International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 18, No. 5 (2010) 539–564

DOI: 10.1142/S0218488510006696

BAYESIAN NETWORK REASONING WITH UNCERTAIN EVIDENCES

YUN PENG
University of Maryland Baltimore County, Computer Science and Electrical Engineering,
1000 Hilltop Circle, Baltimore, MD 21250, USA
ypeng@umbc.edu

SHENYONG ZHANG
University of Science and Technology of China, Hefei, Anhui 230026, China
University of Maryland Baltimore County, Computer Science and Electrical Engineering,
1000 Hilltop Circle, Baltimore, MD 21250, USA

RONG PAN
NexTag Inc., San Mateo, CA 94402, USA

Received 13 October 2009
Revised 12 June 2010

This paper investigates the problem of belief update in Bayesian networks (BN) with uncertain evidence. Two types of uncertain evidences are identified: *virtual evidence* (reflecting the uncertainty one has about a reported observation) and *soft evidence* (reflecting the uncertainty of an event one observes). Each of the two types of evidence has its own characteristics and obeys a belief update rule that is different from hard evidence, and different from each other. The particular emphasis is on belief update with *multiple* uncertain evidences. Efficient algorithms for BN reasoning with consistent and inconsistent uncertain evidences are developed, and their convergences analyzed. These algorithms can be seen as combining the techniques of traditional BN reasoning, Pearl's virtual evidence method, Jeffrey's rule, and the iterative proportional fitting procedure.

Keywords: Bayesian networks; belief update; probabilistic reasoning; uncertain evidence.

1. Introduction

This paper considers the problem of probabilistic reasoning with uncertain evidences. A regular evidence, called *hard evidence* in the literature, is an observation of a random variable, say X_i, having a particular value (or in a particular state), say a, represented as an instantiation $X_i = a$. However, it is not always possible to observe the value a variable is having in a particular case, or to have a complete trust on a claimed observation, thus bringing uncertainty to the evidences. This paper focuses on two types of uncertain evidences. The first type, called *soft evidence* as suggested by others,[19] can be interpreted as *evidence **of** uncertainty*, and is represented as a probability distribution of one or more variables. The second type, called *virtual evidence*, can be interpreted as *evidence **with** uncertainty*, and is represented as a *likelihood ratio*.[16] These two types of evidences reflect different kinds of uncertainty and each obeys a belief update rule that is different from hard evidence, and different from each other.

Based on an in-depth examination of these two types of uncertain evidences, we have developed efficient algorithms for belief update in Bayesian networks (BN) with such evidences. We focus on BN because of its popularity in intelligent systems and its time and space efficiency in representing and reasoning with probabilistic information.[15] However, many theoretical results we obtained hold for belief update of joint distributions that are not represented by BNs.

Related existing work can be found in Refs. 16, 19, 3, 20 and 21. Pearl was among the first to raise the issue of uncertain evidence and proposed the virtual evidence method.[16] However, as can be seen in Sec. 3, this method is not directly applicable to the situation in which multiple soft evidences are presented. Chan and Darwiche provided a thorough analysis that connects Pearl's virtual evidence method and Jeffrey's rule for both general joint distributions as well as BNs.[3] They also showed that a soft evidence can be converted into a virtual evidence, and as the result, belief update with a single soft evidence can be carried out by Pearl's virtual evidence method for both BN and joint distributions. They argued that multiple uncertain evidences should not be allowed for belief update at the same time. Vomlel, on the other hand, argued that multiple uncertain evidences, even if they are inconsistent with each other, should be allowed, and developed an algorithm, named GEMA, for such purpose.[21] However, GEMA was devised for general joint distributions, not for BNs. Valtorta *et al.* proposed to extend the iterative proportional fitting procedure (IPFP) for BN belief update with multiple consistent soft evidences.[19]

Our research extends these works in a number of significant ways. The results presented in this paper can be summarized as follows. (1) We formally established the equivalence of Jeffrey's rule, I-projection (a central operation of IPFP), and virtual evidence method, when dealing with a single uncertain evidence. We also established that Pearl's virtual evidence method works for multiple virtual evidences but not for multiple soft evidences. (2) We, for the first time, proved that I-projection and IPFP, which is known to minimize the I-divergence (or Kullback-Leibler distance), also minimizes the total variation between the source and the projected distributions. (3) We developed BN-IPFP, an efficient algorithm that combines Pearl's virtual evidence method and IPFP for BN belief update with multiple consistent soft evidences, and proved its convergence. (4) We developed SMOOTH, an algorithm for belief update with inconsistent soft evidences and proved its convergence for the case of two evidences. SMOOTH can be easily incorporated into BN-IPFP for BN update with inconsistent evidences.

The rest of the paper is organized as follows. Section 2 provides technical preliminaries with brief introductions to Jeffrey's rule, I-projection, and IPFP. Section 3 analyzes the two types of uncertain evidences. Section 4 develops two versions of algorithm BN-IPFP. Section 5 discusses issues related to inconsistent evidences and develops algorithm SMOOTH. Section 6 concludes with a discussion on evidential reasoning in which different types (hard, virtual, and soft) evidences are given either sequentially or at the same time, followed by the directions of future research.

For presentational clarity, proofs of theorems of our own (Theorems 2, 4, 6, 7) are given in the Appendix. We re-stated some theorems of others that are of immediate relevancy to this work, their proofs are referred to their original publications.

A number of computer experiments with artificial data were conducted to validate our results and to compare the performances with different methods. All experiments were run on an Intel® Core™2 CPU of 2.40G Hz and 2.0G maximum memory for the JVM (Java Virtual Machine). Netica[a] Java API and its junction tree based inference engine were used for standard BN inference.

2. Preliminaries

Throughout this paper, we use upper-case $X = (X_1, X_2,\ldots, X_n)$ for the set of all random variables of interest and X_i for individual random variables; lower-case x and x_i denote particular and arbitrary instantiation(s) of the respective variable(s); and bold upper-case $\mathbf{X}$, $\mathbf{X}_i$ denote the set of all possible instantiations. $Y, Y^1, Y^2, \cdots \subseteq X$ are for subsets of X, and y^j and $\mathbf{Y}^j$ for their instantiations similarly. Upper-case P, Q, R, S, T are reserved for probability distribution; $P(X)$ indicates a joint distribution; and $Q(Y^j)$ denotes the marginal distribution of $Q(X)$ over a subset of variables Y^j. Bold upper case $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{S}$, $\mathbf{T}$ are reserved for sets of distributions. In particular, $\mathbf{P}_{R(Y)} = \{P(X) \mid P(Y) = R(Y)\}$ denotes the set of all distributions over X whose marginals over $Y \subseteq X$ equal $R(Y)$.

2.1. *Jeffrey's rule and I-projection*

How to update a distribution $P(X)$ by another lower dimensional distribution $R(Y)$, $Y \subseteq X$, has been debated for a long time in the mathematics and philosophy communities.[12,16,3] One of the difficulties stems from the fact that the Bayes' rule cannot directly apply here because $R(y)$, although acting as a condition for the update, itself is not an event. One approach proposed by R. Jeffrey[12] is based on two principles: the new, posterior distribution $Q(X)$ should 1) satisfy $R(Y)$ (i.e., $Q(Y) = R(Y)$) and 2) keep the conditional distribution of X, given $Y \subseteq X$, unchanged (e.g., $Q(X \backslash Y \mid Y) = P(X \backslash Y \mid Y)$). The second principle, known as *probability kinematics*, *has the effect of keeping the change in the update minimum*. Then for a given $R(Y)$ and $Z \subseteq X \setminus Y$, we can compute the probabilities

$$Q(z) = \sum_{y \in \mathbf{Y}} P(z \mid y) R(y) = \sum_{y \in \mathbf{Y}} \frac{P(z, y)}{P(y)} R(y) \tag{1}$$

where $y \in \mathbf{Y}$ indicates the summation is over all instantiations of Y. Equation (1) is known as Jeffrey's rule[12] or J-conditioning. From (1), let $Z \subseteq X \setminus Y$, then, for any y we have the updated distribution

[a]Netica: Bayesian network tool from Norsys Software Corp. http://www.norsys.com/

$$Q(x) = \frac{P(x, y)}{P(y)} R(y) = \begin{cases} P(x)\dfrac{R(y)}{P(y)} & \text{if } P(y) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Two functions have been used widely to measure the distance or difference between two distributions over *X*. Their definitions are given below.

Definition 1.[20] The *I-divergence* (also known as Kullback-Leibler distance and relative entropy) between $P(X)$ and $Q(X)$ is given by

$$I(P \| Q) = \begin{cases} \sum_{P(x)>0} P(x) \log \dfrac{P(x)}{Q(x)} & \text{if } P << Q \\ +\infty & \text{otherwise} \end{cases} \tag{3}$$

where $P << Q$, denoting P is *dominated* by Q, if $\{x \mid P(x) > 0\} \subseteq \{x \mid Q(x) > 0\}$.

Note that $I(P \| Q) \geq 0$ for all P and Q, the equality holds only if $P = Q$. Also note that in general $I(P \| Q) \neq I(Q \| P)$, so I-divergence is not a true "distance" metric.

Definition 2. The *total variation* between $P(X)$ and $Q(X)$ is defined as

$$\delta(P, Q) = \sum_{x \in \mathbf{X}} | P(x) - Q(x) | \tag{4}$$

Now we define I-projection, one of the central concepts for our work.

