Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
47 (2008) 202–218
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

An approach to hybrid probabilistic models

E. Di Tomaso [a,b,*,1], J.F. Baldwin [a]

[a] *Department of Engineering Mathematics, University of Bristol, Queen's Building, University Walk, Bristol BS8 1TR, UK*
[b] *Istituto di Metodologie per l'Analisi Ambientale, IMAA/CNR, 85050 Tito, PZ, Italy*

Received 12 July 2006; received in revised form 19 January 2007; accepted 30 April 2007
Available online 10 May 2007

Abstract

This paper is concerned with a development of a theory on probabilistic models, and in particular Bayesian networks, when handling continuous variables. While it is possible to deal with continuous variables without discretisation, the simplest approach is to discretise them. A fuzzy partition of continuous domains will be used, which requires an inference procedure able to deal with soft evidence. Soft evidence is a type of uncertain evidence, and it is also a result of the type of discretisation used. An algorithm for inference in multiply connected networks will be proposed and exploited for filtering and abduction in dynamic, time-invariant models, when continuous variables are present.
© 2007 Elsevier Inc. All rights reserved.

Keywords: Bayesian networks; Fuzzy partition; Soft evidence; Inference; Temporal models

1. Introduction

Reasoning under uncertainty has a central role in Artificial Intelligence. Probabilistic models within Soft Computing, are suitable for dealing with a certain type of uncertainty. Among probabilistic models, in particular *Bayesian networks* (BNs) are a powerful tool for knowledge representation and inference. They combine probability theory and graph theory providing an instrument for dealing with uncertainty and complexity. Their graphical representation allows the decomposition of a complex system in simpler parts and furthermore provides expressive models. The general framework for Bayesian networks was first developed by Pearl [26] and subsequently refined by Lauritzen and Spiegelhalter [16], and Jensen [11]. The first applications were done in probabilistic expert systems responding to the need of having expert systems that could use probability theory in a tractable way. Since then, they have acquired an important role in AI. They have been used primary as diagnostic systems in medicine, then as general tools for decision making, forecasting, monitor and control.

* Corresponding author. Address: Istituto di Metodologie per 1' Analisi Ambientale, IMSS/CNR, 85050 Tito, PZ, Italy. Tel.: +39 0971 427 292.
E-mail addresses: enza.di-tomaso@bristol.ac.uk, ditomaso@imaa.cnr.it (E. Di Tomaso), Jim.Baldwin@bris.ac.uk (J.F. Baldwin).
1 Supported by the British Telecom.

0888-613X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2007.04.004

There are in particular two aspects related to the use of Bayesian networks, that are still subject of research and that may occur in real-world applications: handling continuous variables, and inferring solutions from uncertain findings. There are two basic approaches to deal with continuous variables: the one that resorts to specific families of probability distributions and the one that uses a finite partition of the continuous domains. The first approach consists in the definition of standard families of probability functions representing the priors and conditional probabilities in the BN. The functions are specified by a finite number of parameters. A common choice is to use conditional linear Gaussian (CLG) distributions. CLG models do not allow discrete variables to have continuous parents. There are attempts to overcome this limitation using a combination of Gaussian and softmax functions [19,23], but in most cases, they need to resort to the use of approximate methods for the inference. Their convergence can be quite slow and very sensitive to the parameter choice. In general, these attempts need some strong assumptions particularly on the interaction between discrete variables and their continuous parents. Mixture of Truncated Exponentials (MTE) models [18] allow the use of discrete variables with continuous parents, and exact propagation algorithms can be performed over them; yet they rely on the non-trivial estimation of the parameters describing the MTE densities. The second approach consists in the discretisation of continuous domains. A common choice is to define a partition using a finite number of intervals. This requires the specification of set of threshold values to specify the intervals. The set of thresholds may be supplied by an expert. Techniques have be applied to find the optimal set of thresholds while inducing the model from data but they can become computationally very expensive when applied to large databases [8].

We shall propose an approach to handle continuous variables using a discretisation of continuous domains. We shall define a type of partition that requires only the specification of the number of partition sets. The proposed type of partition requires changes in the traditional approach to probabilistic inference. The same inference procedure will be able to process both continuous findings and soft evidence. Soft evidence is a type of uncertain evidence that occurs when uncertainty is represented through probability. The inference procedure for dealing with soft evidence has been proposed originally by the authors in [2]. It resorts to an iterative proportional fitting procedure in order to perform belief updating when soft evidence is given. A similar procedure has also been independently proposed in [30] for dealing with agent communication, and in [14] where it is suitable for a restricted type of applications where uncertain evidence affects a limited and fixed number of variables.

2. The use of soft evidence for continuous variables

In this section we shall describe an approach to handle continuous variables using a type of uncertain evidence called soft evidence. We shall also discuss the appropriateness of the use of soft evidence rather than likelihood evidence, which is an alternative means to deal with uncertain findings.

2.1. Soft evidence

Evidence is new information about any of the random variables that a Bayesian network models. Typically, evidence is the observation that a variable is in one of its possible states or values. For example, a binary variable *Rain* may be observed being in the state *yes* or *not*; if the state *yes* is observed, the variable *Rain* is instantiated to that value, which we will indicate as $Rain = yes$. The type of evidence described so far is also termed *hard evidence*, as complementary to other types of evidence called soft and likelihood evidence, which are used to model uncertain findings.

While hard evidence assigns one exact value to each of the evidence variables, i.e. the variables subjected to new findings, *soft evidence* [24] is not a delta function but specifies a probability distribution for the evidence variable. Soft evidence maps all the values of the evidence variables to $[0, 1]$.

Definition 2.1 (*Soft evidence*). Given a variable X defined on the domain Ω_X, a soft evidence E is a function

$$E : \Omega_X \mapsto [0, 1]$$

such that

$$\forall x_m \in \Omega_X, \quad 0 \leqslant E(x_m) \leqslant 1, \\ \sum_{x_m \in \Omega_X} E(x_m) = 1. \tag{1}$$

For example, a certain uncertain observation about the variable *Rain* could be represented as the soft evidence $E(yes) = 0.8$ and $E(not) = 0.2$.

2.2. Soft evidence versus likelihood evidence

Uncertain evidence is commonly handled through virtual nodes, and hence likelihood evidence.

The likelihood evidence approach requires adding virtual nodes to the structure, and for each of them a CPT associated with the new link. For example, a certain uncertain observation about the variable *Rain* is handled creating a virtual node *V* with values *true* and *false* as a child node of *Rain*. The CPT of *V* given *Rain* is represented by the likelihood ratio:

$$\frac{P(V = true|Rain = yes)}{P(V = true|Rain = not)} = \frac{Odds(Rain = yes|V = true)}{Odds(Rain = yes)} \tag{2}$$

The evidence $V = true$ is entered, and propagation is performed. It is important to notice that the node *Rain* is not instantiated to any particular value and its beliefs may change if another node that influences *Rain* is subjected to observation. A detailed explanation of the use of likelihood evidence can be found in [15].

Likelihood evidence represents a subjective statement that can be improved by something observed later, while uncertain observations that cannot be improved by anything observed later are represented using soft evidence. We can argue that the use of soft evidence is complementary to the use of the likelihood evidence when dealing with uncertainty.

