Belief Update in Bayesian Networks Using Uncertain Evidence*

Rong Pan, Yun Peng and Zhongli Ding
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County, Baltimore, MD 21250
{panrong1, ypeng, zding1}@csee.umbc.edu

Abstract

This paper reports our investigation on the problem of belief update in Bayesian networks (BN) using uncertain evidence. We focus on two types of uncertain evidences, virtual evidence (represented as likelihood ratios) and soft evidence (represented as probability distributions). We review three existing belief update methods with uncertain evidences: virtual evidence method, Jeffrey's rule, and IPFP (iterative proportional fitting procedure), and analyze the relations between these methods. This in-depth understanding leads us to propose two algorithms for belief update with multiple soft evidences. Both of these algorithms can be seen as integrating the techniques of virtual evidence method, IPFP and traditional BN evidential inference, and they have clear computational and practical advantages over the methods proposed by others in the past.

1. Introduction

In this paper, we consider the problem of belief update in Bayesian Networks (BN) with uncertain evidential findings. There are three main methods for revising the beliefs of a BN with uncertain evidence: *virtual evidence method* [2], *Jeffrey's Rule* [1], and *iterative proportional fitting procedure* (IPFP) [6]. This paper reports our analysis of these three belief update methods and their interrelationships. We will show that when dealing with a single evidential finding, the belief update of both virtual evidence method and Jeffrey's rule can be viewed as IPFP with a single constraint. Also, we present two methods we developed for belief update with *multiple* soft evidences and prove their correctness. Both of these methods integrate the virtual evidence method and IPFP, and they can be easily implemented as a wrapper on any existing BN inference engine.

We adopt the following notations in this paper. A BN is denoted as *N*. *X* , *Y*, and *Z* are for sets of variables in a BN, and *x* or x_i are for a configurations of the states of *X*. Capital letters *A*, *B*, *C* are for single variables. Capital letters *P*, *Q*, *R*, are for probability distributions.

2. Soft Evidence and Virtual Evidence

Consider a Bayesian network *N* over a set of variables *X* modeling a particular domain. *N* defines a joint distribution $P(X)$. When giving $Q(Y)$, an observation of a probability distribution on variables $Y \subseteq X$, Jeffrey's rule claims that the distribution of all other variables under this observation should be updated to

$$Q(X \setminus Y) = \sum_i P(X \setminus Y \mid y_i) Q(y_i)\,, \tag{1}$$

where y_i is a state configuration of all variables in *Y*. Jeffrey's rule assumes $Q(X \setminus Y \mid Y) = P(X \setminus Y \mid Y)$, i.e., invariance of the conditional probability of other variables, given *Y*, under the observation. Thus

$$Q(X) = P(X \setminus Y \mid Y) Q(Y) = P(X)\frac{Q(Y)}{P(Y)} \tag{2}$$

Here $Q(Y)$ is what we called soft evidence. Analogous to conventional conditional probability, we can also write $Q(Y)$ as $P(Y \mid se)$, where *se* denotes the soft evidence behind the soft evidential finding of $Q(Y)$. $P(Y \mid se)$ is interpreted as the posterior probability distribution of *Y* given soft evidence *se*.

Unlike soft evidence, virtual evidence utilizes a likelihood ratio to represent the observer's strength of confidence toward the observed event. Likelihood ratio *L*(*Y*) is defined as

$$L(Y) = (P(Ob(y_1) \mid y_1) : \ldots : P(Ob(y_m) \mid y_m))\,,$$

where $P(Ob(y_i) \mid y_i)$ is interpreted as the probability we observe *Y* is in state y_i if *Y* is indeed in state y_i. The posterior probability of *Y*, given the evidence, is

$$\begin{aligned} P(Y \mid ve) &= c \cdot P(Y) \cdot L(Y) \\ &= c \cdot (P(y_1)L(y_1), \ldots, P(y_n)L(y_n)), \end{aligned} \tag{3}$$

where $c = 1/\sum_i P(y_i)L(y_i)$ is the normalization factor [3]. And since *Y* *d*-separates virtual evidence *ve* from all other