Definition 3. $Q(x)$ is said to be an *I-projection*[b] of $P(x)$ on a convex set of distributions **S** if

$$I(Q \| P) = \min_{\tilde{Q} \in \mathbf{S}} I(\tilde{Q} \| P) \tag{5}$$

It has been shown that because of the convexity of **Q** I-projection is unique.[6] We are particularly interested in I-projections on $\mathbf{P}_{R(Y)}$, the set of distributions whose marginals over Y equal $R(Y)$. $\mathbf{P}_{R(Y)}$ is known to be convex and the I-projection of $P(x)$ on $\mathbf{P}_{R(Y)}$ can be calculated by[20]

$$Q(x) = \begin{cases} P(x) \cdot \dfrac{R(y)}{P(y)} & \text{if } P(y) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Note that (6) is exactly the same as (2). This proves the following theorem.

Theorem 1. *Let $Q(X)$ be the distribution resulted from updating $P(X)$ by $R(Y)$, $Y \subseteq X$ using Jeffrey's rule of* (2). *Then $Q(X)$ is the I-projection of $P(X)$ on $\mathbf{P}_{R(Y)}$.*

[b]I-projection defined here is also called I_1-projection in the literature. Since I-divergence is not symmetric, another projection, namely, I_2-projection Q' on **Q** is defined that minimizes the I-divergence $I(P \| Q')$. Unlike I_1-projection, I_2-projection in general is not unique. In this paper, all I-projections refer to I_1-projections.

Next we show that I-projection by (6) not only minimizes the I-divergence, but also the total variation.

Theorem 2. *Let* $Q(X)$ *be the I-projection of* $P(x)$ *on* $\mathbf{P}_{R(Y)}$. *Then* $\delta(P,Q) = \min_{\tilde{Q}\in\mathbf{P}_{R(Y)}} \delta(P,\tilde{Q})$.

2.2. *IPFP*

For a single constraint $R(Y)$, the I-projection of $P(X)$ on $\mathbf{P}_{R(Y)}$ finds a distribution that satisfies this constraint and is closest to $P(X)$ (in terms of I-divergence), provided $R(Y) << P(Y)$. Iterative proportional fitting procedure (IPFP) extends this idea to modify $P(X)$ with multiple constraints by continuously projecting the distribution resulted from the previous iteration to $\mathbf{P}_{R(Y^j)}$ of the next constraint $R(Y^j)$. This procedure is formally defined as follows.

Definition 4.[20] Let $\mathbf{R} = (R(Y^1),\cdots R(Y^m))$ be a set of constraints and $Q_0(X)$ the initial distribution. Then for $k = 1,2,\cdots$, $j = 1+(k-1) \bmod m$, and $R(Y^j) << Q_{k-1}(Y^j)$ for all *k, j*, IPFP is defined by

$$Q_k(x) = \begin{cases} Q_{k-1}(x)\cdot\dfrac{R(y^j)}{Q_{k-1}(y^j)} & \text{if} \quad Q_{k-1}(y^j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

In (7), m is the number of constraints, k is the iteration index, and j determines the constraint used at step k. For clarity, in the rest of this paper, we write (7) as

$$Q_k(x) = Q_{k-1}(x)\cdot\frac{R(y^j)}{Q_{k-1}(y^j)} \tag{7-1}$$

with the understanding that $Q_k(x) = 0$ when $Q_{k-1}(y^j) = 0$.

IPFP first appeared in the literature in Ref. 13, and shortly after was used as a procedure to estimate cell frequencies in contingency tables under some marginal constraints.[8] IPFP was extended in Refs. 1 and 5 to also allow conditional distributions as constraints (conditional or C-IPFP). The convergence of IPFP was studied in Refs. 7, 10, and 17 with proofs under different conditions, the convergence of C-IPFP can be found in Ref. 5. For our purpose, we cite a result from Ref. 20 in the theorem below, which is based on the I-divergence geometry developed in Ref. 7.

Theorem 3. *Let* $\mathbf{R} = (R(Y^1),\cdots R(Y^m))$ *be a set of constraints. If* $\mathbf{S} = \cap_{j=1}^{m}\mathbf{P}_{R(Y^j)} \neq \varnothing$, *then IPFP of* (7) *converges and the converging distribution* $Q^*(X)$ *is the I-projection of* $Q_0(X)$ *on* $\mathbf{S}$.

If $\mathbf{S} \neq \varnothing$, these constraints are said to be *consistent* with each other, and each distribution in $\mathbf{S}$ satisfies all constraints in $\mathbf{R}$. Therefore, at convergence, $Q^*(X)$, as the

I-projection on **S**, has the minimum I-divergence among those that satisfy all constraints in **R**. Next we show that IPFP also minimizes the total variation in the next two theorems.

Theorem 4. *Consider an initial distribution* $Q_0(X)$ *and a set of consistent constraints* $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$. *Let* $Q^*(X)$ *be the converging distribution when applying IPFP on* $Q_0(X)$ *using constraints in* **R**, *let* $Y = Y^1 \cup Y^2 \cup \cdots \cup Y^m$ *and* $Q^*(Y)$ *be the converging distribution when applying IPFP on* $Q_0(Y)$ *using constraints in* **R.** *Then*

$$Q^*(x) = Q_0(x)\frac{Q^*(y)}{Q_0(y)}. \tag{8}$$

Comparing Theorem 4 and (7-1), IPFP on $Q_0(X)$ with *m* constraints is equivalent to modifying $Q_0(X)$ by a single constraint $Q^*(Y)$. That is, $Q^*(X)$ is the I-projection of $Q_0(X)$ on $\mathbf{P}_{Q^*(Y)}$. This, together with Theorem 2, leads to the following theorem.

Theorem 5. *Let* $Q^*(X)$ *be the converging distribution using IPFP with an initial distribution* $Q_0(X)$ *and a set of constraints* $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$ *with* $\mathbf{S} = \cap_{j=1}^m \mathbf{P}_{R(Y^j)} \neq \varnothing$. *Then* $\delta(P, Q^*) = \min_{\tilde{Q} \in \mathbf{S}} \delta(P, \tilde{Q})$.

To the best of our knowledge, Theorems 2, 4, and 5 are original results which have not been reported in the literature before.

IPFP bears a great resemblance with another family of procedures known as *alternating projection*, which finds a point in the intersection of several convex sets by a sequence of projections onto these sets. Alternating projection has been widely used as an optimization method in areas of sampling theory, signal processing, and neural networks. A comprehensive review of this method can be found in Ref. 4. The difference from IPFP is that alternating projection is primarily for Euclidean spaces and it tends to minimize the square distances while IPFP is for probability spaces and it minimizes I-divergence (and the total variation by our result in Theorem 5) but not the square distances.[10] Several IPFP-based algorithms we will discuss, especially those for inconsistent evidences, can find their counterparts in alternating projection procedures.

3. Uncertain Evidences

Evidences presented for belief update may be uncertain for various reasons. A reported observation may not be totally trusted due to errors or noise in the observation or reporting process; it may be biased due to the observer's preference; it may not hold when the time or location is different. Among all types of uncertain evidences, this paper concentrates on two of them, named *virtual evidence* and *soft evidence*.

3.1. *Virtual evidences*

Pearl[16] proposed the virtual evidence method to deal with BN belief update when one is uncertain about a *claim* of a hard evidence (i.e., an event), say, $X_i = a$. Suppose we believe with probability *p* that this claim is actually due to the occurrence of $X_i = a$, then the probability it is not occurring is $1 - p$. The virtual evidence method requires this uncertainty information be given as a likelihood ratio $L(X_i) = p : (1-p)$, not necessarily

the specific probabilities. To reason with virtual evidence in a BN, Pearl's method extends the given BN by creating a binary *virtual* node, U with state u standing for the event that $X_i = a$ is claimed to have occurred. The virtual node U has X_i as its only parent and its conditional probability table (CPT) satisfies $P(u \mid X_i = a) : P(u \mid X_i \neq a) = L(X_i)$. Then the belief update (with the claimed observation and the uncertainty about this claim in the form of the likelihood ratio L) can be done by instantiating U to u (i.e., treating u as a hard evidence). Many BN engines accept a likelihood ratio as input for the update without explicitly introducing the virtual node.

This method is generalized in Ref. 3 to any arbitrary set of mutually exclusive and exhaustive events and the associated likelihood ratio, and from BN to any joint distributions. Under this generalization, virtual evidence on $Y \subseteq X$ is represented as a likelihood ratio

$$L(Y) = P(ob(y_{(1)}) \mid y_{(1)}) : P(ob(y_{(2)}) \mid y_{(2)}) : \cdots : P(ob(y_{(s)}) \mid y_{(s)}),$$

where $y_{(1)}, y_{(2)}, ..., y_{(s)} \in \mathbf{Y}$ are all instantiations of Y, $ob(y_{(i)})$ denotes the event that we observed $Y = y_{(i)}$ is *True*, and $P(ob(y_{(i)}) \mid y_{(i)})$ is interpreted as the probability we observe $Y = y_{(i)}$ if Y is indeed in state $y_{(i)}$.

3.2. *Soft evidences*

Soft evidence, named by Valtorta,[19] is given as a distribution $R(Y)$, $Y \subseteq X$. This kind of evidence can be seen in many places. For example, one may not be able to observe the precise state of a variable for a given case but may know its distribution. Also sometimes it is more important to know the distribution of a variable than its precise state at a given moment. When two BNs (or some other data and knowledge sources of probabilistic or statistical nature) interact with each other, the information exchanged between them is often in the form of probability distributions of shared variables.