Example 2.1. Let us consider a court case where two witnesses, John and Mary, are asked to give evidence on whether Ms X is guilty or not of having committed a certain crime.

Suppose that the structure in Fig. 1 is used to model the case, where the node 'Guilty' represents 'Ms X is guilty' and takes values 'true' and 'false', the nodes 'John' and 'Mary' represent respectively 'John and Mary saw Ms X committing the crime', and have value 'true' and 'false'. Suppose that Mary thinks she has seen Ms X committing the crime, but she is only 70% sure about it. The uncertain evidence provided by Mary could be interpreted as likelihood evidence. In this case a virtual node *V* is added to the model, the likelihood ratio is computed, and the evidence $V = true$ is entered. The updated values after propagation is performed, are shown in Fig. 2.

Further, suppose that John gives evidence that he saw Ms X committing the crime. According to the likelihood evidence approach, the additional information provided by John will modify the beliefs of the node 'Mary' as shown in Fig. 3.

If we interpret Mary's uncertain evidence as soft evidence that cannot be modified by anything observed later, the updated values given Mary's testimony are equal to the ones shown in Fig. 2. The additional information provided by John will not change the beliefs of the node 'Mary'. The results after propagation are shown in Fig. 4.

The reasoning and the algorithm which lead to these results when soft evidence is considered, will be explained in the following sections.

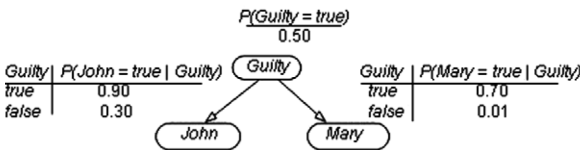

Fig. 1. Two-witness network.

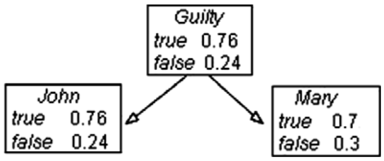

Fig. 2. Posterior probabilities after Mary's testimony.

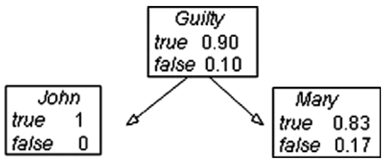

Fig. 3. Posterior probabilities after Mary's and John's testimony (likelihood evidence approach).

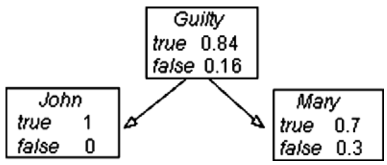

Fig. 4. Posterior probabilities after Mary's and John's testimony (soft evidence approach).

The above simple example illustrates the consequences of using likelihood evidence versus soft evidence. Should Mary's uncertain testimony be 'improved' by John's testimony? The soft evidence approach allows one to fix the node 'Mary' to the values 0.7 and 0.3 in a way that generalises hard evidence, i.e. the findings cannot be modified by further evidence. As a consequence of John's additional testimony of Ms X being guilty, the posterior probability of $Guilty = true$ is higher when using the likelihood evidence approach than when we treat Mary's uncertain statement as soft evidence. Hence soft evidence is the appropriate means of dealing with uncertain evidence when one wants to fix the beliefs of a node to a probability distribution. Further discussions on the distinction between likelihood evidence and soft evidence can be found in [30,4].

2.3. Discretisation of continuous domains

In this section we shall relate soft evidence to the use of continuous variables.

While in many applications it is possible to deal with continuous variables without discretisation, the simplest approach is to discretise them. The continuous variable domains are discretised with a finite set of threshold values, defined on the original continuous frame. Given the continuous variable Y, defined on the domain Λ_Y, the discretisation is the definition of a partition $y_1, \ldots y_{n_Y}$ of Λ_Y.

We shall discretise continuous domains with a *fuzzy partition* [28].

Definition 2.2 (*Fuzzy partition*). A fuzzy partition on the universe Ω is the set of fuzzy sets [33] $\{f_1, \ldots, f_p\}$ such that $\forall x \in \Omega$

$$\sum_{i=1}^{p} \chi_{f_i}(x) = 1 \tag{3}$$

where χ_{f_i} is the membership function of f_i, i.e. a function

$$\chi_f : \Omega \mapsto [0, 1]$$

We shall use equally spaced, symmetric fuzzy sets as in Fig. 5.

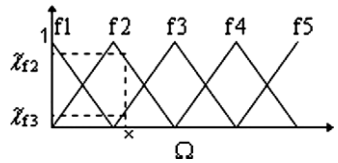

Fig. 5. Fuzzy partition.

Let X be a variable defined on $\Omega = [Inf_\Omega, Sup_\Omega]$ and $\{f_1, \ldots, f_p\}$ a partition of Ω as in Fig. 5. Let $I = \frac{Sup_\Omega - Inf_\Omega}{p-1}$. For $X = x$ and $k \in \{1, \ldots, p-1\}$, if

$$Inf_\Omega + (k-1)I \leqslant x \leqslant Inf_\Omega + kI$$

the values $\chi_{f_k}(x)$ and $\chi_{f_{k+1}}(x)$ are

$$\chi_{f_k}(x) = k - \frac{x - Inf_\Omega}{I} \tag{4}$$

$$\chi_{f_{k+1}}(x) = \frac{x - Inf_\Omega}{I} - k + 1 \tag{5}$$

The two membership values define a probability distribution over the elements of the partition [1]. This is due to the fact that we use equally spaced, symmetric fuzzy sets.

The proposed discretisation is related to the use of soft evidence. Let Y be a continuous variable defined on Ω_Y, and let $\{f_1, \ldots, f_p\}$ be a fuzzy partition of Ω_Y. The continuous finding $Y = y$ corresponds to the soft evidence $E(f_k) = \chi_{f_k}(y)$ and $E(f_{k+1}) = \chi_{f_{k+1}}(y)$, where $k \in \{1, \ldots, p-1\}$, and the two membership functions χ_{f_k} and $\chi_{f_{k+1}}$ are as in (4) and (5).

Continuous findings cannot be treated as likelihood evidence since they do not correspond to subjective statements that can be modified by further evidence. Soft evidence is appropriate to deal with continuous variables discretised as explained above, since it is able to generalise hard evidence: the beliefs of a continuous node discretised with a 2-element fuzzy partition will be fixed for example to the values 0.95 and 0.05, as the beliefs of a binary discrete node are fixed for example to the values 1 and 0.

3. Soft updating of a joint distribution

The type of dicretisation proposed in the previous section requires a probabilistic inference able to deal with soft evidence. We shall consider here the process of updating a joint distribution. Soft evidence can be interpreted as a constraint to the distribution. The *minimum relative entropy* (MRE) criteria finds a solution to the problem of updating a distribution when another distribution is given as constraint. Let q_k be a prior distribution that is subjected to some constraints C. Among all the distributions p_k that satisfy the constraints, the minimum relative entropy criteria estimates the one that yields the minimum relative entropy with respect to the prior distribution q_k:

$$d(p'_k, q_k) = \min_{p_k \in C} d(p_k, q_k) \tag{6}$$

where d is the *relative entropy* of p_k with respect to q_k.

