Inference via Fuzzy Belief Networks*

Carl G. Looney and Lily R. Liang
Computer Science Department/171
University of Nevada, Reno
Reno, NV 89557
<looney,liang>@cs.unr.edu

Abstract. *The power of belief networks lies in its connective edges where the influences are bidirectional. While Bayesian methods capture bidirectional influences, we propose a simpler and faster method of inferencing from nodal observations that uses bidirectional fuzzy influences that are propagated via fuzzy set membership functions. We need neither the conditional probability tables nor constraining mathematical structure that make inferencing NP-hard.*

1. Introduction

Bayesian Networks. A *Bayesian* (or *belief*) *network* (BN) is an acyclic directed graph [2, 4] where the nodes represent (usually discrete) variables and the arrows represent influences. Each variable has a *state* (its current outcome) and the *network state* is the ensemble of nodal states. The *prior probability tables* (PPT's) at the root nodes and *conditional probability tables* (CPT's) elsewhere are required to compute the *joint probability distribution* (jpdf) over all variables. From a subset of nodes called the *observation* nodes, whose states are detected (nodal *instantiations*) with probabilities as beliefs, the states at a *query* subset of nodes, along with the *hidden nodes* in between, are to be inferred.

Figure 1 shows a simple BN of nodes whose states are true (T) or false (F). Let T_a and F_a be the respective true and false states for A, T_b and F_b be the same for B, etc. For examples, we see from Figure 1 that $P[C=F_c|E=T_e] = 0.3$ and that $P[E=T_e|A=T_a, B=F_b] = 0.4$.

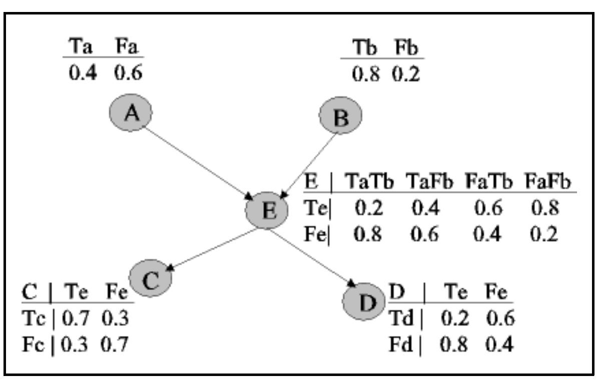

Figure 1. A simple BN.

* Supported by US Army Research Office, Contract No. *DAAD19-99–1-0089*

Let $p(X_k)$ denote the set of parent nodes of X_k. A fundamental principle [4,8] is that each X_k is conditionally independent of its non-descendants given the outcomes of its parents so that the jpdf for Fig. 1 can be factored as

$$P[A,B,C,D,E] = P[A]P[B]P[E|A,B]P[C|E]P[D|E] \quad (1)$$

We can now compute the jpdf for all $2^5 = 32$ possible BN states in Fig. 1, which are the states of $ABCDE$ in $\{T_aT_bT_cT_dT_e, T_aT_bT_cT_dF_e, T_aT_bT_cF_dT_e,...,F_aF_bF_cF_dF_e\}$. By Eqn. (1), the PPT's and CPT's we obtain, e.g.

$$P[T_aF_bF_cT_dT_e] = P[T_a]P[F_b]P[T_e|T_a,F_b]P[F_c|T_e]P[T_d|T_e]$$

$$= (0.40)(0.20)(0.40)(0.30)(0.20) = 0.000192 \quad (2)$$

Inferencing as Queries. To query $E=T_e$ based on the observation $C = T_c$, we employ Bayes' rule of Eqn. (3) below. The CPT at C gives $P[C=T_c|E=T_e] = 0.7$, but the two marginal probabilities $P[C=T_c]$ and $P[E=T_e]$ must be computed. To compute $P[E=T_e]$, we sum the probabilities of all 16 combinations of outcomes $ABCD$ where $E = T_e$ is fixed per Eqn. (4) below ($P[C=T_c]$ is computed similarly). We use these to compute the probability $P[E=T_e|C=T_c]$ via Eqn. (3).

$$P[E=T_e|C=T_c] = P[C=T_c|E=T_e]P[E=T_e]/P[C=T_c] \quad (3)$$

$$P[E=T_e] = \sum_{A,B,C,D} P[A,B,C,D,E=T_e] \quad (4)$$

There are two types of algorithms for BNs, of which the first learns the network structure. The second uses a given fixed structure and computes the update query probabilities based on observations, for which there are *exact* [3] and *approximate* methods [5], but both are NP-hard [1].