* This work was supported in part by DARPA contract F30602-97-1-0215 and NSF award IIS-0326460.

variables, beliefs on $X \setminus Y$ are updated using Bayes' rule. Similar to equation (2), this *d*-separation leads to

$$P(X \mid ve) = P(X)\frac{P(Y \mid ve)}{P(Y)} = c \cdot P(X) \cdot L(Y) \qquad (4)$$

Virtual evidence can be incorporated into any BN inference engine using a dummy node. This is done by adding a binary node ve_Y for the given $L(Y)$. This node does not have any child, and has all variables in Y as its parents. The CPT of ve_Y should conform to the likelihood ratio. By instantiating ve_Y to *True*, the virtual evidence $L(Y)$ is entered into the BN and the belief can then be update by any BN inference algorithm.

3. IPFP on Bayesian Network

Iterative proportional fitting procedure (IPFP) is a mathematical procedure that modifies a joint distribution to satisfy a set of probability constraints [6]. A *probability constraint* $R(Y)$ to distribution $P(X)$ is a distribution on $Y \subseteq X$. We say $Q(X)$ is an I_1-*projection* of $P(X)$ on a set of constraints $\boldsymbol{R}$ if the *I-divergence* between P and Q is the smallest among all distributions that satisfy $\boldsymbol{R}$.

I-divergence (also known as *Kullback-Leibler distance* or *cross-entropy*) is a measurement of the distance between two joint distributions P and Q over X:

$$I(P \| Q) = \sum_{P(x)>0} P(x) \log \frac{P(x)}{Q(x)}\ . \qquad (5)$$

$I(P \| Q) \geq 0$ for all P and Q, the equality holds only if $P = Q$.

For a given distribution $Q_0(X)$ and a set of *consistent*[1] constraints $\boldsymbol{R} = \{R(Y_1), \ldots, R(Y_m)\}$, IPFP converges to $Q^*(X)$ which is an I_1-projection of $Q_0(X)$ on $\boldsymbol{R}$ (assuming there exists at least one distribution that satisfies $\boldsymbol{R}$). $Q^*(X)$, which is unique for the given $Q_0(X)$ and $\boldsymbol{R}$, can be computed by iteratively modifying the distributions according to the following formula, each time using one constraint in $\boldsymbol{R}$:

$$Q_k(X) = Q_{k-1}(X) \cdot \frac{R(Y_i)}{Q_{k-1}(Y_i)}\ , \qquad (6)$$

where m is the number of constraints in $\boldsymbol{R}$, and $i = ((k-1) \bmod m) + 1$.

We can see that equations (2), (4) and (6) are in the same form. We can regard the belief update with soft evidence by Jeffrey's rule as an IPFP process of a single constraint $P(Y \mid se)$, and similarly regard belief update with virtual evidence by likelihood ratio as an IPFP process of a single constraint $P(Y \mid ve)$. As such, we say that belief update by uncertain evidence amounts to change the given distribution so that 1) it is consistent with the evidence; and 2) it has the smallest I_1-divergence to the original distribution.

Moreover, IPFP provides a principal approach to belief update with multiple uncertain evidential findings. By treating these findings as constraints, the iterative process of IPFP leads to a distribution that is consistent with ALL uncertain evidences and is as close as possible to the original distribution.

Note that, unlike virtual evidence method, both Jeffrey's rule and IPFP *cannot* be directly applied to BNs because their operations are defined on the *full* joint probability distribution, and they do not respect the structure of BN [4].

4. Inference with Multiple Soft Evidential Findings

Valtorta, Kim and Vomlel have devised a variation of Junction-Tree (JT) algorithm for belief update with multiple soft evidences using IPFP [5]. In this algorithm, when constructing the JT, a clique (the *Big Clique*) is specifically created to hold all soft evidence nodes. Let C denote this big clique, $Y = \{Y_1, \ldots, Y_k\}$ and $\{se_1, \ldots, se_k\}$ denotes soft evidence variables and the respective soft evidences, and X denotes the set of all variables. This Big Clique algorithm absorbs soft evidences in C by updating the potential of C with the following IPFP formulae, iterating over all evidences $Q(Y_j)$s:

$$Q_0(C) = P(C)$$
$$Q_i(C) = Q_{i-1}(C)\frac{P(Y_j \mid se_j)}{Q_{i-1}(Y_j)}$$

where $j = 1+((i\text{-}1) \bmod k)$. The above procedure is iterated until $Q_n(Y_j)$ converges to $P(Y_j \mid se_j)$ for all j. Finally, $Q(C)$ is distributed to all other cliques, again using traditional JT algorithm.