Definition 3.1 (*Relative entropy*). Let p_k and q_k be two discrete probability distributions on a measurable space $(\Omega, \mathscr{F})$. The relative entropy of p_k with respect to q_k is defined by

$$d(p_k, q_k) = \sum_k p_k \log_2 \frac{p_k}{q_k} \tag{7}$$

where d is a convex function of p_k, is always non-negative, and equals zero if and only if $p_k = q_k$ [6]. d is also called Kullback–Leibler (K-L) distance or I-divergence and denotes the information difference between two distributions. Minimising the difference between the estimated p_k and the prior q_k is equivalent to projecting the prior onto the set of possible distributions allowed by the data. The existence of a solution has been discussed in [7]. If a solution to the MRE criteria exists, then the solution is unique. The uniqueness is straightforward to prove due to $d(p_k, q_k)$ being a convex function of p_k.

The *iterative proportional fitting* (IPF) procedure adjusts a distribution q_k to a set of c arbitrary marginal distributions. It consists in the cyclical application of the MRE criteria for each of the c constraints and converges to p'_k, the solution to the MRE criteria subjected to all the given constraints. It starts from $p^0_k = q_k$ and evaluates p^1_k as the distribution that yields the minimum relative entropy with respect to p^0_k and satisfies the first constraint. Subsequently, p^1_k takes the place of p^0_k and is updated with respect to the second constraint to p^2_k. For a generic constraint n it yields that

$$d(p^n_k, p^{n-1}_k) = \min_{p_k \in C_n} d(p_k, p^{n-1}_k) \tag{8}$$

where C_n is the set of probabilities distributions that satisfies the nth constraint. The constraints are cyclically repeated so that in the sth cycle

$$C_n = C_i, \quad n = (s-1) \times c + i \tag{9}$$

where C_i is the set of distributions satisfying the ith constraint for $1 \leqslant i \leqslant c$. The solution to the IPF procedure is a distribution p'_k that satisfies

$$p'_k = \lim_{n->\infty} p^n_k \tag{10}$$

The convergence of this procedure has been discussed in [7] proving the following theorem:

Theorem 3.1. *Let $C_1, \ldots, C_c$ be arbitrary sets of distributions on a finite space with $C = \cap^c_{i=1} C_i$ and $C \neq \emptyset$. Let q_k be a probability distribution such that there exists a distribution $p_k \in C$ with $d(p_k, q_k) < \infty$. If $p^1_k, p^2_k, \ldots$ are defined recursively as in (8) and (9), then p^n_k converges to p'_k satisfying*

$$d(p'_k, q_k) = \min_{p_k \in C} d(p_k, q_k) \tag{11}$$

We shall apply the minimum relative entropy criteria to the inference process of updating the joint distribution of a set of variables when soft evidence is given. Let $V = \{V_1, \ldots, V_n\}$ be a set of n variables and $P(V)$ their joint distribution. The updated distribution $P'(V)$ given a set of constraints, is chosen as close as possible in the probability space, to the prior distribution $P(V)$, that is it satisfies:

$$\min_{\{P'(V)\}} \sum_V P'(V_i) \log_2 \frac{P'(V_i)}{P(V_i)} \tag{12}$$

Suppose a new finding on a continuous variable V_i is known, and the finding is specified by the soft evidence $E(V_i)$. Minimizing the relative entropy of $P'(V)$ with respect to $P(V)$ is equivalent in this case, to applying Jeffrey's rule of updating [10]:

$$P'(V) = \sum_j P(V|v_{ij}) E(v_{ij}) \tag{13}$$

where v_{ij} is a possible state of V_i.

The Eq. (13) can equivalently be expressed as

$$P'(V) = \frac{P(V)E(V_i)}{P(V_i)} \tag{14}$$

Example 3.1. Let us consider a set of binary variables $V = \{A, B, C\}$ with a joint probability as in Table 1 and a soft evidence on the variable A as $E(A) = (0.2, 0.8)$. According to Jeffrey's rule as in (13) or (14), the updated distribution is in Table 2. Computing the marginal probabilities for A, we find that $P'(A) = (0.2, 0.8)$, as expected. The finding on the variable A remains unchanged after it has been absorbed in the model.

Table 1
Joint distribution $P(ABC)$

	b_1c_1	b_1c_2	b_2c_1	b_2c_2
a_1	0.28	0.12	0.09	0.01
a_2	0.23	0.12	0.145	0.005

Table 2
Updated joint distribution given $E(A)$

	b_1c_1	b_1c_2	b_2c_1	b_2c_2
a_1	0.112	0.048	0.036	0.004
a_2	0.368	0.192	0.232	0.008

Table 3
Updated joint distribution given $E(A)$ and $E(B)$ with 0 iterations

	b_1c_1	b_1c_2	b_2c_1	b_2c_2
a_1	0.047	0.02	0.09	0.01
a_2	0.153	0.08	0.58	0.02

Table 4
$P'(A)$ at different steps of iteration

	it. 0	it. 1	it. 2	it. 3	it. 4
$P(a_1)$	0.16666667	0.19963702	0.19999607	0.19999996	0.2
$P(a_2)$	0.83333333	0.80036298	0.80000393	0.80000004	0.8

In case the finding consists of evidence on more than one variable, Jeffrey's rule is applied within an iterative procedure, which is consistent with the minimum relative entropy criteria.

Example 3.2. Let us consider again the joint distribution as in Table 1. Suppose this time we have two findings as $E(A) = (0.2, 0.8)$ and $E(B) = (0.3, 0.7)$. According to Jeffrey's rule as in (13) or (14), the updated distribution when no iteration is applied, is as in Table 3. Computing the marginal probabilities for B, we find that $P'(B) = E(B)$, while for A we obtain $P'(A) = (0.167, 0.833)$, which is different from the finding on A. The presence of a second evidence variable, B, does not allow the finding on the first variable, A, to remain unchanged after the updating. If we iterate our updating procedure 4 times, we obtain that $P'(A) = E(A)$. The resulting values of $P'(A)$ after each iteration are in Table 4. The number of iterations depends on the precision we require. Table 4 shows that we get the exact solution for $P(A)$ after 1 iteration with the precision of 10^{-3}, after 2 iterations with the precision of 10^{-5} and after 3 iterations with the precision of 10^{-7}.

4. Soft updating of hybrid Bayesian networks

The inference process in Bayesian networks determines the posterior probability for a set of query variables, given that some evidence variables have been instantiated to specific values. The inference algorithms can be classified into two major groups according to if they provide an exact or an approximate solution. We shall base our analysis on the first group. The existing algorithms for exact inference tend to handle only exact values for the evidence variables, i.e. hard evidence. In order to deal with continuous findings, we shall use the MRE criteria for updating the joint distribution $P(V)$ of a set of variables V, where $P(V)$ is represented through a Bayesian network. We base our inference procedure on Lauritzen and Spiegelhalter's *join tree* algorithm [16], which produces exact inference. It is based on Pearl's algorithm [25] and allows a Bayesian network to have any sparse structure. Other inference algorithms like the HUGIN algorithm [13] or the lazy propagation algorithm [17], could equivalently be considered for the same purpose.