Neighborhood Updates. A (*deleted*) *1-neighborhood* (1-nbhd) of a node N is the set of all parents $P(N)$ and all children $C(N)$ of N (single link connections). The 1-nbhd of D in Fig. 1 is $\{E\}$ and the 1-nbhd of E is $\{A, B, C, D\}$. Similarly, a *2-nbhd* uses the second links in all directions from N so that, e.g., the 2-nbhd of D is $\{A, B,C\}$. We can now state some important principles: i) the prior probabilities at any node depend conditionally on its parents only; ii) the posterior probabilities at any node N, given the observations, are conditionally dependent on its 1-nbhd nodes that are updated as results of the observations; iii) any posterior update on any link can only be in a single direction; iv) posterior updates due to a change at node M must be done first on the 1-nbhds of M, and then on 2-nbhds, etc.; and v) if M is observed then the parents $P(M)$ are conditionally dependent (d-connected) through M, while the children $C(M)$ are conditionally independent (blocked by M).

For example, in Fig. 1, observations at C and D update their 1-nbhd $\{E\}$, which depends conditionally on both C and D. But the updating then of C and D from $\{E\}$ would violate the one-direction rule and so is prohibited. From the updated E we update the 2-nbhd $\{A, B\}$ of C and D and the updated A and/or B can not then be used to update their 1-neighbor $\{E\}$. D and B are *d-separated* by E, which means that when E is observed, B and D do not influence each other but are blocked by E. This also applies to C and D. The case of A and B influencing each other is different. When E is observed, then A and B are conditionally dependent via E (*d-connected* by instantiation of E).

2. The Fuzzy Concept in Belief Networks

Probabilistic Belief Rules. Rules of the form $A \Rightarrow B$ are modeled [8] probabilistically via $P[B|A]$ with an axiomatic structure based either on empirical evidence or on models of equally likely elementary outcomes. The power of Bayesian probability comes from capability to apply the influences of other variables, e.g., $P[B|A]$ can be further influenced by the outcome of another event C described by $P[B|A,C]$. The initial probabilities are problematic in that they are either subjective (from experts) or must be mined from data.

Fuzzy Beliefs. Fuzzy logic is axiomatic but is also intuitive and allows simple common sense reasoning based on empirical or reasoned results. The influences can be tuned to fit real world data. Fuzzy influences can work both in both the forward and backward directions as do Bayesian probabilities to adjust the beliefs of the query and unobserved variables when given the observed states. Fuzzy processing establishes conditional belief relationships by observing nodes and then adjusting the fuzzy beliefs of their k-nbhds accordingly with fuzzy conditional influences. Our *fuzzy belief network* (FBN) is describe below and is well suited to the *maximum mode* BN of [7].

In Fig. 1 the influences $(A \Rightarrow E)$ and $(B \Rightarrow E)$ constitute a disjunctive rule $((A \ OR \ B) \Rightarrow E)$. Any single observation at A (or B) can influence all of A's (or B's) k-nbhds that are not blocked, but both A and B can also yield a single update at E. It is known [6] that a probability measure is bounded below by a *necessity measure* and above by a *possibility measure* on the Borel sets of the domain (the power set in the finite discrete case). Our fuzzy beliefs are to satisfy the same bounds and emulate a non-Bayesian probability measure. For simplicity each variable here has two states: the *presence* or *absence* of an entity. The belief that A is present takes the form of a fuzzy truth f_A. The extent to which the belief of a variable state influences the state beliefs of a parent or child is modeled by a *fuzzy set membership function* (FSMF): one for each influence direction.