This Big Clique algorithm becomes inefficient in both time and space when the size of the big clique itself becomes large. Besides, it works only with Junction Tree, and thus cannot be adopted by those using other inference mechanisms[2]. Also, it requires incorporating IPFP operations into the JT procedure, causing re-coding of the existing inference algorithm. To address these shortcomings, we propose two new algorithms for inference with multiple soft evidential findings. Both algorithms utilize IPFP, although in quite different ways. The first algorithm combines the idea of IPFP and the encoding of soft evidence

[1] A set of constraints $\boldsymbol{R}$ is said to be consistent if there exists a distribution $Q(X)$ that satisfies all R_i in $\boldsymbol{R}$. Obviously, two constraints are inconsistent if they give different distributions to the same variable. More discuss of this matter is given in Section 7.

[2] Valtorta and his colleagues also developed another algorithm iteratively 1) updates the potential of the clique which contains variables of one soft evidence by (6) and 2) propagates the updated potential to the rest of the network. They mentioned the possibility of implementing this method as a wrapper around Hugin shell or other JT engines, but no suggestion of how this can be done was given [12].

by virtual evidence. The second algorithm is similar to the Big Clique algorithm but it *decouples* the IPFP with Junction Tree.

4.1 Iteration on the Network

As pointed out by Pearl [3], soft evidence can be easily translated into virtual evidence when it is on a single variable. Given a piece of soft evidence *se* on variable *A*, if we want to find a likelihood ratio $L(A)$ such that

$$P(A) \cdot L(A) = P(A \mid se),$$

then we have

$$L(A) = \frac{P(A \mid se)}{P(A)} = \left(\frac{P(a_1 \mid se)}{P(a_1)}, \ldots, \frac{P(a_n \mid se)}{P(a_n)}\right). \quad (7)$$

A problem arises when multiple soft evidences se_1, se_2, …, se_m are presented. Applying one virtual evidence ve_i will have the same effect as applying the soft evidence se_i, in particular, the posterior probability of Y_i is made equal to $P(Y_i \mid se_i)$. This is no longer the case when all of these virtual evidences are present. Now, the belief on Y_i is not only influenced by ve_i, but also by all other virtual evidences. As the result, the posterior probabilities of Y_i's are **NOT** equal to $P(Y_i \mid se_i)$. Therefore, what is needed is a method that can convert a set of soft evidences to one or more likelihood ratios which, when applied to the BN, update the posterior probability of Y_i to $P(Y_i \mid se_i)$.

Algorithm 1 presented below accomplishes this purpose by combining the idea of IPFP and the virtual evidence method. Roughly speaking, this algorithm, like the IPFP, is an iterative process and one soft evidence se_i is considered at each iteration. If the current probability of Y_i equals $P(Y_i \mid se_i)$, then it does nothing, otherwise, a new virtual evidence is created based on the current probability of Y_i and the evidence $P(Y_i \mid se_i)$. We will show that when this algorithm converges, the probability of Y_i is equal to $P(Y_i \mid se_i)$. To better describe the algorithm, we adopt the following notations:

- *P*: the prior probability distribution.
- P_k: the probability distribution at k^{th} iteration.
- $ve_{i,j}$: the j^{th} virtual evidence created for the i^{th} soft evidence.

