in positivity of the influence of node C on node B along the trail $B \to C$. Analogously, a negative intercausal influence implies that an increase in the probability of b will result in a decrease of positivity of the influence of C on A along the trail $A \to C$ and that an increase in the probability of a will result in a decrease in positivity of the influence of C on B along the trail $B \to C$.

For the network from Fig. 1 we now derive an expression that captures the intercausal influence that is induced between the nodes A and B by the evidence d. We assume that the exact probability distributions for nodes A and B are non-degenerate; note that for degenerate distributions in fact no loop is present. To separate the intercausal influence from the direct influence, we suppose that A and B are independent in the prior network. The intercausal influence then is captured by

$$\Pr(a \mid bd) - \Pr(a \mid \bar{b}d) = \frac{\Pr(abd)}{\Pr(bd)} - \frac{\Pr(a\bar{b}d)}{\Pr(\bar{b}d)}$$

We find that

$$\frac{\Pr(abd)}{\Pr(bd)} = \frac{x \cdot e}{x \cdot e + (1-x) \cdot g}$$

and that

$$\frac{\Pr(a\bar{b}d)}{\Pr(\bar{b}d)} = \frac{x \cdot f}{x \cdot f + (1-x) \cdot h}$$

and hence that

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

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

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

We will use $\mathit{II}(A, B \mid d_j)$ to denote these expressions for $d_j \in \{d, \bar{d}\}$.

We now relate the two qualitative features reviewed above to the over- or underconfidence of the approximations computed for the inner loop nodes by the loopy-propagation algorithm. We recall that, since all entries of the transition matrix $M^{\circlearrowleft A,d}$ are positive, we have that its eigenvalues λ_1 and λ_2 are real numbers and both λ_1 and $\lambda_1 + \lambda_2$ are positive. From the relationship

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

established in the previous section, we therefore find that overconfident approximations will be computed for the probability $\Pr(a_i \mid d_j)$ whenever $\lambda_2 \geq 0$; underconfident approximations are found if $\lambda_2 \leq 0$. With $\lambda_2 = 0$, the exact probabilities are computed. From $\lambda = \frac{1}{2} \cdot [(l + o) \pm \sqrt{(l + o)^2 - 4 \cdot (l \cdot o - m \cdot n)}]$ we observe that the sign of the eigenvalue λ_2 equals the sign of the expression $l \cdot o - m \cdot n$. By simple manipulation of the terms involved, we find that, for $\Pr(a) \in (0, 1)$, the sign of this expression is equal to the sign of

$$(p - q) \cdot \mathit{II}(A, B \mid d_j)$$

We thus find that the approximate probabilities given the evidence d_j, established for node A, are overconfident if the sign of $p - q$ is equal to the sign of $\mathit{II}(A, B \mid d_j)$, that is, if the sign of the qualitative influence between nodes A and B is equal to the sign of the intercausal influence that is induced between A and B by the evidence d_j; the approximations are underconfident otherwise. We recall that the transition matrices of A and B have the same eigenvalues. These nodes therefore will have the same decisiveness.

Example We consider once again the network from Fig. 2. The sign of the qualitative 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 intercausal influence between nodes A and B that is induced 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 nodes A and B therefore have opposite signs and the approximations established for the inner loop nodes will be underconfident. Indeed, we find the underconfident approximations $\widetilde{\Pr}(a \mid d) \approx 0.17$ and $\widetilde{\Pr}(b \mid d) \approx 0.30$ for the exact probabilities $\Pr(a \mid d) \approx 0.09$ and $\Pr(b \mid d) \approx 0.26$ upon loopy propagation. 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 approximations will result. Upon loopy propagation, we indeed find the overconfident approximations $\widetilde{\Pr}(a \mid \bar{d}) \approx 0.28$ and $\widetilde{\Pr}(b \mid \bar{d}) \approx 0.52$ for the exact probabilities $\Pr(a \mid \bar{d}) \approx 0.36$ and $\Pr(b \mid \bar{d}) \approx 0.51$. □