For a given soft evidence, say $R(X_i)$, even though we are uncertain about the specific state X_i is in, we are certain about its distribution. In other words, $R(X_i)$ is a true (and certain) observation, and this distribution should be preserved in the updated joint distribution Q^* (i.e., $Q^*(X_i) = R(X_i)$). In this sense, soft evidences should be treated the same as hard evidence. In fact, a hard evidence, say $X_i = a$, is a special case of soft evidence ($R(X_i = a) = 1, R(X_i = b) = 0$ for all states $b \neq a$).

As suggested in Ref. 3, Jeffrey's rule of (2) is a natural choice for updating a joint distribution $P(X)$ by a soft evidence $R(Y \subseteq X)$ because the updated distribution preserves $R(Y)$ while making minimum changes to the original distribution. However, Jeffrey's rule cannot directly apply when the joint distribution is represented as a BN. This can be overcome by converting a soft evidence to a virtual evidence, as suggested by Ref. 3.

Consider a distribution $P(X)$ and a soft evidence $R(Y), Y \subseteq X$. All possible instantiations of Y, $y_{(1)}$, $y_{(2)}, \cdots, y_{(l)} \in \mathbf{Y}$, form a mutually exclusive and exhaustive set of events. $R(Y)$ then can be converted to a virtual evidence with the likelihood ratio

$$L(y) = \frac{R(y_{(1)})}{P(y_{(1)})} : \frac{R(y_{(2)})}{P(y_{(2)})} : \cdots : \frac{R(y_{(l)})}{P(y_{(l)})} \tag{9}$$

As shown in Theorem 5 of Ref. 3, when this virtual evidence is applied to $P(X)$, the new distribution is exactly the same as the one obtained by applying $R(Y)$ using the Jeffrey's rule of (2).

3.3. *Multiple uncertain evidences*

Like hard evidences, multiple uncertain evidences can arrive at the same time or in a sequence. There is no problem for belief update by multiple virtual evidences, because what is required is that the updated distribution preserves the given likelihoods. Update can be done by simply treating each virtual evidence as a hard evidence on the virtual node and instantiating that node to true. Note that, since a virtual node U is independent of all other virtual nodes, given the parent of U (i.e., they are d-separate), the likelihood ratio reflected on U will not be affected by the belief update operations with other virtual (and hard) evidences.

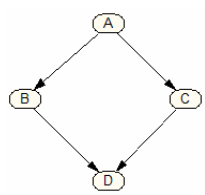
However, this is not the case when updating by two soft evidences $se1 = R(Y^1)$ and $se2 = R(Y^2)$. To satisfy both $se1$ and $se2$, the updated distribution Q is required to have its marginals $Q(Y^1) = R(Y^1)$ and $Q(Y^2) = R(Y^2)$. Update cannot be done by first converting $se1$ and $se2$ to two virtual evidences and then applying the virtual evidence method with these two virtual evidences. This is because, after applying the first evidence, there is no way to hold $Q(Y^1) = R(Y^1)$ when the second evidence is applied. Furthermore, as can be seen in the example below, when the soft evidences are presented in different orders or altogether, different update results will be generated. This problem, known as the *commutativity* of iterated revisions, has been viewed as a problem for Jeffrey's rule.[3,22]

Example 1. As depicted in Fig. 1, we are given a BN of four binary variables *A, B, C*, and *D* and two soft evidences $se1$: $R(B) = (0.7, 0.3)$ and $se2$: $R(C) = (0.3, 0.7)$. To convert them to virtual evidences, we first compute from the BN the marginals $P(B) = (0.44, 0.56)$ and $P(C) = (0.45, 0.55)$, then compute the likelihood ratios by (9) as $L(B) = 0.7/0.44{:}0.3/0.56 = 1.5909{:}0.5357$ and similarly $L(C) = 0.6667{:}1.2727$.

A	
1	**0**
0.4	0.6

A	B	
	1	**0**
1	0.20	0.80
0	0.60	0.40

A	C	
	1	**0**
1	0.60	0.40
0	0.35	0.65



B	C	D	
		1	**0**
1	**1**	0.10	0.90
1	**0**	0.85	0.15
0	**1**	0.45	0.55
0	**0**	0.70	0.30

Fig. 1. An example BN of 4 variables.

As can be seen in rows 2 and 3 of Table 1 below, when the two virtual evidences are applied separately, the updated beliefs satisfy the corresponding *se*1 and *se*2 (belief on $B = 1$ and $C = 1$ are updated to 0.7 and 0.3, respectively). Rows 4 and 5 show the update results when these two virtual evidences applied together and in a sequence, respectively. It is not surprise that the results are the same, since, as mentioned earlier, belief update with multiple virtual evidences are equivalent to belief update with multiple hard evidences of the virtual evidence nodes. Let $U1$ and $U2$ be the two virtual evidence nodes. It can be verified that $P(u1|B = 1, u2):P(u1|B = 0, u2) = L(B)$ and $P(u2|C = 1, u1):P(u2|C=0, u1) = L(C)$, i.e., the likelihood ratios are preserved when the other evidence is presented. However, as can be seen in Rows 4 and 5, none of these two *soft* evidences is satisfied by the resulting distributions. To deal with this problem, one may suggest that, before applying *se*2, we first recalculate a new likelihood ratio $L'(C)$ for *se*2 using the distribution updated by *se*1 (Row 2). By (9), we have $L'(C) = 0.3/0.425:0.7/0.575 = 0.7.59:1.2174$. Row 6 shows the update result where *se*2 is satisfied but belief on $B = 1$ is moved away from what is required by *se*1 (from 0.700 to 0.710).

Table 1. Belief update on BN of Example 1.

Evidences	Belief on $B = 1$	Belief on $C = 1$
1. original	0.440	0.450
2. using $L(B)$	**0.700**	0.425
3. using $L(C)$	0.455	**0.300**
4. $L(B)$ and $L(C)$	*0.712*	*0.279*
5. $L(B)$ then $L(C)$	*0.712*	*0.279*
6. $L(B)$ then $L'(C)$	*0.710*	**0.300**

Some argued based on the "All things considered" interpretation of soft evidence, that belief update with such evidences should not be commutative.[3] In contrast, we argue that soft evidences are true observations of distributions of some events, and as such, they all should be preserved in the updated "posterior" distribution; also that, if one or more such distributions exist, the one with the minimum I-divergence to the original distribution can be found by IPFP, using these evidences as constraints. However, IPFP works on full joint distributions, and thus is not directly applicable to belief update in BNs. In the next section, we develop algorithm BN-IPFP for BN belief update with multiple soft evidences. This algorithm first converts all soft evidences to virtual evidence form and then iterates in IPFP style to update the BN until it settles down to a distribution that satisfies all given soft evidences.

Another issue that arises with multiple soft evidences is that these evidences may not be consistent with each other, i.e., there is no distribution that satisfies all given evidences. This problem is dealt with in Sec. 5.

4. BN-IPFP

The problem is stated as follows. We are given a BN on variables $X = (X_1, X_2,\ldots, X_n)$ with the joint probability $P(X) = \Pi_{X_i \in X} P(X_i \mid \pi_i)$, where $P(X_i \mid \pi_i)$ is the CPT for

variable X_i, and a set of soft evidences $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$ where $Y_1, Y_2, \cdots, Y_m \subseteq X$. Suppose the constraints in **R** 1) are consistent, and 2) satisfy the dominance condition: for all $R(Y^j) \in \mathbf{R}$, $P(Y^j) << R(Y^j)$. Then the belief update of the given BN by **R** is to find $Q^*(X)$ which 1) satisfies all evidence in **R**; and 2) has minimum I-divergence to $P(X)$.

For small BNs, one can explicitly generate the full joint distribution $P(X)$ from the given BN and then apply IPFP using the soft evidences in **R** as constraints to update the distribution. This, however, is infeasible for large BN, because the distribution would be prohibitively large and IPFP would be computationally extremely expenses as it needs to literally modify each entry of the joint distribution table in each iteration. To address this problem, Valtorta, Kim and Vomlel have devised a variation of Junction-Tree (JT) algorithm based on IPFP[19] that utilizes the interdependencies captured in the BN structure. One version of this algorithm works in situation where all variables in each Y^j are contained in one clique C^j in the JT. Then the belief update goes iteratively over the evidences in cycle: in each iteration, $Q(C^j)$ is updated by the corresponding $R(Y^j)$ and then the change of $Q(C^j)$ is propagated to the rest of the JT by the regular JT method. The general situation where a soft evidence may involve variables in more than one cliques is dealt with by another version called *big clique* algorithm. In this algorithm, when constructing the JT, all soft evidence nodes (i.e., those variables that are involved in any of the soft evidences) are fully connected with each other by additional undirected edges. After triangulation, all soft evidence nodes appear in a single clique (the *Big Clique*). The belief update is done by first updating the big clique using all evidences in **R** by running IPFP to convergence and then propagating the resulting distribution of this clique to the rest of the JT. The Big Clique algorithm becomes time and space inefficient when the size of the big clique itself becomes large. Both versions are shown to converge and the converging joint distribution satisfies all evidences in **R**, provided these constraints are consistent to each other.

One limitation with these JT based belief update algorithms is that they cannot be easily adopted by those using inference mechanisms other than JT. Also, they require incorporating IPFP operations into the JT procedure, causing re-coding of the existing JT inference engine. The authors of Ref. 19 mentioned the possibility of implementing the first version of their algorithm as a wrapper around Hugin shell or other JT engines, but no suggestion of how this can be done was given.