4.0.1. Lauritzen and Spiegelhalter's algorithm

Lauritzen and Spiegelhalter developed an exact probabilistic method for propagating the evidence in Bayesian networks. Their method involves topological changes in the structure of the net: it builds a join tree whose vertices are cliques of a moral and triangulated graph derived from the graph of the BN. The topological order of the cliques in the join tree is such that the cliques hold the running intersection property. A probabilistic propagation method is applied to the clique tree and the related potential representation $(\{Clq_1 \ldots Clq_{n_C}\}, \Psi)$, where $\{Clq_1 \ldots Clq_{n_C}\}$ is the set of cliques and Ψ is a function

$$\Psi : \Omega_1 \times \ldots \times \Omega_n \mapsto \mathbb{R}$$

such that

$$P(V) = \text{const} \prod_{i=1}^{n_C} \Psi(Clq_i) \tag{15}$$

The join tree algorithm computes the probability $P(Clq_i)$ or, once the evidence has been absorbed in the function Ψ, the updated probability $P'(Clq_i)$, where Clq_i is a generic clique of the join tree. The marginal distribution of a variable can be computed from the probability of the clique the variable belongs to. We shall give an overview on the theory that justifies the join tree algorithm [21].

Theorem 4.1. *Let (S_i, R_i) be partition of a clique Clq_i such that $S_i = Clq_i \cap (Clq_1 \cup \ldots \cup Clq_{i-1})$ and $R_i = Clq_i - S_i$, then*

$$P(Clq_i|S_i) = P(R_i|S_i) \tag{16}$$

It follows from Theorem 4.1 that for $1 \leqslant i \leqslant n_C$

$$P(Clq_i) = P(R_i|S_i)P(S_i) \tag{17}$$

The aim of the join tree algorithm is to solve (17) for each clique of the tree. The following two theorems deal with the computation of the first factor of Eq. (17).

Theorem 4.2. *Let $\lambda(S_{n_C}) = \sum_{R_{n_C}} \Psi(Clq_{n_C})$, then it holds that*

$$P(R_{n_C}|S_{n_C}) = \frac{\Psi(Clq_{n_C})}{\lambda(S_{n_C})} \tag{18}$$

Theorem 4.3. *Let $(\{Clq_i\}, \Psi)$ for $1 \leqslant i \leqslant n_C$ be a potential representation of the joint distribution $P(V)$, Clq_j be one of the clique that satisfy $S_{n_C} \subseteq Clq_j$ and Ψ'' be a function defined for $1 \leqslant i \leqslant (n_C - 1)$ as*

$$\Psi''(Clq_i) = \begin{cases} \Psi(Clq_i) & \text{if } i \neq j \\ \Psi(Clq_i)\lambda(S_{n_C}) & \text{if } i = j \end{cases} \tag{19}$$

$(\{Clq_1 \ldots Clq_{n_C-1}\}, \Psi'')$ is a potential representation of the marginal distribution on the set $\{Clq_1 \ldots Clq_{n_C-1}\}$, relative to $P(V)$.

Theorem 4.2 allows one to compute $P(R_i|S_i)$ for the clique that is last in the topological order of the cliques, Theorem 4.3 builds a potential representation on the remaining cliques so that Theorem 4.2 can subsequently be applied to them. The alternate application of the two theorems till reaching the root clique allows one to compute the conditional probabilities $P(R_i|S_i)$ for each clique of the tree. λ can be thought of as a message that a clique sends upwards to its parents. Being $S_1 = \emptyset$ by definition, $P(R_1|S_1) = P(R_1)$ and $P(Clq_1) = P(R_1)$, which solves the (17) for Clq_1. It follows that for any clique Clq_j such that $S_j \subseteq Clq_1$

$$P(S_j) = \sum_{Clq_1 - S_j} P(Clq_1) \tag{20}$$

and hence, let $\pi(S_j) = P(S_j)$,

$$P(Clq_j) = P(R_j|S_j)\pi(S_j) \tag{21}$$

The alternate application of (20) and (21) till reaching the leaf cliques allows one to compute the probabilities $P(Clq_i)$ for each clique of the tree. π can be thought of as a message that a clique sends downwards to its children. As mentioned above, if the evidence has been absorbed into the potential representation, i.e. the evidence variables have been instantiated, the algorithm computes the updated marginal over the cliques. Lauritzen and Spiegelhalter's join tree algorithm deals only with hard evidence.

4.0.2. Soft updating algorithm

We shall distinguish here the case of one evidence variable only, and the one of two or more evidence variables. In the first case we shall apply the join tree algorithm n_X times, where n_X is the number of possible values of the evidence variable X. The soft evidence $\mathrm{E}(X)$ is split in n_X hard evidence function $\mathrm{E}(X) = \delta(r)$, for $1 \leqslant r \leqslant n_X$. The resulting updated marginals over the cliques $P'(Clq_i) = P(Clq_i|\mathrm{E}(X))$ are combined in $P''(Clq_i) = P(Clq_i|E(X))$ according to (13). If the finding consists of the evidence on two or more nodes, we shall apply the soft updating within an iterative procedure. The iterative procedure will be built taking into account that the join tree algorithm applies to potential representations. Let $U \subseteq V$ be a set of m evidence nodes of the Bayesian network and $(\{Clq_1 \dots Clq_{n_C}\}, \Psi)$ a potential representation of $P(V)$. We shall apply the soft updating procedure considering the first variable U_1 and obtain $P''(Clq_i) = P(Clq_i|E(U_1))$. Let $\pi^{-1}(S_j) = 1/\sum_{Clq_1 - S_j} P''(Clq_1)$ for any clique Clq_j such that $S_j \subseteq Clq_1$. It follows from Theorem 4.1 that

$$P''(R_j|S_j) = P''(Clq_j)\pi^{-1}(S_j) \tag{22}$$

The application of (22) till the leaf cliques allows one to compute $P''(R_i|S_i)$ for $2 \leqslant i \leqslant n_C$. π^{-1} can be thought of as a message a clique sends downwards to its children. Let us consider the function

$$\Psi''(Clq_i) = \begin{cases} P''(Clq_i) & i = 1 \\ P''(R_i|S_i) & 2 \leqslant i \leqslant n_C \end{cases} \tag{23}$$

$(\{Clq_i\}, \Psi'')$, for $1 \leqslant i \leqslant n_C$, is a potential representation of $P(V)$ because of the following theorem [21]:

Theorem 4.4. *Let $(\{Clq_i\}, \Psi)$ for $1 \leqslant i \leqslant n_C$ be a potential representation of the joint distribution $P(V)$, if the ordering $\{Clq_1 \dots Clq_{n_C - 1}\}$ has the running intersection property, then*

$$P(V) = P(Clq_1) \prod_{i=2}^{n_C} P(R_i|S_i) \tag{24}$$

We shall apply the soft updating procedure considering the second variable U_2 and Ψ'' as in (23). The application of (22) and (23), and the soft updating for each $U_k \in U$ complete the first loop of the iterative process. We repeat the procedure till the solution converges. Let $U \subseteq V$ be a set of m evidence nodes of the Bayesian network. Let u_{kl} be one of the m_k generic values of the variable $U_k \in U$. Let $Ch(Clq_i)$ be the set of children of a clique Clq_i in the join tree. The description of the soft updating algorithm is in Algorithm 1.