Fig. 2 shows a Tsukamoto FSMF [9] for propagating a belief influence from a variable X to a parent or child. An observation provides a belief x of the presence of an entity, also denoted by X. The propagating fuzzy belief function from X to Y is f_{XY} that is given by the sigmoid with *center b* and *rate a*. The output fuzzy confidence level f_o is provided by the sigmoid of Eqn. (5) for the input x_o.

$$f_{XY}(x_o) = g(x_o) = 1 / [1 + exp\{-a(x_o - b)\}], \quad 0 \le x \le 1 \quad (5)$$

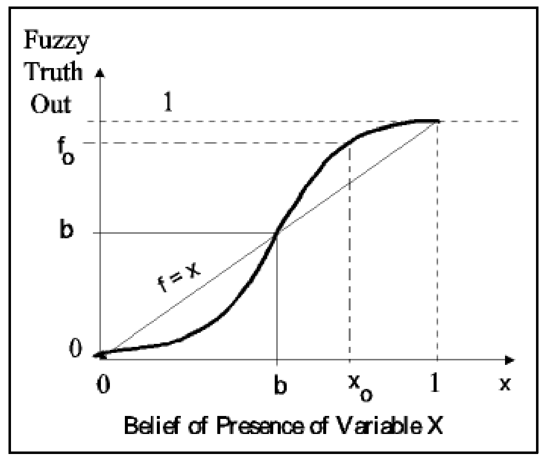

Figure 2. A Tsukamoto FSMF for X.

Fuzzy Propagation. In Fig. 3, the influences $(D \Rightarrow H)$ and $(E \Rightarrow H)$ follow from the observations of D and E. If only one, say D, is observed, then we have only the update from $(D \Rightarrow H)$ that yields

$$f_{H^*} = \alpha f_{DH}(f_D) + (1 - \alpha) f_H \quad (6)$$

where f_{H^*} is the new updated fuzzy belief at H, $f_{DH}(\text{-})$ is the FSMF for influences from D to H, f_D is the observed fuzzy confidence (belief) of the state D and f_H is the previous fuzzy belief for H. But if both D and E are observed, then both must be used in the updating of H. If we update H by D and E separately, then the orders in which we update them will yield different results. Thus we use D and E simultaneously to update H using the positive weights of Eqn. (7).

$$f_{H^*} = \alpha f_{DH}(f_D) + \beta f_{EH}(f_E) + \gamma f_H, \quad (\alpha + \beta + \gamma = 1) \quad (7)$$

Our data structures for the updating influence rules must show schedule of influence rules (developed in the next section) where the influence rules depend on the nodes to be observed and the k-nbhds for k = 1, 2, etc.

3. The Fuzzy Belief Inferencing System

At the nodes of Fig. 3 are the probabilistic prior and conditional beliefs for the states that we designate as *A*, *~A*, *B* and *~B*, etc. The influences go in both directions according to Bayesian principles. However, our fuzzy influence schedule is shown in Table 1 from the observation set {*G*} to implement its k-nbhds.

Table 1. Influences for Figure 1.

G=>D; D=>H; D=>B; H=>E; B=>C; C=>A; C=>F

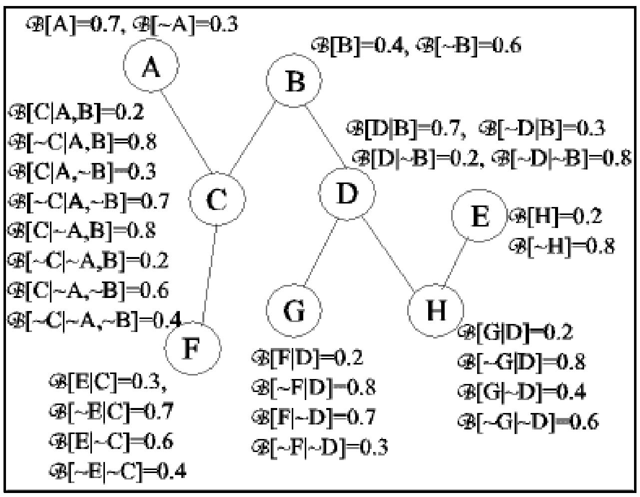

Figure 3. A Bayesian belief network example.