To address these issues, we propose two new algorithms for BN inference with multiple soft evidences. Both algorithms utilize IPFP, although in quite different ways. The first algorithm combines the idea of IPFP and the encoding of soft evidence by virtual evidence of (9). The second algorithm is based on Theorem 4, it is similar to the Big Clique algorithm but it *decouples* the IPFP from JT (or any specific BN inference engine). These two algorithms are presented in the next two subsections.

4.1. *BN-IPFP-1*

As shown earlier, although a single soft evidence can be applied to BN belief update by first converting it to a virtual evidence, this approach does not work with multiple

evidences. As can be seen in Example 1 at the end of last section, after updating by se_2, the distribution no long satisfies se_1. What is needed is a method that can convert soft evidences in **R** to one or more likelihood ratios which, when applied as virtual evidences to the BN, preserve marginal distributions specified in every soft evidence.

Algorithm BN-IPFP-1 presented below accomplishes this by combining the idea of IPFP and the virtual evidence method. Roughly speaking, this algorithm goes as follows. Like the IPFP, it is an iterative process, starting with $Q_0(X) = P(X)$, and one soft evidence $R(Y^j)$ is considered at each iteration. If the marginal $Q_{k-1}(Y^j)$ of the current distribution equals $R(Y^j)$, then it does nothing; otherwise, a new virtual evidence (in the form of a likelihood ratio) is created based on the current $Q_{k-1}(Y^j)$ and $R(Y^j)$ according to (9) and applied to modify $Q_{k-1}(Y^j)$. The algorithm is given below.

Algorithm BN-IPFP-1. Consider a BN with prior distribution $P(x)$, and a set of m consistent soft evidences $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$. We use the following iterative procedure for belief update:

1. $Q_0(X) = P(X)$; $k = 1$;
2. Repeat the following until convergence;
   2.1 $j = 1 + (k-1) \bmod m$; $l = 1 + \lfloor (k-1)/m \rfloor$;
   2.2 construct virtual evidence with likelihood ratio
   $$L_{j,l}(Y^j) = \frac{R(y^j_{(1)})}{Q_{k-1}(y^j_{(1)})} : \frac{R(y^j_{(2)})}{Q_{k-1}(y^j_{(2)})} : \ldots : \frac{R(y^j_{(j_s)})}{Q_{k-1}(y^j_{(j_s)})}$$
   where $y^j_{(1)}, y^j_{(2)}, ..., y^j_{(j_s)} \in \mathbf{Y}^j$ are state configurations of Y^j;
   2.3 Obtain $Q_k(X)$ by updating $Q_{k-1}(X)$ with $L_{j,l}(Y^j)$ using Pearl's virtual evidence method;
   2.4 $k = k + 1$;

The core of this algorithm is Step 2.2, which adds a new virtual evidence with likelihood ratio $L_{j,l}(y^j)$ where the second subscript, l, is the number of virtual evidences created for $R(y^j)$, incremented for every m iterations. Note that the sequence of likelihood ratios for each $R(Y^j)$ can be cumulated as a single one $L_j(Y^j) = \Pi_l L_{j,l}(Y^j)$.

4.2. *BN-IPFP-2*

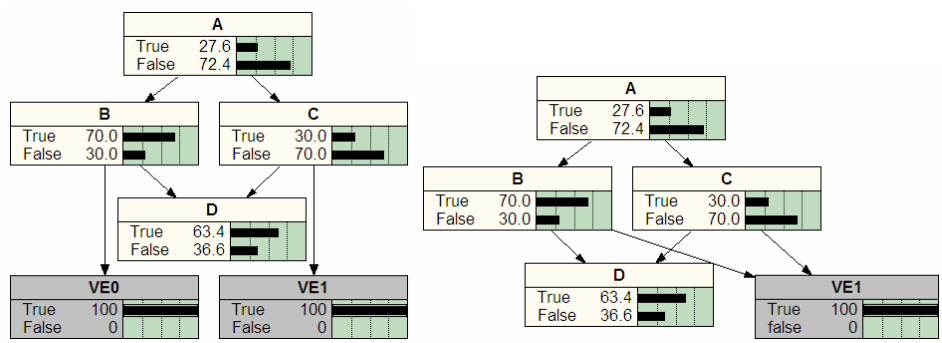
BN-IPFP-1 may become expensive when the given BN is large because it computes the marginal $Q_{k-1}(Y^j)$ (Step 2.2) and updates the beliefs of the entire BN (Step 2.3) in each iteration. Algorithm BN-IPFP-2 avoids repeated BN computation by first constructing a single virtual evidence node from the marginal of $P(Y)$, where Y contains all variables in all of the given soft evidences, and then updating the BN by this virtual evidence.

Algorithm BN-IPFP-2. Consider a BN with prior distribution $P(X)$, and a set of m consistent soft evidences $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$. Let $Y = Y^1 \cup Y^2 \cup \cdots \cup Y^m$. We use the following procedure for belief update:

1. Use any BN inference method to obtain $P(Y)$ from $P(X)$.
2. Update $P(Y)$ by IPFP using $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$ as constraints until converging to $Q^*(Y)$.
3. Construct a virtual evidence with likelihood ratio $L(Y)$ computed from $Q^*(Y)$ and $P(y)$ by (9).
4. Apply $L(Y)$ as a single virtual evidence to update $P(X)$.

The convergence and correctness of both BN-IPFP algorithms are established in Theorem 6 below.

Theorem 6. *If soft evidences in* $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$ *are consistent with each other and* $P(Y^j) << R(Y^j)$ *for all* $R(Y^j) \in \mathbf{R}$, *then both algorithms BN-IPFP-1 and BN-IPFP-2 converge to the same distribution, which is the I-projection of P(X) on* $\mathbf{S} = \cap_{j=1}^{m} \mathbf{P}_{R(Y^j)}$.



(a) BN-IPFP-1 (b) BN-IPFP-2

Fig. 2. Running results of Example 1 with BN-IPFP-1 and 2.

Figure 2 shows the running results of BN-IPFP-1 and 2 for the example BN given in Fig. 1. The two virtual evidence nodes VE0 and VE1 in Fig. 2 (a) are generated by BN-IPFP-1 for the two soft evidences $R(B)$ and $R(C)$; the virtual evidence VE1 In Fig. 2 (b) is created from $R(B)$ and $R(C)$ according to BN-IPFP-2. Both algorithms converge in 4 iterations to the same distribution that satisfies both constraints $R(B)$ and $R(C)$.The final combined likelihood ratios at convergence are $L^*(B)$ = (1.0:0.354) and $L^*(C)$ = (0.578:1.0) for BN-IPFP-1 and $L^*(B, C)$ = (0.578:1.0: 0.205:0.354) for BN-IPFP-2.

4.3. *Time and space performance*

The iterations of BN-IPFP-1, BN-IPFP-2 and Big Clique algorithm all converge to the same distribution. At each iteration, Big Clique algorithm updates beliefs of the joint

probabilities of the big clique C, BN-IPFP-2 updates the joint distribution of Y, and BN-IPFP-1 updates the belief of the whole BN, i.e., all variables in X. Clearly, $Y \subseteq C \subseteq X$. However, the time complexity for one iteration of Big Clique is $O(2^{|C|})$, and $O(2^{|Y|})$ for IPFP because both require modifying a joint distribution table. On the other hand, the time complexity of BN-IPFP-1 is equal to the complexity of the BN inference algorithm it uses for belief update, which is often more efficient than modifying the joint distribution. For example, if we use JT, the time complexity for one iteration of BN-IPFP-1 is exponential to the size of the largest clique in JT of the original BN, which may be smaller than C and Y, especially for sparse BNs.

Both Big Clique and BN-IPFP-2 are space inefficient, they need exponential space for the joint potential of C, and the joint distribution of Y, respectively. In contrast, BN-IPFP-1 only needs additional space for virtual evidence, which is $O(\Sigma_{j=1}^{m} 2^{|Y_j|})$. BN-IPFP-2 is thus more suitable for problems with a large BN but a small number of soft evidence variables and BN-IPFP-1 is more efficient when the number of soft evidence variables is large. Also, both BN-IPFP-1 and 2 have the advantage that users do not have to stick to junction tree and modify the JT related procedures in the inference engine. They can be easily implemented as wrappers on any BN inference engines.

To empirically evaluate our algorithms and to get a sense of how expensive these two algorithms may be, we have conducted some experiments with artificially constructed BNs of different sizes and with different constraint sets. The reported memory consumption does not include those that was used by the JT-based inference engine of Netica, but the reported running time is the total running time.

Experiment 4-1 compares the algorithms' performance with varying number of soft evidences. It used a BN of 15 variables and three sets of 2, 4, 8 soft evidences each. One half of these evidences involved 2 variables, and the other half involved 1 variable. The experiment results are given in Table 2. It can be seen that, when the number of evidences increases, both the time and memory consumptions for BN-IPFP-1 increase at much slower rates than BN-IPFP-2.

Table 2. Experiment 4-1: performance with different numbers of soft evidences.

# of evidences	# Iterations		Exec. Time		Memory	
	BN-IPFP-1	BN-IPFP-2	BN-IPFP-1	BN-IPFP-2	BN-IPFP-1	BN-IPFP-2
2	24	14	0.57s	0.62s	590,736	468,532
4	79	23	0.63s	0.83s	726,896	696,960
8	95	17	0.71s	15.34s	926,896	2,544,536

Experiment 4-2 compares the algorithms' performances with different size of BN. Four BNs of 30, 60, 120, and 240 binary variables were used, each of which was updated by the same set of 4 soft evidences involving a total of 6 variables. For each algorithm, experimental runs for the four BNs were all converged after the same number of iterations (43 for BN-IPFP-1 and 14 for BN-IPFP-2).