Algorithm 1. Consider a BN *N* with prior distribution $P(X)$, and a set of *m* soft evidential findings $SE = (se_1, se_2, \ldots, se_m)$ with $P(Y_1 \mid se_1),\ldots, P(Y_m \mid se_m)$. We use the following iteration method for belief update:

1. $P_0(X) = P(X)$; $k = 1$;
2. Repeat the following until convergence;
   - 2.1 $i = 1 + (k-1) \bmod m$; $j = 1 + \lfloor (k-1)/m \rfloor$;
   - 2.2 construct virtual evidence $ve_{i,j}$ with likelihood ratio
   $$L(Y_i) = \left(\frac{P(y_{i,1} \mid se)}{P_{k-1}(y_{i,1})}, \ldots, \frac{P(y_{i,s} \mid se)}{P_{k-1}(y_{i,s})}\right)$$
   where $y_{i,1},\ldots,y_{i,s}$ are state configurations of Y_i;
   - 2.3 Obtain $P_k(X)$ by updating $P_{k-1}(X)$ with $ve_{i,j}$ using standard BN inference;
   - 2.4 $k = k + 1$;

The algorithm cycles through all soft evidences in *SE*. At the k^{th} iteration, the i^{th} soft evidence se_i is selected (step 2.1) to update the current distribution $P_{k-1}(X)$. This is done by constructing a virtual evidence $ve_{i,j}$ according to equation (7). The second subscript here, *j*, is the number of virtual evidences created for se_i, it is incremented in every *m* iterations. When converged, we can form a single virtual evidence node ve_i for each soft evidence se_i with the likelihood ratio that is the product of likelihood ratios of all $ve_{i,j}$, $ve_i = \prod_j ve_{i,j}$. The convergence and correctness of Algorithm 1 is established in Theorem 1.

Theorem 1. If the set of soft evidence $SE = (se_1, se_2, \ldots, se_m)$ is consistent, then Algorithm 1 converges with joint distribution $P^*(X)$, and $P^*(Y_i) = P(Y_i \mid se_i)$ for all se_i in *SE*.

4.2 Iteration on Local Distributions

Algorithm 1 may become expensive when the given BN is large because it updates the beliefs of the entire BN in each iteration (step 2.3). Following is another algorithm that iterates virtual evidence on joint distribution of only evidence variables:

Algorithm 2. Consider a Bayesian network *N* and a set of *m* soft evidential findings $SE = (se_1, se_2, \ldots, se_m)$ to *N* with $P(Y_1 \mid se_1),\ldots, P(Y_m \mid se_m)$. Let $Y = Y_1 \cup \ldots \cup Y_m$. We use the following iteration method for belief update:

1. Use any BN inference method on *N* to obtain $P(Y)$, the joint distribution of all evidence variables.
2. Apply IPFP on $P(Y)$, using $P(Y_1 \mid se_1)$, $P(Y_2 \mid se_2)$, …, $P(Y_m \mid se_m)$ as the probability constraints. Then we have $P(Y \mid se_1, se_2, \ldots, se_m)$.
3. Add to *N* a virtual evidence dummy node to represent $P(Y \mid se_1, se_2, \ldots, se_m)$ with likelihood ratio $L(Y)$ calculated according to equation (7).
4. Apply $L(Y)$ as a single piece of virtual evidence to update beliefs in *N*.

Algorithm 2 also converges to the I_1-projection of $P(X)$ on the set of soft evidences *SE*, even though the iterations are carried out only on a subset of *X*.

Theorem 2. Let $R_1(Y_1), R_2(Y_2), \ldots, R_m(Y_m)$ be probability constraints on distribution $P(X)$. Let $Y = \bigcup_i Y_i$ and $Y \subseteq Z \subseteq X$. Suppose from IPFP we get the I_1-projection of $P(Y)$ on $\{R_1, R_2, \ldots, R_m\}$ as $Q(Y)$ and the I_1-projection of $P(Z)$ on $\{R_1, R_2, \ldots, R_m\}$ as $Q'(Z)$. Let $Q(X)$ and $Q'(X)$ be obtained by applying the Jeffrey's rule on $P(X)$ using $Q(Y)$ and $Q'(Z)$. Then $Q(X) = Q'(X)$.