To illustrate fuzzy inferencing, let *G* be observed to be present with fuzzy confidence 0.8. The 1-nbhd of *G* is {*D*}, so the FSMF $f_{GD}(0.8)$ is computed and combined with f_D as

$$f_{D*} = \alpha f_{GD}(0.8) + (1-\alpha)f_D, \quad (0 < \alpha < 1) \qquad (8)$$

which updates *D* according to the influence prescribed in the FSMF f_{GD}. At this point, the influence *G => D* is unflagged so it can not be used to update again. The nodes of the 2-nbhd {*B, H*} of *G* are now updated using the FSMFs for influences *D => H* and *D => B*, after which the influences *D =>H* and *D => B* are unflagged and can not be re-used. Next, the 3-nbhd {*C, E*} of *G* is updated by means of the fuzzy influences *B => C* and *H => E*, after which they are unflagged. Finally, we update the 4-nbhd nodes {*A, F*} of *G* with the influences *C => A* and *C => F*, which are then unflagged. To run another query, we must first restore the initial beliefs and flags. The entire updating is seen to have linear time complexity in the number of nodes.

4. A Computer Run

We use Fig. 4 for this run, which simplifies Fig. 3. We first select the set of observation nodes to be {3,4} and this determines the k-nbhds and thus the fuzzy influence rules listed in Table 2. Unlike BN's, no conditional probabilities are needed and we need to know only the current fuzzy observation beliefs for the nodes. We input these to be those shown in Figure 4. We also renumber the Nodes in Fig. 4.

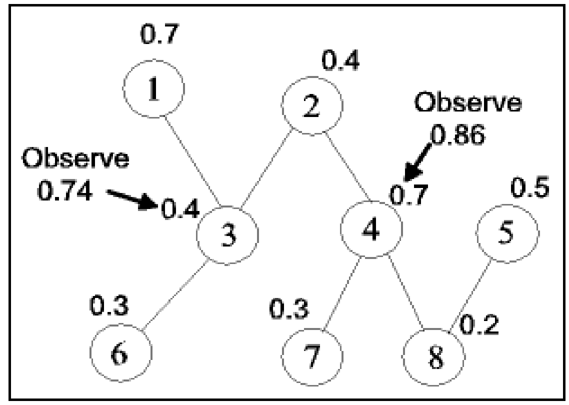

Figure 4. The FBN version for inferencing.

At this point we enter the network file as the fuzzy influence rules shown in Table 2 according to the order of the k-nbhds working outward from the observation nodes of {3,4}. The observation nodes are flagged as active influences so as to be able to fire fuzzy influence rules. From the active observables in {3,4}, we update first the 1-nbhd {*1,2, 6, 7, 8*} and flag each consequent node as the result of firing an influence rule. Then we update the 2-nbhd {5} of {3,*4*} and flag Node *5*. All influence rules in Table 2 have now fired.

Table 2. Updating from Observations of C and D.

Influences: *4=>7; 4=>8; 3,4=>2; 3=>1; 3=>6; 8=>5*

We take $a = 3.0$ and $b = 0.5$ in Eqn. (5) as sigmoid parameters and run our inferencing program with a single sigmoid to obtain the results shown in Table 3. In the general case, each influence rule would use its own sigmoid.

Table 3. Results of Fuzzy Inferencing from C and D.

Node 1:	Start belief = 0.7,	Final belief = 0.686
Node 2:	Start belief = 0.4,	Final belief = 0.511
Node 3:	Start belief = 0.74,	Final belief = 0.74
Node 4:	Start belief = 0.86,	Final belief = 0.86
Node 5:	Start belief = 0.5,	Final belief = 0.426
Node 6:	Start belief = 0.3,	Final belief = 0.419
Node 7:	Start belief = 0.3,	Final belief = 0.468
Node 8:	Start belief = 0.2,	Final belief = 0.402

Table 4 shows the inward influence rules that were entered and the outward reverse influence rules that our program constructs for inferencing from the observation nodes outward via the 1-nbhd, 2-nbhd, etc. of {3,4}. Our rule scheduler follows these reverse influence rules in the order given in Table 2 with the constraints that any influence rule encountered must have the antecedent nodes flagged to fire, in which case they flag the consequents. These consequents appear later as consequents of influence rules in the order and so are fired later.