Table 3. Experiment 4-2: performance of BN with different size.

Size of BN	# of Iterations		Exec. Time		Memory	
	BN-IPFP-1	BN-IPFP-2	BN-IPFP-1	BN-IPFP-2	BN-IPFP-1	BN-IPFP-2
30	43	14	0.58s	0.67s (0.64s)	721,848	691,042
60			0.71s	0.69s (0.66s)	723,944	691,424
120			1.71s	0.72s (0.66s)	726,904	691,416
240			103.1s	3.13s (0.72s)	726,800	696,842

From Table 3 we can see that when the number of soft evidences is fixed, the running time of BN-IPFP-2 increases slowly with the increase of the network size. Especially, the time for IPFP on $P(Y)$ (the time in parentheses) increases only slightly. This is because computing the single constraint $Q^*(Y)$ (Step 2) is the most expensive step in BN-IPFP-2 and Y is fixed. On the other hand, the execution time for BN-IPFP-1 increases at a much faster pace (roughly exponentially). This is because each iteration requires updating the entire BN. These experiments results confirm our theoretical analysis for the proposed algorithms.

5. Inconsistent Soft Evidences

A set of m soft evidences or constraints $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$ is said to be inconsistent if $\mathbf{S} = \cap_{j=1}^{m} \mathbf{P}_{R(Y^j)} \neq \varnothing$. Since there does not exist a distribution that satisfies all constraints in **R**, IPFP or methods based on IPFP such as those we developed in the previous section will not converge. Instead, the update will go into cycles around several distributions,[21] and the specific distributions it cycles around may be different, depending on the order the constraints are presented.[4] Several approaches to this problem based on IPFP have been suggested in the literature. A simple approach is to first run IPFP until it goes into a cycle of $Q_1^*(X), Q_2^*(X), \cdots, Q_m^*(X)$, each of which satisfies one of the given m constraints, and then take the average of these distributions $Q^*(X) = \Sigma_{j=1}^{m} Q_j^*(X)/m$ as the solution. Several disadvantages can be seen for this simple approach. The result may be different when these constraints are presented in different orders; there is not much we can say about $Q^*(x)$ except that it is somewhere in the middle of $Q_1^*(X), Q_2^*(X), \cdots, Q_m^*(X)$. Moreover, this approach is hard to apply to BN because it operates on full joint distributions.

Another approach modifies the IPFP of (7-1) as follows[20]:

$$Q_k(x) = (1-\alpha_k) Q_{k-1}(x) + \alpha_k Q_{k-1}(x) \cdot \frac{R(y^j)}{Q_{k-1}(y^j)} \tag{10}$$

where $0 < \alpha_k < 1$. This approach will be referred to as SR-IPFP, as it can be seen to be analogous to the *serial relaxation* method of alternating projection that can be used to find an approximate solution when the solution set **S** is empty (see Eq. (38) of Ref. 4). This method converges with constant $\alpha_k = \alpha$ when **R** is consistent; it converges when **R** is inconsistent if α_k gradually decreases toward 0. To allow each constraint to take its effect, α_k needs to start with a value very close to 1 and to decrease very slowly.

However, if the decreasing rate is too small, the convergence will take too many iterations; on the other hand, if the rate is too big, the process will be biased in favor of earlier constraints.

A more principled method was proposed in Ref. 21, named GEMA (Generalized EM Algorithm). GEMA assigns a weight w_j to each constraint $R(Y^j) \in \mathbf{R}$, $\Sigma_{j=1}^{m} w_j = 1$, which can be understood as the credibility one has for the evidence. The update is again an iterative process, and it takes two steps in each iteration. Take as an example, consider the k^{th} iteration that starts with $Q_{k-1}(X)$. In Step 1, it first uses (7-1) to compute m I-projections of $Q_{k-1}(X)$ to $\mathbf{P}_{R(Y^j)}$ for each $R(Y^j)$, denoted $\tilde{Q}_{k-1,j}(X)$, and then takes a weighted sum of these k I-projections to obtain a distribution $\tilde{Q}_{k-1}(X) = \Sigma_{j=1}^{m} w_j \tilde{Q}_{k-1,j}(X)$. In Step 2, GEMA first computes m marginals $\tilde{R}(Y^j) = \tilde{Q}_{k-1}(Y^j)$, then performs m steps of the standard IPFP on $Q_{k-1}(X)$ using these m $\tilde{R}(Y^j)$ as constraints to obtain $Q_k(X)$. Note that these new constraints are consistent with each other since they are marginals from the same distribution $\tilde{Q}_{k-1}(X)$. It has been shown that GEMA converges to a distribution which has a minimum I-aggregate Ψ, the weighted sum of I-divergences to all of the original constraints in **R**:

$$\Psi(Q(X) \mid R(Y^1), \cdots, R(Y^m) = \sum_{j=1}^{m} w_j I(R(Y^j) \| Q(y^K)) \cdot \tag{11}$$

GEMA can be seen as analogous to a parallel method of alternating projection that can be used to find an approximate solution when the solution set is empty (see Eq. (35) of Ref. 4). Our experiments (see Subsection 5.3) show that the time performance of GEMA is very sensitive to the data. For some combinations of $Q_0(X) = P(X)$ and **R**, it converges within a few hundreds of iterations, but for other combinations of similar size, millions of iterations are needed.

5.1. *Algorithm SMOOTH*

One thing in common for both GEMA and SR-IPFP of (10) is that both of them only modify the joint distribution $Q_{k-1}(X)$ while keeping the constraints unchanged through the iterations. Alternatively, one can make the modification *bi-directional*: at each iteration, not only the joint distributions are pulled closer to the constraints but also the constraints are pulled towards the joint distributions. By doing so, the inconsistency among the constraints is gradually reduced or *smoothened*, which may lead to a faster convergence. Based on this idea we developed our new method SMOOTH.

The procedure of SMOOTH consists of two phases. Phase 1 performs the standard IPFP using all of the original constraints in **R**. It stops when the process converges (for consistent constraints) or starts to cycle (for inconsistent constraints). Phase 2, executed only when cycle is detected at the end of Phase 1, differs from Phase 1 in that at each iteration, not only the current distribution $Q_{k-1}(X)$ is modified by the chosen constraint $R(Y^j)$, $R(Y^j)$ itself is also modified by $Q_{k-1}(X)$.

Specifically, we denote the modified constraints as $R_l(Y^j)$, with $R_0(Y^j) = R(Y^j)$ and $l = 1 + \lfloor (k-1)/m \rfloor$. At iteration k, first the constraint is modified by

$$R_l(Y^j) = \alpha R_{l-1}(Y^j) + (1-\alpha) Q_{k-1}(Y^j) \tag{12}$$

where $\alpha \in (0,1)$ is the *smooth factor* and it controls the speed of smoothing. From (12) we can see that the modified constraint $R_l(Y^j)$ is a mixture of the previous constraint $R_{l-1}(Y^j)$ and the marginal of the current distribution $Q_{k-1}(X)$. Since $Q_{k-1}(X)$ has been modified by all other constraints, (12) has the effect of pulling $\mathbf{P}_{R_l(Y^j)}$ closer to $\mathbf{P}_{R_l(Y^i)}, i \neq j$, thus reducing or smoothing the inconsistency among the constraints. To ensure that the smoothing is unbiased α should be chosen as very close to 1. Then Q_{k-1} is modified by the new constraint by

$$Q_k(x) = Q_{k-1}(x)\frac{R_l(y^j)}{Q_{k-1}(y^j)} \tag{13}$$

By (12), (13) can be rewritten as

$$\begin{aligned} Q_k(x) &= Q_{k-1}(x)\frac{R_l(y^j)}{Q_{k-1}(y^j)} \\ &= Q_{k-1}(x)\frac{\alpha R_{l-1}(y^j) + (1-\alpha)Q_{k-1}(y^j)}{Q_{k-1}(y^j)} \\ &= \alpha Q_{k-1}(x)\frac{R_{l-1}(y^j)}{Q_{k-1}(y^j)} + (1-\alpha)Q_{k-1}(x) \end{aligned} \tag{13-1}$$

Equation (13-1) is very similar to (10) for SR-IPFP. The different is that (10) always uses the original constraints while in (13-1) a changed constraint is used at each iteration. It is this difference that makes SMOOTH converges with constant α when the constraints are inconsistent. The algorithm SMOOTH is given below.

Algorithm SMOOTH. Consider an initial distribution $P(x)$ and a set of m soft evidences $\mathbf{R} = (R(Y^1), \cdots R(Y^m))$. SMOOTH consists of the following two phases:

Phase 1: do the standard IPFP using all constraints in **R** until it converges or goes into cycles;
if convergence is reached then exit;
Phase 2:
1. for $j = 1$ *to* m, $R_0(y^j) = R(y^j)$;
2. $k = 1$;
3. repeat the following until converging
   3.1 $j = 1 + (k-1) \bmod m$; $l = 1 + \lfloor (k-1)/m \rfloor$;
   3.2 $R_l(y^j) = \alpha R_{l-1}(y^j) + (1-\alpha)Q_{k-1}(y^j)$;
   3.3 $Q_k(x) = Q_{k-1}(x)\frac{R_l(y^j)}{Q_{k-1}(y^j)}$;
   3.4 $k = k + 1$;

Note that SMOOTH is exactly the same as the standard IPFP except that in Phase 2 it uses modified constraints, not the original one to update the current Q_k. This makes SMOOTH directly applicable to BN belief update in BN-IPFP style. The only thing that needs to be changed when applying SMOOTH to BN is to replace the operation of I-projection (Step 3.3 in Phase 2) by virtual evidence method of BN-IPFP-1 (Steps 2.2 and 2.3) of Sec. 4.