Algorithm 1 (*Soft updating for multiply connected nets*).

$(\{Clq_i\}, \Psi)$, $1 \leqslant i \leqslant n_C$
(S_i, R_i)
$E(U)$
ϵ_S
$\Psi''(Clq_i) \Leftarrow \Psi(Clq_i)$ {initialisation}
repeat
  $n_{\text{it.}} \Leftarrow n_{\text{it.}} + 1$ {a new iteration starts}
  $\Phi(Clq_i) \Leftarrow P''(Clq_i)$ {store P'' at each iteration in Φ}
  {go through the variables U_k in U}
  for $k = 1$ to m **do**

$\Psi(Clq_i) \Leftarrow \Psi''(Clq_i)$ {the updating is done in a sequence}
$P''(Clq_i) \Leftarrow 0$ {initialization of P'' for the step $(mn_{\text{it.}} + k)$}
{go through the values u_{kl} of U_k}

for $l = 1$ to m_k **do**
  $\mathrm{E}(U_k) \Leftarrow \delta(u_{kl})$
  $P'(Clq_i) \Leftarrow P(Clq_i|\mathrm{E}(U_k))$ {from hard updating algorithm}
  $P''(Clq_i) \Leftarrow P''(Clq_i) + P'(Clq_i)E(u_{kl})$
endfor

$\Psi''(Clq_1) \Leftarrow P''(Clq_1)$
if $Ch(Clq_i) \neq \emptyset$ **then**
  $\pi^{-1}(S_j), S_j \subseteq Clq_i$
endif

if $Clq_i \leftarrow \pi^{-1}(S_i)$ {Clq_i receives a π^{-1} message from a parent} **then**
  $\Psi''(Clq_i) \Leftarrow P''(Clq_i)\pi^{-1}(S_i)$
endif
endfor
until $|P''(Clq_i) - \Phi(Clq_i)| < \epsilon_S$
$P''(Clq_i)$ {the updated distribution over the cliques: $P(Clq_i|E(U))$}
$P''(V)$ {the updated marginals estimated from $P''(Clq_i)$}

We shall discuss the time computational complexity of the soft evidence algorithm for multiply connected networks in terms of the number of elementary arithmetic operations needed. The parameters involved in the complexity analysis are as in Table 5.

We shall consider the different phases of the algorithm:

The λ and π procedures. There are 5 major executions related with this first phase: the computation of λ, the divisions and multiplications respectively in (18) and (19), the computation of π and the multiplications in (21). Each of them consists of basic operations on every combination of the variables in a clique. Hence each of them consists of pr^m operations at the most. Overall the upper bound of the number of basic operations required in the λ and π procedures is $5pr^m$. This phase corresponds to the hard updating algorithm for multiply connected nets. Considering that $p \leqslant n$, its computational complexity is $O(nr^m)$.

The cycle for every value u_{kl} of U_k. Given an evidence variable U_k, the procedures in the first phase are repeated for every value u_{kl} of U_k such that $E(u_{kl}) \neq 0$. Taking r as an upper bound for the number of such values, the number of operations needed at this stage is of the order of $5pr^{m+1}$.

Jeffrey rule. Every cycle of the previous phase is followed by a weighted sum. This requires pr^{m+1} further operations. The overall amount of operations at this stage is $6pr^{m+1}$

The π^{-1} procedure. There are 2 major executions related with this phase: the computation of π^{-1} and the multiplications in (22). This adds $2pr^m$ basic operations.

The cycle for every variable $U_k \in U$. All the previous phases are repeated for every variable U_k in the set of evidence variables U. Considering that $|U| \leqslant n$, the total number of operations is multiplied by n.

Table 5
Parameters in the complexity analysis of the soft updating algorithm

n	Number of variables
r	Maximum number of values for a variable=$\max_i|\Omega_i|$
p	Number of cliques
m	Maximum number of variables in a clique=$\max_j|Clq_j|$
$n_{\text{it.}}$	Number of iterations

The iterative procedure. Given that the convergence is reached in $n_{\text{it.}}$ iterations, the total number of basic operations needed by the execution of the algorithm is $(6r+2)pr^m nn_{\text{it.}}$. Considering that $p \leqslant n$, its upper bound is $(6r+2)n^2r^m n_{\text{it.}}$ From the above analysis it arises that the computational complexity of the soft updating algorithm is $O(n^2r^m n_{\text{it.}})$. As expected, the algorithm has an exponential complexity depending on the parameter m that is inherited from the hard updating algorithm. In fact the computation of probabilistic inference in BNs is an NP-hard problem [5]. Comparing further the soft updating algorithm with the hard updating one, the linear dependency on n becomes quadratic and there is an ulterior linear dependency on $n_{\text{it.}}$. The parameter $n_{\text{it.}}$ depends on the cardinality of U or more exactly on $n_{\text{val.}}$: the overall number of values u_{kl} such that $E(u_{kl}) \neq 0$. The upper bound of $n_{\text{val.}}$ is rn. The dependency of $n_{\text{it.}}$ from $n_{\text{val.}}$ can be kept linear for relative small value of $n_{\text{val.}}$.

In order to show the convergence of the soft evidence algorithm for multiply connected networks, we shall show that the hypothesis of Theorem 3.1 are satisfied by the features involved in the algorithm, and we shall discuss when those hypothesis are not satisfied. The algorithm considers a joint distributions $P(V)$ relative to a set of variables V which is represented by $P(Clq_i)$, the probability distribution over the set of cliques. Let us consider m sets of multivariate distributions $F_k(U)$ defined on the finite set of evidence variables $U \subseteq V$ such that

$$\sum_{U_k} F_k(U) = E(U_k) \tag{25}$$

The soft evidence functions are therefore the constraints to $P(Clq_i)$ and the sets of distributions $F_k(U)$ correspond to the $C_1, \ldots, C_c$ arbitrary sets of distributions of Theorem 3.1. Let $F(U) = \cap_{k=1}^{m} F_k$, then $F(U)$ is the set of multivariate distributions whose marginals over U_k are the functions $E(U_k)$ and therefore satisfy all the m constraints. The constraints $E(U_k)$ are cyclically considered in the algorithm as in (9). Considering a generic step $n = n_{\text{it.}} \times m + k$, then

$$d(P^n(Clq_i), P^{n-1}(Clq_i)) = \min_{P(Clq_i)\in F_n} d(P(Clq_i), P^{n-1}(Clq_i)) \tag{26}$$

since the application of Jeffrey's rule at each nth step satisfies the MRE criteria. According to Theorem 3.1, $P^n(Clq_i)$ converges to the right solution if

$$S(P(Clq_i), \infty) \cap F(U) \neq \emptyset \tag{27}$$

Eq. (27) is satisfied only when the constraints $E(U_k)$ are consistent with the model represented by $P(V)$. Inconsistent evidence in probabilistic models has been studied in [32]. We shall describe an example where there is inconsistency between the evidence and the model.