4.3 Time and Space Performance

The iterations of Algorithm 1, Algorithm 2 and Big Clique algorithm all lead to the same distribution. But at each iteration, Big Clique algorithm updates beliefs of the joint probabilities of the big clique *C*, Algorithm 2 updates the belief of evidence variables *Y*, and Algorithm 1 updates the belief of the whole BN, or say, of all variables in *X*. Clearly, $Y \subseteq C \subseteq X$. However, the time complexity for one iteration of Big Clique is exponential to $|C|$, and Algorithm 2 exponential to $|Y|$, because both require modifying a joint distribution (or potential) table. On the other hand, the time complexity of Algorithm 1 equals to the complexity of the BN inference algorithm it uses for belief update. Both Big Clique and Algorithm 2 are space inefficient. Big Clique needs additional space for the joint potential of *C*, whose size is exponential to $|C|$. Algorithm 2 also needs additional space for the joint distribution of *Y*, and the dummy node of virtual evidence in Step 4 leads to a CPT with size exponential to $|Y|$. In contrast, Algorithm 1 only needs additional space for virtual evidence, which is linear to $|Y|$.

Algorithm 2 is thus more suitable for problems with a large BN but a few soft evidential findings and Algorithm 1 is more suitable for small to moderate-sized BNs. Also, both Algorithm 1 and 2 have the advantage that users do not have to stick to and modify the junction tree when conducting inference with soft evidence. They can be easily implemented as wrappers on any BN inference engines.

5. Experiments and Evaluation

To empirically evaluate our algorithms and to get a sense of how expensive these approaches may be, we have conducted two experiments with artificially made networks of different sizes. We implemented our algorithms as wrappers on a Junction-Tree-based BN inference algorithm. The reported memory consumption does not include those that were used by the Junction Trees, but the reported running time is the total running time.

The first experiment used a BN of 15 binary variables. The results, as can be seen in Table 1 showed that both the time and memory consumptions of Algorithm 1 increase slightly when the number of evidences increases. However, those for Algorithm 2 increase rapidly, consistent with our analysis.

Table 1. Experiment 1

# of findings	# Iterations (Alg 1|Alg 2)		Exec. Time (Alg 1|Alg 2)		Memory (Alg 1|Alg 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	2544,536

Experiment 2 involved BN of different sizes. In all cases we entered the same 4 soft evidential findings involving a total of 6 variables. AS shown in Table 2, the running time of Algorithm 2 increases slightly with the increase of the network size. Especially, the time for IPFP (the time in parentheses) is stable when the network size increases, which means that most increased time was spent on constructing the joint probability distribution from the BN (Step 1 of Algorithm 2). These experiments results confirm our theoretical analysis for the proposed algorithms.

Table 2: Experiment 2.

Size of *N*	# Iterations (Alg 1|Alg 2)		Exec. Time (Alg 1|Alg 2 (IPFP))		Memory (Alg 1|Alg 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

6. Conclusions

In this paper, we analyzed three existing belief update methods for Bayesian networks and established that belief update with one piece of virtual evidence or soft evidence is equivalent to an IPFP with a single constraint. Besides, IPFP can be easily applied to BN with the help of virtual evidence. We proposed two algorithms for belief update with multiple soft evidences by integrating methods of virtual evidence, IPFP and traditional BN inference with hard evidence. Compared with previous soft evidential update methods such as Big Clique, our algorithms have practical advantage of being independent of any particular BN inference engine.

7. References

[1] R. Jeffrey, *The Logic of Decisions, 2nd Edition*, University of Chicago Press. 1983.

[2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, CA. 1988.

[3] J. Pearl. "Jeffery's Rule, Passage of Experience, and Neo-Bayesianism". In *H.E. et al. Kyburg, Jr., editor, Knowledge Representation and Defeasible Reasoning*, 245-265. Kluwer Academic Publishers. 1990.

[4] Y. Peng and Z. Ding, "Modifying Bayesian Networks by Probability Constraints", in *Proceedings of 21st Conference on Uncertainty in Artificial Intelligence*, Edinburgh, Scotland, July 26-29, 2005.

[5] M. Valtorta, Y. Kim, and J Vomlel, "Soft Evidential Update for Probabilistic Multiagent Systems", *International Journal of Approximate Reasoning*, **29**(1), 71-106, 2002.

[6] J. Vomlel, "Methods of Probabilistic Knowledge Integration", *PhD Thesis*, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, December 1999.