Next we investigate the convergence of SMOOTH.

5.2. *Convergence and performance of SMOOTH*

According to the algorithm, when the set of constraints is consistent, SMOOTH is reduced to the standard IPFP, and it converges in Phase 1. Next we discuss what happens when constraints are not consistent.

Figure 3 shows an example involving four constraints (m = 4) where $\mathbf{S}_j(X) = \mathbf{P}_{R(Y^j)}(X)$ is the set of all distributions whose marginal on Y^j equals $R(Y^j)$. At the end of Phase 1, a cycle (solid lines) is formed through $Q_{0,1}, Q_{0,2}, Q_{0,3}, Q_{0,4}$. In the first iteration of Phase 2, constraint $R_0(Y^1) = R(Y^1)$ is modified to $R_1(Y^1)$ by (12). This changes $\mathbf{S}_{0,1}(X)$ to $\mathbf{S}_{1,1}(X)$, which is closer to $Q_{0,4}$ than $\mathbf{S}_{0,1}(X)$. As the process continues, $\mathbf{S}_{j,l}(X)$ are moving closer to each other, and the cycles (dotted lines) formed by the resulting distributions become smaller and smaller until they merge into a single distribution.

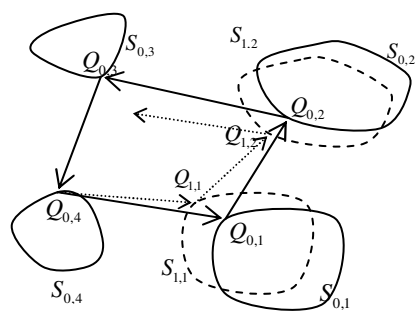


Fig. 3. Example showing the convergence of SMOOTH.

We formally establish the convergence of SMOOTH for m = 2 in the next theorem.

Theorem 7. *For an initial distribution P(X), two inconsistent soft evidences* $R(Y^1), R(Y^2)$, *and* $\alpha \in (0,1)$, *Phase 2 of SMOOTH converges.*

Experiments show that Phase 2 of SMOOTH converges for $m > 2$, and when $\alpha \to 1$ the converging distribution Q^* minimizes the sum of distances, in both I-divergence and total variation, to all constraints in **R**. We leave this general claim as a conjecture.

The time performance of SMOOTH, like all IPFP based methods, depends on the number of iterations it takes to reach convergence. Experiments show that SMOOTH moves towards the convergence point fairly fast at the beginning, even with α very close to 1. However, it slows down drastically at the end, forming a long and flat tail (see Fig. 4 where 90% of the time is spent to bring the flat tail to the convergence point). As discussed before, keeping α large at the beginning ensures information in the original constraints is not lost too soon by smoothing before it gets a chance to be absorbed. When the process gets closer to the convergence point, we can afford to use smaller α since most information of the original constraints that can be absorbed has largely been absorbed. By (12), a smaller α pulls the constraints toward the current Q_k faster, leading to a faster convergence at the end. We have experimented with a number of schedules for reducing α. The one performed best is the sigmoid function:

$$\alpha_k = \exp(A - k/B)/(1+\exp(A - k/B)) \tag{14}$$

where k is the iteration steps of Phase 2. It can be seen by (14) that with a large positive A, α is close to 1 at the beginning (k is small), and close to 0 when k becomes very large, and that α decreases very slowly at the two ends, but fast in the middle. Parameter A controls how long α_k is to remain large (longer for larger A) and B controls how fast α increases in the middle (faster for smaller B). If the desired initial value α_0 is given, then A can be determined by $\alpha_0 = 1/(1+\exp(A))$. For example, to have $\alpha_0 \approx 0.99$, we set $A = 4.595$.

We call SMOOTH using (14) to reduce α_k *Accelerated* SMOOTH (A-SMOOTH for short). Replacing α by α_k in (13-1), when $k \to \infty$, since $\alpha_k \to 0$, so $Q_k(x) \to Q_{k-1}(x)$, therefore, a-SMOOTH converges.

5.3. *Experiments*

To empirically validate algorithm SMOOTH and to get a sense of how well it performs in comparison to the existing methods, we have conducted computer experiments with different initial distributions and different constraints.

The algorithms compared in the experiments include: (1) GEMA, (2) SR-IPFP, (3) SMOOTH, (4) A-SMOOTH. For SR-IPFP, we use $\alpha_k = 1/(1+k)$ in (10), which is the fastest schedule for reducing α_k suggested by the authors.[20] For SMOOTH we set $\alpha \approx 0.99$ in Phase 2, and for A-SMOOTH, we set $A = 4.595$ and $B = 150$.

We use the number of I-projections instead of the number of iterations to measure the time performance of an algorithm because an iteration may involve different number of I-projections for different algorithms. For example, number of I-projections in one iteration is 1 for our SMOOTH and 2m for GEMA (m for each of the two steps).

In all our experiments, convergence is reached if at iteration $k = l \cdot m$ the sum of total variations $\Sigma_{j=1}^{m} |Q_{k+j}(y^j) - Q_{k+j-1}(y^j)|$ is within the given error bound of 10^{-12}.

Experiment 5-1 uses the data taken from Ref. 21 involving three variables X_1, X_2, X_3. The initial joint distribution JPD1 is a uniform distribution of the three variables. Three constraints, each a distribution of two variables, are generated according to the scheme in Table 4. These constraints are consistent with each other when $\varepsilon = 4/20$ (called CONS0), inconsistent when $\varepsilon = 3/20$ (called CONS1).

Table 4. Constraint generator.

$P_j, j = 1,2$	$X_{j+1} = 0$	$X_{j+1} = 1$
$X_j = 0$	$1/2 - \varepsilon$	ε
$X_j = 1$	ε	$1/2 - \varepsilon$
P_3	$X_3 = 0$	$X_3 = 1$
$X_1 = 0$	ε	$1/2 - \varepsilon$
$X_1 = 1$	$1/2 - \varepsilon$	ε

The experiment results for consistent constraints CONS0 are given in Table 5. All three algorithms converged to the same the I-projection on $\mathbf{S} = \mathbf{P}_{R(y^1)}(x) \cap \mathbf{P}_{R(y^2)}(x) \cap \mathbf{P}_{R(y^3)}(x)$. SMOOTH is significantly faster than the other two. This is because for the consistent constraints SMOOTH is reduced to the standard IPFP (only Phase 1 is executed).

Table 5. Experiment 5-1 results for CONS0 ($\varepsilon = 4/20$).

Algorithm	**GEMA**	**SR-IPFP**	**SMOOTH**
# projections	1164	3507	**84**
I-divergence	0.10453816	0. 10453816	0. 10453816

Experiment 5-2 compares performance with inconsistent constraints CONS1 in which every two constraints are consistent with each other, but they together are inconsistent with the third one. Besides JPD1, another initial joint distributions JPD2 is also used. The experiment results are given in Table 6 where for the two versions of SMOOTH numbers of I-projections for both phases are given. It can be seen from the I-divergences of the converging distributions to the initial distributions and the I-aggregates that GEMA, SMOOTH, and A-SMOOTH converge to distributions that are very close to each other, with A-SMOOTH significantly faster than the others (SR-IPFP was stopped when the time limit of 10 million I-projections is reached before the convergence).

Table 6. Experiment 5-2 results for Inconsistent CONS1 ($\varepsilon = 3/20$).

	# projections	**I-divergence**	**I-aggregate**
GEMA			
JPD-1	7,744,446	0.41502431	0.00367169
JPD-2	9,064,080	0.71979040	0.05727919
SR- IPFP			
JPD-1	>10,000,000	0.37048603	0.00461839
JPD-2	>10,000,000	0.70127029	0.05742945
SMOOTH			
JPD-1	177+3825	0.41503774	0.00367172
JPD-2	129+4899	0.71306584	0.05729201
A-SMOOTH			
JPD-1	177+375	0.41503891	0.00367227
JPD-2	129+402	0.71439294	0.05729532

We plot I-aggregates of all the four algorithms for JPD1 in Fig. 4. The plot starts at the 178th I-projection, which is the beginning of Phase 2 of SMOOTH and A-SMOOTH, and ends at the 4200th I-projection. It is clear that I-aggregate decreases fastest for A-SMOOTH, followed by SMOOTH, with CC-IPFP the slowest.

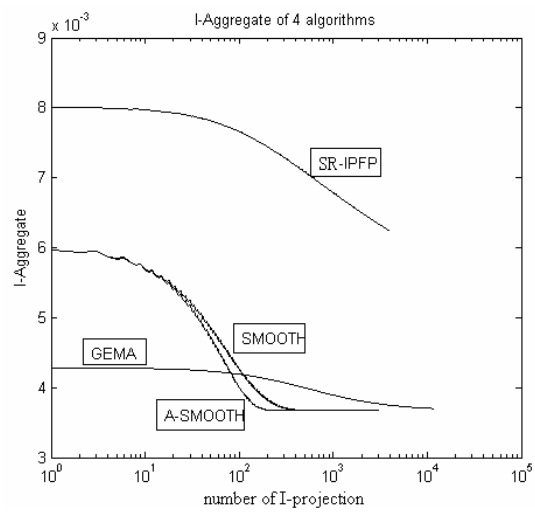


Fig. 4. Plot of I-aggregates of the four algorithms.