Example 4.1. Let us consider two nodes A and B part of a certain Bayesian network, being A parent of B. Let A and B be defined respectively on the continuous domains $\Omega_A = [0, 12]$ and $\Omega_B = [0, 1]$ and suppose that Ω_A and Ω_B are discretised by a fuzzy partition of respectively 5 and 3 sets. The CPT related to $P(B|A)$ is in Table 6.

Let us consider the findings $A = 12$ and $B = 0.5$. They correspond to the hard evidence $E(a_5) = 1$ and $E(b_2) = 1$. As expected, there is no solution in this case since $P(b_2|a_5) = 0$. Similarly, there is no solution for the findings $A = 6$ and $B = 1$. Every 0 entry in the CPT is related to a domain of inconsistent evidence which, in the case of hard evidence, corresponds to two points in the cross product space $\Omega_A \times \Omega_B$:

$$\Omega_{\text{ih}} = \{(6, 1), (12, 0.5)\} \subseteq \Omega_A \times \Omega_B$$

The domain of inconsistent soft evidence is instead more extended. In this example there is not exact solution if the evidence function is defined on the domain:

$$\Omega_{\text{is}} = [3, 9] \times [0.5, 1] + [9, 12] \times [0, 1] \subseteq \Omega_A \times \Omega_B$$

Although there is no exact solution in Ω_{is}, the soft evidence algorithm provides an approximate solution in part of the Ω_{is} domain that is far enough from Ω_{ih}. The approximate solution is dependent on the order in which the evidence variables are considered in the updating process. In Table 7 there are some examples of approximate solutions for the variable A when $B = 0.75$, and when the prior probability of A is $P(a_1, a_2, a_3, a_4, a_5) = (0.03, 0.3, 0.2, 0.3, 0.17)$.

Table 6
Conditional probability $P(B|A)$

	a_1	a_2	a_3	a_4	a_5
b_1	0.7	0.5	0.25	0.4	0.75
b_2	0.2	0.3	0.75	0.5	0
b_3	0.1	0.2	0	0.1	0.25

The example above shows that the soft evidence algorithm converges even in some inconsistent situations to a solution that is somehow close to the findings. This does not solve the problem of inconsistent evidence since in general the algorithm cycles or the solution is dependent on the order in which the evidence variables are considered.

A further important aspect in the convergence analysis is the study of the behaviour of $n_{\mathrm{it.}}$, the number of iterations which lead to the convergence. $n_{\mathrm{it.}}$ increases as the number of soft evidence variables, or more exactly as the number $n_{\mathrm{val.}}$ of values u_{kl} with $E(u_{kl}) \neq 0$, increases. The number of iterations in the IPF is related to the stopping criteria which verifies that the procedure converges. In the Algorithm 1 the stopping criteria is $|P''(Clq_i) - \Phi(Clq_i)| < \epsilon_S$. We shall study the behaviour of $n_{\mathrm{it.}}$ at different values of the parameter ϵ_S. Let us consider the Stud farm model [11]. In Figs. 6 and 7 are represented the values of $n_{\mathrm{it.}}$ as function of $n_{\mathrm{val.}}$ when ϵ_S is respectively equal to 10^{-6} and 10^{-7}. Fig. 8 shows that the value of $n_{\mathrm{it.}}$ increases exponentially with $n_{\mathrm{val.}}$ when $\epsilon_S = 10^{-8}$.

The soft updating algorithm presented in this paper, and proposed originally by the authors in [2], produces the same results as Netica system [22] when hard evidence findings are given, and as the big clique algorithm [14] implemented in BC-Hugin, when soft evidence findings are given. An equivalent procedure for soft evidential update has also been independently proposed in [30] for dealing with agent communication, but evaluation or analysis of the algorithm are not provided.

The *big clique* algorithm applies the IPF procedure for soft updating to a peculiar *junction tree* where all the variables subjected to soft evidence are grouped in one of the cliques, namely the big clique. The junction tree is also known as Hugin architecture. It consists of the join tree plus separating registers between the cliques. The algorithm iterates only on a subset of the variables, the ones in the big clique. In this respect, it is more

Table 7
Approximate solutions for $P(A)$ in Ω_{is}

	a_1	a_2	a_3	a_4	a_5
$A = 3.5$	0	0.8833	0.1667	0	0
solution	0	0.8833	0.1667	0	0
$A = 4$	0	0.6667	0.3333	0	0
solution	0	0.6667	0.3333	0	0
$A = 4.5$	0	0.5	0.5	0	0
solution	0	0.512	0.488	0	0
$A = 5$	0	0.3333	0.6667	0	0
solution	0	0.5001	0.4999	0	0

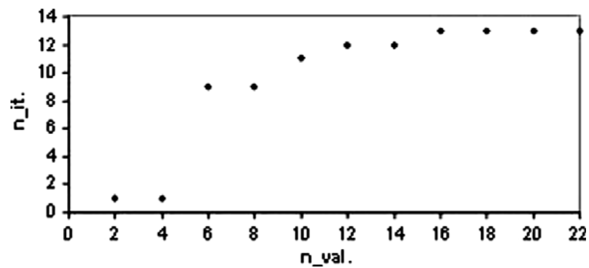

Fig. 6. Number of iterations in the Stud farm net when $\epsilon_S = 10^{-6}$.

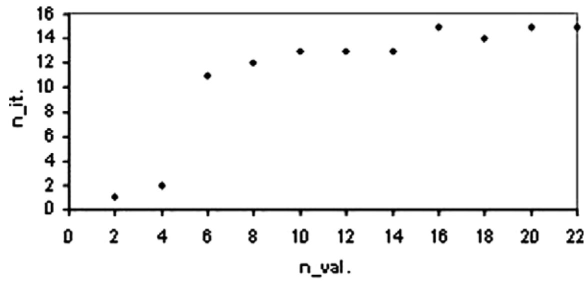

Fig. 7. Number of iterations in the Stud farm net when $\epsilon_S = 10^{-7}$.

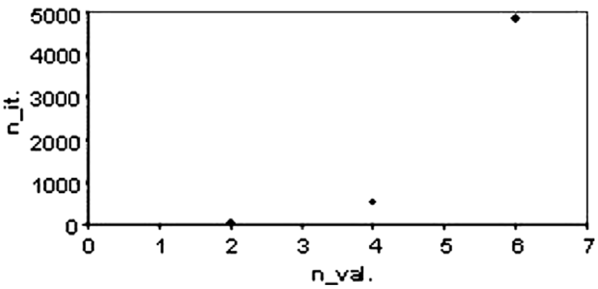

Fig. 8. Number of iterations in the Stud farm net when $\epsilon_S = 10^{-8}$.

efficient than the Algorithm 1. On the other hand, there are situations in which the use of Algorithm 1 is more convenient. There are two main drawbacks of the big clique algorithm: the junction tree has to be built every time the set of evidence variables changes (building an optimal junction tree is NP-hard [12]); if all the variables in the model are subjected to soft evidence, there is only one big clique (the computational complexity depends exponentially on the number of nodes in a clique). Hence the big clique algorithm is suitable for a restricted type of applications with a limited and fixed number of soft evidence variables.