We found that loopy propagation will result in exact probabilities whenever $\lambda_2 = 0$, that is, whenever $(p - q) \cdot \mathit{II}(A, B \mid d_j) = 0$. If the factor $p - q$ equals zero, then the nodes A and B are a priori independent and in fact no loop is present. If the factor $\mathit{II}(A, B \mid d_j)$ equals zero, then the messages that node C sends to node B are independent of the probabilities for A and the messages that C sends to A are independent of the probabilities for B. Nodes A and B therefore receive the correct messages from node C. These messages, moreover, do not change in a next cycle of the algorithm. The algorithm will therefore converge and yield exact probabilities in just a single cycle. We would further like to note that for the probabilities $\Pr(a_i \mid d_j) = 0.5$, loopy propagation will result in exact probabilities, irrespective of the eigenvalues of the transition matrices. This observation can easily be verified by calculating $\widetilde{\Pr}(a \mid d_j)$ from the previously established 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 $\Pr(a \mid d_j) = 0.5$.

We are now also able to explain the different approximation behaviour of the loopy-propagation algorithm demonstrated in Figs. 3 and 4. We recall that Fig. 3 pertains to the algorithm's approximation behaviour for node A given the evidence d in the network from Fig. 2. The qualitative influence between A and B is positive; the evidence d, moreover, serves to induce a negative intercausal influence between the two nodes. We now first consider the algorithm's behaviour during a clockwise cycle. If, in a particular cycle, the influence of entering the evidence d on the probability of 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 the probability of b along the trail $D \leftarrow C \leftarrow A \rightarrow B$ will be positive as well. Because of the negative intercausal influence between nodes A and B, this positive influence on b will result in a decrease in the strength of the positive influence of C on A. In the next cycle, therefore, the increase in the probabilities of a and b induced by the clockwise process, will be less than in the previous cycle. In the subsequent cycle, the positive influence between of C on A, induced by the clockwise process again will be stronger than in the previous cycle, and so on. Such an oscillating behaviour would also be found given a negative influence of D on A along the trail $D \leftarrow C \leftarrow A$. In the counterclockwise cycles of the algorithm a similar behaviour is observed. Both processes add up to the oscillating behaviour that we observed in Fig. 3 for our example network. Alternatively, if the qualitative influence and the intercausal influence have equal signs, the approximations will go steadily towards their final values. If, for example, both influences between the nodes A and B are positive, the increase in the probability of b induced in the clockwise cycle, will result in an even stronger positive influence of C on A. In the next cycle, the increase of the probability of b induced by the clockwise process will further increase, and so on.

So far, we addressed the situation where evidence is entered for node D in the network from Fig. 1. If the convergence node C itself is observed, a similar analysis holds. 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. Although we analysed the influence of the cycling error for the simple network from Fig. 1 only, the essence of our analysis extends to networks of more complex topology. Our analysis, for example, extends directly to networks with simple loops with more than two inner nodes, to networks with multiple simple loops, and to networks with simple loops in which the inner nodes have additional neighbours outside the loop. The essence of our analysis also extends to networks with simple loops having more than one convergence node. In such a network, a cycling error occurs only if for each convergence node of a loop, either the node itself or one of its descendants is observed. An intercausal influence is then found between the parent nodes of each convergence node. We address an arbitrary convergence node C with parents A and B on the loop. Because we consider simple loops only, there will be exactly two trails between A and B in the loop. By exploiting the property of transitivity, we can derive the sign of the indirect influence between A and B along the trail not containing C. We consider this influence to be 'the' qualitative influence between A and B. The intercausal influence between the nodes A and B that is induced by evidence for C or for one of its descendants now can be looked upon as 'the' intercausal influence between A and B. The decisiveness of the approximations established for the inner loop nodes then can be derived, as before, from the signs the two influences mentioned above. Because, effectively this procedure comes down to $\otimes$-combining the signs of all influences along the loop, the convergence node C, used for establishing the decisiveness, indeed can be chosen arbitrarily.