Experiment 5-3. To see that GEMA is data sensitive, we generated another set of 3 constraints (CONS2), each of which also involves two of the three variables X_1, X_2, X_3. Unlike CONS1 shown in Table 4, CONS2 is pair-wise inconsistent. The results using CONS2 against JPD1 and JPD2 are given in Table 7. It can be seen from Tables 6 and 7 that GEMA is very slow for three of the four combinations of JPDs and constraints but very fast (780 I-projections) for one combination (JPD1+CONS2). Similar phenomena have also been observed in some of our other experiments. On the other hand, both versions of SMOOTH have uniform performance for all combinations.

Table 7. Experiment 5-3 result for CONS2.

Algorithm	GEMA	SR-IPFP	SMOOTH
JPD1	780	>10,000,000	54+3405
JPD2	12,400,542	>10,000,000	216+3933

Experiment 5-4 tests the scalability of these algorithms with larger JPDs of 8 and 15 variables. The results shown in Tables 8 and 9 are consistent with those reported earlier for smaller JPD. For these experiments we did not run SR-IPFP because it took too much time to reach a point that was close to a convergence.

Table 8. Result for JPD of 8 variables and 4 inconsistent constraints.

	# projections	I-divergence	I-aggregate
GEMA	912	0. 93720845	0.02345286
SMOOTH	48+5000	0.94365473	0.02347122
A-MOOTH	48+568	0.94366720	0.02347214

Table 9. Result for JPD of 15 variables and 4 inconsistent constraints.

	# projections	I-divergence	I-aggregate
GEMA	617	0.45972134	0.03407491
SMOOTH	1736+5460	0.45978419	0.03408528
A-MOOTH	1736+584	0.45989650	0.03408416

Finally, we conducted an experiment to compare the performance of belief updates on full joint distributions and on BNs. The experiment reported in Table 10 used a BN of 14 binary variables and 4 inconsistent constraints involving a total of 7 district variables. Both GEMA and SMOOTH were run on the full joint distribution (of 10^{14} entries) generated from this BN. The SMOOTH version of BN-IPFP-1 was run directly on the BN. As can be seen in Table 10, belief updates on the full JPD are several orders of magnitudes slower than that on the BN. When these constraints were modified to be consistent, the convergence time for the standard IPFP on the full JPD was 27 second while the time for BN-IIPFP-1 was only 0.323 second.

Table 10. Result for inconsistent constraints: Full JPD vs BN.

Algorithms	# projections	Time
Full JPD using GEMA	784	459s
Full JPD using SMOOTH	2769	1887s
SMOOTH on BN-IPFP-1	380	0.656s

Recall (Table 6) that GEMA took more than 7 million I-projections to converge in Experiment 5-2 to modify the belief in a tiny JPD of only three variables. We applied the SMOOTH version of BN-IPFP-1 to the same task after first converting the original JPD to a BN of three nodes; and, much to our surprise, it took only 102 I-projections to converge! Although anecdotal, these results clearly demonstrated significant computational advantages of using BN to represent joint distributions and the practical value of belief update methods based on BN such as the algorithms we developed in this work.

6. Conclusions

In this paper we presented our results on Bayesian network belief update with uncertain evidences. We defined two types of uncertain evidences. The virtual evidence, given as a likelihood ratio, represents uncertainty one has for an observation and it requires the

likelihood ratio be preserved in updated BN. The soft evidence, given as a distribution over one or more variables, represents the uncertainty of an event one is observing, and it requires this distribution be preserved in the updated BN. After establishing the close relations between the Pearl's virtual evidence method, the Jeffrey's rule, and the I-projection, we developed the efficient algorithms for BN belief updates with multiple soft evidences. One advantage of BN-IPFP-1, in contrast to some existing methods, is that it can easily work with any BN inference engines. BN-IPFP-2 can provide efficient computation when the number of variables involved in the soft evidences is small. Algorithm SMOOTH was developed by modifying the standard IPFP to support belief update with inconsistent evidences. The convergence of these algorithms was analyzed and experiments of limited scales were conducted to validate these algorithms and to demonstrate their effectiveness. In addition, we for the first time formally established that Equation (6), which is used to compute I-projection in IPFP, not only minimizes the I-divergence but also the total variation between the source and the projected distributions.

BN belief update may be subject to multiple evidences of different types (hard, virtual, and soft), and these evidences may arrive at the same time or at different time. Our BN-IPFP-1 is flexible to support such inference. When all evidences arrive at the same time or hard and virtual evidences arrive before soft evidences, one can first update the beliefs with the given hard and virtual evidences using the conventional BN inference methods and then apply BN-IPFP-1 on the updated BN. A hard or virtual evidence arriving after soft evidences having been absorbed will change the beliefs in the BN, if this change causes $Q(Y^j) \neq R(Y^j)$ for any soft evidence $R(Y^j)$ (i.e., $L_{j,l}(Y^j) \neq 1:1:\cdots 1$ in Step 2.2 of BN-IPFP-1), then BN-IPFP-1 is activated and the iterations renewed until convergence.

As mentioned earlier, one can use virtual evidence to represent the doubt he has on a hard evidence, this can also be applied when one is in doubt of a soft evidence. Recall that in our approach, a soft evidence $R(Y)$ is first converted into a virtual evidence with a virtual node U. If our doubt of $R(Y)$ can be represented as a likelihood $L(U)$, then we can create another virtual node V with U as its only parent and its CPT determined by $L(U)$. Then instantiation of V to true will apply $R(Y)$ with uncertainty of $L(U)$ to the BN.

We are continuing our research effort in this fruitful area along several directions. Our proof of convergence of SMOOTH is only done for the case of two inconsistent constraints, we are actively working on generalizing it to any arbitrary number of constraints. Our experiments show that SMOOTH has a uniform time performance while GEMA is data sensitive and it sometime converges much faster than SMOOTH. We are examining the factors that may be the causes for the performance differences and hoping to find a way to utilize some of the findings to improve the efficiency. We realized that GEMA, although originally devised for general joint distributions, may be adapted to BNs. We are working on developing a BN version of GEMA algorithm.

In this work, we considered constraints $R(y^j)$ as soft evidences to modify the current beliefs. These low dimensional distributions can also be pieces of new knowledge which are more up-to-date, more accurate, or more location specific, and absorbing these into a larger distribution is a process of knowledge integration or knowledge-base update. In the

past we have developed IPFP-based algorithms that absorb the low dimensional distributions by a BN by only modifying its CPTs. The ideas of SMOOTH can be easily incorporated into these algorithms to deal with inconsistent data. However, when the degree of inconsistency is large, it is more sensible to also change the network structure (DAG) of the given BN. We are working on devising such algorithms based on the description length minimization approach.

Acknowledgment

This work was supported in part by NIST award 60NANB6D6206, NSF award IIS-0326460, and the China Scholarship Council (CSC).

Appendix

Proof of Theorem 2.

Let $R(Y)$ be a constraint and $Q(X)$ be the I-projection of $P(X)$ on $\mathbf{P}_{R(Y)}$. Let $\tilde{Q}(X) \in \mathbf{P}_{R(Y)}$ and $Z = X \backslash Y$. Note that $Q(x) = P(x)R(y)/P(y)$ by (6), then

$$\delta(P,Q) = \sum_{x\in\mathbf{X}} \left|P(x) - Q(x)\right| = \sum_{x\in\mathbf{X}} \left|P(x) - P(x)\frac{R(y)}{P(y)}\right| = \sum_{x\in\mathbf{X}} P(x)\left|1 - \frac{R(y)}{P(y)}\right|$$

$$= \sum_{y\in\mathbf{Y}}\sum_{z\in\mathbf{Z}} P(y,z)\left|1 - \frac{R(y)}{P(y)}\right| = \sum_{y\in\mathbf{Y}} P(y)\left|1 - \frac{R(y)}{P(y)}\right| = \sum_{y\in\mathbf{Y}} \left|P(y) - R(y)\right|. \quad \text{(A-1)}$$

Note that $\tilde{Q}(Y) = R(Y)$ since $\tilde{Q}(X) \in \mathbf{P}_{R(Y)}$, then

$$\delta(P,\tilde{Q}) = \sum_{x\in\mathbf{X}} \left|P(x) - \tilde{Q}(x)\right| = \sum_{y\in\mathbf{Y}}\sum_{z\in\mathbf{Z}} \left|P(y,z) - \tilde{Q}(y,z)\right|$$

$$\geq \sum_{y\in\mathbf{Y}} \left|\sum_{z\in\mathbf{Z}} P(y,z) - \sum_{z\in\mathbf{Z}} \tilde{Q}(y,z)\right| = \sum_{y\in\mathbf{Y}} \left|P(y) - \tilde{Q}(y)\right| = \sum_{y\in\mathbf{Y}} \left|P(y) - R(y)\right| \quad \text{(A-2)}$$

Comparing (A-1) and (A-2), we have $\delta(P,Q) \leq \delta(P,\tilde{Q})$ for any $\tilde{Q}(X) \in \mathbf{P}_{R(Y)}$, thus I-projection minimizes the total variation. □

Proof of Theorem 4.

Note that, since for any constraint $R(Y^j) \in \mathbf{R}$, $Y^j \subseteq Y \subseteq X$. Then, for any distribution $Q(X) \in \mathbf{P}_{R(Y^j)}(X)$, its marginal $Q(Y) \in \mathbf{P}_{R(Y^j)}(Y)$. This implies that if $\mathbf{R}$ is consistent with respect to X, it is consistent with respect to $Y \subseteq X$.