Table 4. Influences and Reverse Influences.

Influence 1: 7 => 4;	Reverse 1: 4 => 7
Influence 2: 8 => 4;	Reverse 2: 4 => 8
Influence 3: 2 => 3, 4;	Reverse 3: 3,4 => 2
Influence 4: 1 => 3;	Reverse 4: 3 => 1
Influence 5: 6 => 3;	Reverse 5: 3 => 6
Influence 6: 5 => 8;	Reverse 6: 8 => 5

Table 5. Initial and Observed Beliefs.

Node 1,	belief = 0.7	Flag = 0
Node 2,	belief = 0.4	Flag = 0
Node 3,	belief = 0.74	Flag = 1 (Observed)
Node 4,	belief = 0.86	Flag = 1 (Observed)
Node 5,	belief = 0.5	Flag = 0
Node 6,	belief = 0.3	Flag = 0
Node 7,	belief = 0.3	Flag = 0
Node 8,	belief = 0.2	Flag = 0

Table 5 lists the nodes and their starting fuzzy beliefs that consist of prior beliefs and also the latest fuzzy beliefs (confidences) of the observations. Nodes 3 and 4 were observed with the new fuzzy beliefs as shown, so these two nodes were the only ones flagged to be active at the start of the process. As the influences spread outward from Nodes 3 and 4, the new consequent nodes are flagged and then are able to fire the continue the outward spread of influences.

Only a single pass through the influence rules is required to update the beliefs at all nodes that can be updated. Each node can only be updated a single time because of the structure of the rules according to the order of the k-nbhds, which is the reason for Influence 3 in Table 2 and the reason that we enter the inward rules with single antecedents.

5. Analysis and Conclusions

The idea of influences between states of variables in a connected system has been around since 1921 (see [8]) and has been worked rigorously with BN's that use probabilistic and evidence theoretic beliefs. However, constricting the influences to fit those theoretic structures is rather limiting conceptually, computationally and in the obtaining of the conditional probabilities or evidences. The basic concept of influences is intuitive, simple and should be computationally cheap. This is true when fuzzy influence propagation is used.

The fuzzy influence propagation is bidirectional, conditional and the FSMF's can be fine tuned from real world data without being overly constrained by theory that demands mathematical axiomatic correctness even at the expense of lowered performance.

The modeling of a real world system by means of nodes and influence connections should be done by an expert in systems of the type to be captured. Once this is in place, our simple FBN algorithm not only works (and it is very difficult to fault something that works), but it is computationally simple with linear time complexity rather than being NP-hard as are other BN's.

To run this experiment, one can connect to the following URL and enter various parameters for *a* and *b*.

http://ultima.cs.unr.edu/fzBN2/fbn.htm

References

[1] P. Dagum and M. Luby, "Approximating probabilistic inference in Bayesian belief networks is NP-hard," *Artificial Intelligence* **60**(1), 141-153, 1993.

[2] Marek J. Druzdzel, "Five useful properties of probabilistic knowledge representations from the point of view of intelligent systems," *Fundamenta Informaticae* **30**(3-4), 241-254, 1997.

[3] C. Huang and A. Darwiche, "Inference in belief networks: a procedural guide," *Intl. J. Approximate Reasoning*, **15**(3), 225-263, 1996.

[4] Finn V. Jensen, *Bayesian Networks and Decision Graphs*, Springer-Verlag, Inc., New York, 2001.

[5] M. I. Jordan (editor), *Learning in Graphical Models*, MIT Press, Cambridge, MA, 1998.

[6] C. T. Lin and C. S. George Lee, *Neural Fuzzy Systems*, Prentice-Hall, Upper Saddle River, NJ, 1996.

[7] Sucheta Nadkarni and Prakash P. Shenoy, "A Bayesian network approach to making inferences in causal maps," *European J. Operational Research* **128**, 479-498, 2001.

[8] Judea Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, CA, 1988.

[9] Y. Tsukamoto, "An approach to fuzzy reasoning method," in *Advances in Fuzzy Set Theory and Applications*, editors M. M. Gupta, R. K. Ragade and R. R. Yager, North-Holland, Amsterdam, 137-149, 1979.