An application of the IPF procedure has also been applied in [27] to modify, but not update, probability distributions represented as Bayesian networks.

5. Hybrid probabilistic temporal models

Temporal models or state-space models (SSMs), relate observations to unobserved states, using state and sensor variables replicated over time. They are suitable for sequential data modelling and dynamic systems. The same reasoning applies as well to static processes that have an underlying spacial dimension and hence can generate sequential data. A convenient approach to sequential data is on-line analysis, as it can deal with sequences with a variable length and requires storing less information. We shall however restrict our analysis to time-invariant and Markov models. We shall consider here hybrid *Dynamic Bayesian networks* (DBNs) [20,9] that are a type of representation of stochastic space-models, including both discrete and continuous variables, in order to solve state estimation problems and calculations of optimal paths. The soft updating algorithm presented in the previous section, is used as subroutine of the online inference for DBNs when dealing with continuous variables. An extended algorithm is presented for the estimation of the most likely explanation.

5.1. Soft filtering algorithm

Filtering is a type of inference in temporal models that estimates online the state of the world given the observations up to the current time. We shall consider filtering when soft findings occur due to continuous-valued observation variables, and call the soft updating algorithm for BNs as subroutine of a *soft filtering* for DBNs. Online inference in DBNs means that we can store a two-slice BN (2TBN) at a time, and we can deal with sequences which have a variable length. We shall exploit the simple approach of prediction–estimation cycle [29] suitable for general monitoring tasks.

Let Z_t be a set of variables representing the state of a system at time t, and (B_0, B_1) be the DBN modelling this system, where B_0 defines the prior $P(Z_0)$, and B_1 is a two-slice BN. Let $Z_t = (X_t, Y_t)$, where X_t is the set of nodes X_t^i called *state nodes* and Y_t is the set of nodes Y_t^j called *observation nodes*. The state nodes represent the variables termed *hidden states*, since they cannot be observed directly. The system allows to collect findings related merely to the observation nodes. Let us suppose to gather at each time step t some new continue-valued information about the system that can be expressed through a soft evidence function $E(Y_t)$. The basic idea of the prediction–estimation cycle is to store the information about the past in an updated prior $Bel(X_t)$ for the current state:

$$Bel(X_t) = P(X_t | E_0, E_1 \dots E_{t-1}) \tag{28}$$

Let $Pa(X_t)$ be the set of parents of X_t. The soft filtering algorithm at a generic time t consists of two phases:

– *prediction phase*: the expected probability distribution over the states X_t is computed using the estimated distribution over the past states X_{t-1}:

$$\widehat{Bel}(X_t) = \sum_{X_{t-1}} P(X_t | X_{t-1}) Bel(X_{t-1}) \tag{29}$$

$$\widehat{Bel}(X_t | Pa(X_t)) = \sum_{X_{t-1}} P(X_t | Pa(X_t)) Bel(X_{t-1}) \tag{30}$$

– *estimation phase*: the updated probability distribution over the states X_t, given the evidence $E(Y_t)$, is computed using the soft updating algorithm on the BN B_2, where B_2 is obtained from B_1 deleting the inter-slice topology and modifying the prior of X_t with the prediction $\widehat{Bel}(X_t)$ from the previous phase, as in (29) or (30).

The pseudo-code of the soft filtering is in Algorithm 2.

Algorithm 2. Soft filtering

$Z = (X, Y)$
(B_0, B_1)
if $t = 0$ **then**
  $Bel(X_0)\{P(X_0 | E(Y_0))$ from Algorithm 1$\}$
  $Bel(X_{t-1}) \Leftarrow Bel(X_0)$
else
  $\widehat{Bel}(X_t)$
  $Bel(X_t)\{P(X_t | E(Y_t))$ from Algorithm 1$\}$
  $Bel(X_{t-1}) \Leftarrow Bel(X_t)$
endif

When dealing with hard evidence, the Algorithm 2 is equivalent to the process of unfolding the 2TBN for t slices (i.e. up to the current time t) and applying a standard inference algorithm on the resulting structure. In the soft evidence case Algorithm 2 produces a different estimation of the state of the world than the soft inference algorithm applied for every time t, to the 2TBN stretched out t times. This is due to the fact that the soft inference algorithm performs differently if one considers a whole set of observations or one observation at a time. As shown in the previous section, the number of iterations of the inference process is related to the number of soft evidence findings. The iterative proportional fitting needs to consider sequentially each constraint and iterate over them the necessary number of times that lead to the convergency. No iteration is required, even when soft findings occur, only when the variables subjected to soft evidence are independent, which, in the case of the observation nodes in a DBN, does not occur unless the state variables are instantiated.

Soft inference on the unfolded structure provides the correct solution, however the two-slice filtering is more convenient since it needs to store only two slices at a time, best exploiting the richness of temporal models.

5.2. Soft abduction algorithm

Abduction or most likely explanation is the estimation of the most likely sequence of hidden states given the observations. We shall solve the abduction task in DBNs drawing on the soft updating algorithm. If no further assumptions are done, the most likely explanation does not necessary coincide with a sequence of local optimisations, i.e. optimisations obtained in a single time slice. We shall apply Bellman's principle of optimality for dynamic programming [3] to the propagation of probabilities in the join tree, similarly to the Viterbi algorithm [31].

Let (B_0, B_1) be a DBN modelling the temporal behaviour of a set of variables $Z_t = (X_t, Y_t)$, and $E(Y_{1:T})$ be the observed evidence during a period of time T. The 2TBN is unrolled T times.

Phase 1. Let us apply the soft updating algorithm for BNs to the T slices structure and obtain $P'(Clq_i) = P(Clq_i|E(Y_{1:T}))$ from Algorithm 1. Let us apply the π^{-1} message to $P'(Clq_i)$ as in (22) and obtain $P'(R_i|S_i)$.

Phase 2. Let $\pi_{\max}(S_i) = \max_{S_i} P'(R_i|S_i)$ for any clique Clq_i such that $S_i \subseteq Clq_1$. We define the function $\Psi^{\max}(Clq_i)$ as follows:

$$\Psi^{\max}(Clq_i) = P'(R_i|S_i)\pi_{\max}(S_i) \tag{31}$$

The application of (31) till the leaf cliques allows one to compute $\Psi^{\max}(Clq_i)$ for $2 \leqslant i \leqslant n_C$. $\pi_{\max}$ can be thought of as a message a clique sends downwards to its children in order to select the candidates for the optimal path.

Phase 3. Once we reach the leaf nodes, the configuration w^* of the nodes with the max value of $\Psi^{\max}$ is selected, and a backwards procedure is applied going up the tree. For a generic clique Clq_i, it is selected the configuration w_i^* holding the max value of $\Psi^{\max}$, and with the values of the sets S_j as selected in its children. The pseudo-code of the soft abduction is in Algorithm 3.