Example We consider the example network with a loop with multiple convergence nodes from Fig. 7. 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, we compute the signs of the two influences between A and B. We choose C to be 'the' convergence node of the loop. The sign of 'the' intercausal influence between A and B given d, then 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 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, we derive that the approximations for the inner loop nodes A and B will be overconfident. We indeed find 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 $\Pr(b \mid d) \approx 0.40$. □

We would like to note that in networks of more complex topology, the probabilities that are needed to determine the signs of the qualitative and intercausal influences between the parents of a convergence node, are not necessarily part of the specification of the network. In some situations, however, these signs can be derived by qualitative reasoning with an abstraction of the network [14,15].

6 Conclusions

When Pearl's propagation algorithm for singly connected networks is applied to networks with loops, the algorithm is no longer exact and approximate probabilities are yielded. In this paper, we identified the two types of error that may be introduced into the approximations: the convergence error and the cycling error. For binary Bayesian networks with simple loops, we identified the factors that determine the effect of the cycling error on the decisiveness of the approximations calculated for the inner loop nodes. We 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 otherwise. Knowledge of the specification of the network thus provides directly for establishing properties of the approximate probabilities computed for the inner loop nodes. So far, we studied the effect of the cycling error on the decisiveness of the approximations for the nodes of a loop 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. In future research, we will focus on gaining further insight into the general performance of the loopy-propagation algorithm by means of controlled experiments. The insights yielded by our study of simple loops will serve to set up such experiments.

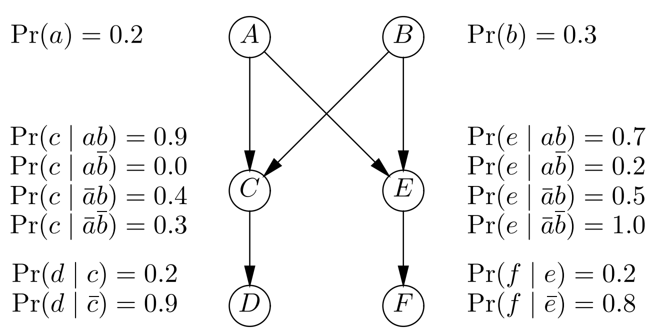

Fig. 7. An example Bayesian network, containing a loop with two convergence nodes

Aknowledgment This research was (partly) supported by the Netherlands Organisation for Scientific Research (NWO).

References

1. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Palo Alto, 1988
2. G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, **42**, 393–405, 1990
3. P. Dagum, M. Luby. Approximate inference in Bayesian networks is NP hard. *Artificial Intelligence*, **60**, 141–153, 1993
4. K. Murphy, Y. Weiss, M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In K.B. Laskey, H. Prade (editors) *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers, San Francisco, 467–475, 1999
5. B.J. Frey. *Bayesian Networks for Pattern Classification, Data Compression and Channel Coding*. MIT Press, 1998
6. D. Mackay, R.M. Neal. Good codes based on very sparse matrices. In C. Boyd (editor) *Cryptography and Coding: 5th SIAM Conference*. Lecture Notes in Computer Science, Springer-Verlag, **1025**, 100–111, 1995
7. Y. Weiss. Interpreting images by propagating Bayesian beliefs. In M.C. Mozer, M.I. Jordan, T. Petsche (editors) *Advances in Neural Information Processing Systems*, **9**, 908–915, 1997
8. Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, **12**, 1–41, 2000
9. Y. Weiss, W.T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, **13**, 2173–2200, 2001
10. J.H. Bolt, L.C. van der Gaag. The convergence error in loopy propagation. Paper presented at the International Conference on Advances in Intelligent Systems: Theory and Applications, in cooperation with the IEEE computer society, Luxembourg, 2004
11. G. Strang. *Linear Algebra and Its Applications*. Academic Press, 1980
12. M.P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, **44**, 257–303, 1990
13. M.P. Wellman, M. Henrion. Qualitative intercausal relations, or explaining "explaining away". In *KR-91, Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*. Morgan Kaufmann Publishers, Menlo Park, California, 535–546, 1991
14. 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
15. M.J. Druzdzel, M. Henrion. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*. AAAI Press, Menlo Park, California, 548–553, 1993