Now consider the I-projection at any iteration of IPFP. By (7-1) we have

$$Q_k(x) = Q_{k-1}(x)\cdot\frac{R(y^j)}{Q_{k-1}(y^j)} = Q_{k-1}(x \mid y)\cdot Q_{k-1}(y)\frac{R(y^j)}{Q_{k-1}(y^j)} = Q_{k-1}(x \mid y)\cdot Q_k(y). \quad \text{(A-3)}$$

Note that $Q_{k-1}(x \mid y)$ is kept constant, therefore, for all k, $Q_k(x \mid y) = Q_0(x \mid y)$. Also, by (6), $Q_k(Y)$ is the I-projection of $Q_{k-1}(Y)$ on $\mathbf{P}_{R(Y^j)}(Y)$. In other words, the iterative I-projections of $Q_k(x)$ are realized by the I-projections of $Q_k(y)$. Since $\mathbf{R}$ is consistent, then when $Q_k(y)$ converges to $Q^*(y)$ we have

$$Q^*(x) = Q_0(x \mid y) \cdot Q^*(y) = Q_0(x) \frac{Q^*(y)}{Q_0(y)}. \qquad \square$$

Proof of Theorem 6.

(1) Prove the convergence of BN-IPFP-1. Consider the k^{th} iteration of BN-IPFP-1 in which constraint $R(Y^j) \in \mathbf{R}$ is selected to update $Q_{k-1}(X)$, the joint distribution of the BN at beginning of the iteration. In Step 2.2, $R(Y^j)$ is converted to a virtual evidence based on $R(Y^j)$ and $Q_{k-1}(X)$

$$L(Y^j) = \frac{R(y^j_{(1)})}{Q_{k-1}(y_{(1)})} : \frac{R(y^j_{(2)})}{Q_{k-1}(y_{(2)})} : \cdots : \frac{R(y^j_{(s)})}{Q_{k-1}(y_{(s)})}.$$

According to Theorem 5 of Ref. 3, the new distribution $Q_k(X)$ obtained by applying this virtual evidence to the current BN is identical to the one obtained by applying $R(Y^j)$ to $Q_{k-1}(X)$ by Jeffrey's rule of (2). As shown in (6) in Subsection 2.1, $Q_k(X)$ is the same as the I-projection of $Q_{k-1}(X)$ on $\mathbf{P}_{R(Y^j)}$. That is

$$Q_k(x) = Q_{k-1}(x) \cdot \frac{R(y^j)}{Q_{k-1}(y^j)},$$

which by Definition 4, is one step of the IPFP. Therefore, the sequence $(Q_{k-1}(x))_{k=1}^{\infty}$ generated from BN-IPFP-1 is the same as that generated by IPFP using **R**, starting at $Q_0(x) = P(x)$. Since IPFP converges with consistent constraints, so does BN-IPFP-1.

(2) Prove the convergence of BN-IPFP-2. It is a direct consequence of Theorem 4. $\square$

Proof of Theorem 7.

Without loss of generality, let Phase 2 start with $Q_0(X)$ which satisfies $R(Y^2)$. By Step 1 of Phase 2, the two constraints start with $R_0(Y^1) = R(Y^1)$ and $R_0(Y^2) = R(Y^2)$. Let $Q_1(x)$ be the I-projection of $Q_0(X)$ on $\mathbf{S2} = \mathbf{P}_{R_1(Y^2)}$, $Q_2'(X)$ the I-projection of $Q_1(X)$ on $\mathbf{S1} = \mathbf{P}_{R_0(Y^2)}$, and $Q_2(X)$ the I-projection of $Q_1(X)$ on $\mathbf{P}_{R_1(Y^2)}$, respectively where by Step 3.2 $R_1(y^2) = \alpha R_0(y^2) + (1-\alpha) Q_1(y^2)$ (see Fig. A-1).

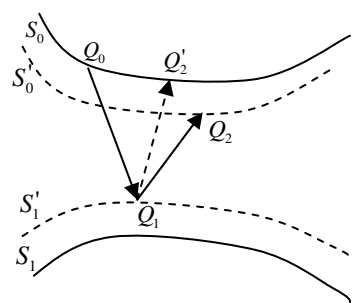


Figure A-1. SMOOTH convergence for $m = 2$.

Since $Q_2'(x)$ is the I-projection of $Q_1(x)$ on $\mathbf{P}_{R_0(y^2)}$, we have by Theorem 2

$$\delta(Q_1, Q_2') \le \delta(Q_0, Q_1). \tag{A-4}$$

Since

$$Q_2'(x) = Q_1(x)\frac{R_0(y^2)}{Q_0(y^2)},$$

and by (13-1)

$$\begin{aligned} Q_2(x) &= Q_1(x)\frac{R_1(y^2)}{Q_0(y^2)} \\ &= \alpha Q_1(x)\frac{R_0(y^2)}{Q_1(y^2)} + (1-\alpha)Q_1(x) \\ &= \alpha Q_2'(x) + (1-\alpha)Q_1(x), \end{aligned}$$

we have

$$\begin{aligned} \delta(Q_1, Q_2) &= \Sigma_{x\in\mathbf{X}} \mid Q_1(x) - Q_2(x) \mid \\ &= \Sigma_{x\in\mathbf{X}} \mid Q_1(x) - (\alpha Q_2'(x) + (1-\alpha)Q_1(x)) \mid \\ &= \Sigma_{x\in\mathbf{X}} \alpha \mid Q_1(x) - Q_2'(x) \mid \\ &= \alpha \cdot \delta(Q_1, Q_2'). \end{aligned} \tag{A-5}$$

Combining (A-4) and (A-5) and the fact that $\alpha \in (0,1)$, we have

$$\delta(Q_1, Q_2) < \delta(Q_0, Q_1). \tag{A-6}$$

Since (A-4), (A-5) and (A-6) hold for any two consecutive I-projections (two iterations of Step 3 in Phase 2), $\delta(Q_{k-1}, Q_k)$ is strictly decreasing, and thus Phase 2 of SMOOTH converges. □

References

1. H. H. Bock, A conditional iterative proportional fitting (CIPF) algorithm with applications in the statistical analysis of discrete spatial data, *Bull. ISI*, *Contributed papers of 47th Session in Paris*, Vol. 1, 1989, pp. 141–142.
2. D. T. Bwon, A note on approximation to discrete probability distributions, *Information and Control* **2** (1959) 386–392.
3. H. Chan and A. Darwiche, On the revision of probabilistic beliefs using uncertain evidence, *Artificial Intelligence* **163** (2005) 67–90.
4. P. L. Combettes, The foundations of set theoretic estimation, *Proceedings of IEEE* **81**(2) (February, 1993).
5. E. Cramer, Probability measures with given marginals and conditionals: *I*-projections and conditional iterative proportional fitting, *Statistics and Decisions* **18** (2000) 311–329.
6. I. Csiszár, Information-type measures of difference of probability distributions and indirect observation, *Studia Sci. Math. Hungar.* **2** (1967) 229–318.
7. I. Csiszar, *I*-divergence geometry of probability distributions and minimization problems, *The Annuals of Probability* **3**(1) (1975) 146–158.
8. W. E. Deming and F. F. Stephan, On a least square adjustment of a sampled frequency table when the expected marginal totals are known, *Ann. Math. Statist.* **11** (1940) 427–444.

9. Z. Ding, Y. Peng and R. Pan, A Bayesian approach to uncertainty modeling in OWL ontology, in *Proc. Int. Conf. Advances in Intelligent Systems − Theory and Applications,* November 2004, Luxembourg.
10. S. E. Fienberg, An iterative procedure for estimation in contingency tables, *Ann. Math. Statist*. **41**(3) (1970) 907−917.
11. S. Haberman, *The Analysis of Frequency Data* (University of Chicago Press, Chicago, 1974).
12. R. Jeffrey, *The Logic of Decisions*, 2nd Edition (University of Chicago Press, Chicago, 1983).
13. R. Kruithof, *Telefoonverkeersrekening*, De Ingenieur, **52** (1937) 15−25.
14. S. Kullback and R. A. Leibler, On information and sufficiency, *Ann. Math. Statist.* **22** (1951) 79−86.
15. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufman, San Mateo, 1988).
16. J. Pearl, Jeffery's rule, passage of experience, and neo-Bayesianism, in *Knowledge Representation and Defeasible Reasoning*, eds. H. E. Kyburg, Jr., R. P. Loui and G. N. Carlson (Kluwer Academic Publishers, Boston, 1990), pp. 245−265.
17. Y. Peng and Z. Ding, Modifying Bayesian networks by probability constraints, in *Proc. 21st Conf. Uncertainty in Artificial Intelligence*, July 2005, Edinburgh.
18. L. Rüschendorf, Convergence of the iterative proportional fitting procedure, *Ann. Statist*. **23**(4) (1995) 1160−1174.
19. M. Valtorta, Y. Kim and J. Vomlel, Soft evidential update for probabilistic multiagent systems, *Int. J. Approximate Reasoning* **29**(1) (2002) 71−106.
20. J. Vomlel, Methods of probabilistic knowledge integration, PhD Thesis, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, December 1999.
21. J. Vomlel, Integrating inconsistent data in a probabilistic model, *J. Applied Non-Classical Logics* **14**(3) (2004) 1–20.
22. C. Wagner, Probability kinematics and commutativity, *Philosophy of Science* **69** (2002) 266−278.