Algorithm 3 (*Soft abduction*).

```
({Clq_i}, Ψ), 1 ⩽ i ⩽ n_C
(S_i, R_i)
E(Y)
{Phase 1}
P''(Clq_i) ⇐ P(Clq_i|E(Y_k)) {from Algorithm 1}
Ψ''(Clq_1) ⇐ P''(Clq_1)
if C(Clq_i) ≠ ∅ then
   π^{-1}(S_j), S_j ⊆ Clq_i
end if
if Clq_i ← π^{-1}(S_i) {Clq_i receives a π^{-1} message from a parent Clq_j} then
   Ψ''(Clq_i) ⇐ P''(Clq_i)π^{-1}(S_i)
end if
{Phase 2}
if C(Clq_i) ≠ ∅ then
   π_max(S_j), S_j ⊆ Clq_i
end if
if Clq_i ← π_max(S_i) {Clq_i receives a π_max message from a parent Clq_j} then
   Ψ^max(Clq_i) ⇐ Ψ''(Clq_i)π_max(S_i)
end if
{Phase 3}
if C(Clq_i) = ∅ then
```

$w_i^* \Leftarrow \arg\max_{w \in Clq_i} \Psi^{\max}(w)$
end if
if $(Clq_j \in Pa(Clq_i)) \wedge (s_i^* \in w_i^*)$ **then**
$(w_j^* \Leftarrow \arg\max_{w \in Clq_j} \Psi^{\max}(w)) \wedge (s_i^* \in w_j^*)$
end if
$\cup_{i=1}^{n_C} w_i^*$ {the most likely configuration of nodes}

6. Conclusions

This paper has presented a possible model for hybrid Bayesian networks, where continuous and discrete variables are both present in any position in the graph. The inference algorithm is an extension of Lauritzen and Spiegelhalter's join tree algorithm, and it is able to process both continuous findings and soft evidence. The number of iterations for the convergence of the inference procedure depends on the number of values been instantiated. We have dealt with continuous variables using a fuzzy partition of continuous domains but no fuzzy logic formalism has been used throughout the paper. We have exploited the algorithm for soft evidence for filtering and abduction in hybrid temporal models. The online filtering has required some further investigation. As result of the IPF procedure, the online filtering in two-slice temporal BNs performs differently than applying soft inference on the unfolded structure.

In a future implementation of the inference algorithm for soft evidence we intend to consider other propagation methods like the HUGIN algorithm [13] or the lazy propagation algorithm [17].

Influence diagrams (IDs) and Dynamic Decision networks (DDNs) have not been considered in this paper. IDs and DDNs are an augmented version of Bayesian networks and Dynamic Bayesian networks with decision and utility nodes. They model situations where decisions are taken maximising the expected utility minus the cost. The results of this work could be exploited for IDs and DDNs. They are in particular suitable to represent agents that act and plan and for general control systems applications.

References

[1] J.F. Baldwin, Fuzzy and probabilistic uncertainty, in: S.C. Shapiro (Ed.), Encyclopaedia of Artificial Intelligence, vol. 1, Wiley, New York, 1992, pp. 528–537.
[2] J.F. Baldwin, E. Di Tomaso, Modified algorithm for fuzzy Bayesian networks inference, in: Proceedings of the IPMU Conference, Annecy, France, July 2002, pp. 33–38.
[3] R. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ, 1957.
[4] H. Chan, A. Darwiche, On the revision of probabilistic beliefs using uncertain evidence, Artificial Intelligence 163 (2005) 67–90.
[5] G.F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artificial Intelligence 42 (2–3) (1990) 393–405.
[6] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley, New York, USA, 1991.
[7] I. Csiszar, I-divergence geometry of probability distributions and minimization problems, The Annals of Probability 3 (1) (1975) 146–158.
[8] N. Friedman, M. Goldszmidt, Discretizing continuous attributes while learning Bayesian networks, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence, 1996.
[9] Z. Ghahramani, Learning dynamic Bayesian networks, in: C.L. Giles, M. Gori (Eds.), Adaptive Processing of Sequences and Data Structures, Springer-Verlag, Berlin, 1997, pp. 168–197.
[10] R.C. Jeffrey, The Logic of Decision, second ed., University of Chicago Press, Chicago, 1983.
[11] F.V. Jensen, An Introduction to Bayesian Networks, UCL Press, London, 1996.
[12] F.V. Jensen, F. Jensen, Optimal junction trees, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence, Seattle, Washington, July 1994.
[13] F.V. Jensen, S.L. Lauritzen, K.G. Olesen, Bayesian updating in causal probabilistic networks by local computations, Computational Statistics Quarterly 4 (1990) 269–282.
[14] Y.-G. Kim, M. Valtorta, J. Vomlel, A prototypical system for soft evidential update, Applied Intelligence 21 (2004) 81–97.
[15] K.B. Korb, A.E. Nicholson, Bayesian Artificial Intelligence, CRC Press, 2004.
[16] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their applications to expert systems, The Journal of the Royal Statistical Society 50 (2) (1988) 157–224.
[17] A.L. Madsen, F.V. Jensen, Lazy propagation in junction trees, in: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman Publishers, 1998, pp. 362–369.

[18] S. Moral, R. Rumí, A. Salmerón, Mixtures of truncated exponentials in hybrid Bayesian networks, in: ECSQARU'01: Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Springer-Verlag, London, UK, 2001, pp. 156–167.
[19] K. Murphy, A variational approximation for Bayesian networks with discrete and continuous latent variables, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence, 1999, pp. 467–475.
[20] K.P. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. thesis, University of California, Berkeley, 2002.
[21] R.E. Neapolitan, Probabilistic Reasoning in Expert Systems, Wiley, 1990.
[22] Norsys software corp. <http://www.norsys.com/netica.html>.
[23] H. Pan, L. Liu, Fuzzy Bayesian networks – a general formalism for representation, inference and learning with hybrid Bayesian networks, International Journal of Pattern Recognition and Artificial Intelligence 14 (2000) 941–962.
[24] H. Pan, D. McMichael, Fuzzy casual probabilistic networks – a new ideal and practical inference engine, in: Proceedings of International Conference on Multisensors Information Fusion, Las Vegas, USA, July 1998.
[25] J. Pearl, Fusion, propagation, and structuring in Bayesian networks, Artificial Intelligence 29 (3) (1986).
[26] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufman, San Francisco, CA, 1988.
[27] Y. Peng, Z. Ding, Modifying Bayesian networks by probability constraints, in: Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), AUAI Press, Arlington, Virginia, 2005, p. 459.
[28] E.H. Ruspini, A new approach to clustering, Information and Control 15 (1969) 22–32.
[29] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[30] M. Valtorta, Y.-G. Kim, J. Vomlel, Soft evidential update for probabilistic multiagent systems, International Journal of Approximate Reasoning 29 (2002) 71–106.
[31] A.J. Viterbi, Error bounds for convolutional codes and asymptotically optimum decoding algorithm, IEEE Transactions on Information Theory 13 (2) (1967) 260–269.
[32] J. Vomlel, Integrating inconsistent data in a probabilistic model, Journal of Applied Non-Classical Logics 14 (3) (2004) 367–386.
[33] L.A. Zadeh, Fuzzy sets, Journal of Information and Control 8 (1965) 338–353.