Inference via Fuzzy Belief Petri Nets

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

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

Abstract

The fuzzy belief Petri net we propose in this paper propagates fuzzy beliefs from observations at nodes that represent measured parameters to fuzzy beliefs of the truths of parameters at hidden and decision nodes. The fuzzy influences spread from the observation nodes throughout our new enhanced bidirectional fuzzy belief Petri net. Compared with Bayesian belief networks, it is simpler and faster in that it needs neither the conditional probability tables that are difficult or impossible to obtain nor is it overly constrained by the mathematical axiomatic structure that makes Bayesian belief inferencing NP-hard. Compared with our previous fuzzy belief networks, it is more flexible in modeling particular situations. We develop here the concept, data structures and algorithm for this network, while future work will make comparative runs.

1. Introduction

Bayesian networks were invented [8] from Wright's *influence graphs* of 1921. They are directed acyclic graphs [1, 3, 4] where each node represents a random variable, the inter-node links designate influences and certain independence assumptions [3, 8] hold. The purpose of a Bayesian network is to aid decision making via conditional probabilistic reasoning from observations of certain nodal parameters to update the probabilistic beliefs of the other nodes based on Bayesian relationships. The major disadvantage is that such updating is NP-hard [2], while a major strength is that the influences can work in either direction along the influence links. Various approximate algorithms exist [7, 9, 10, 11] other than exact methods, but all require the joint distribution function and computation of marginal distributions. The axiomatic structure of Bayesian probability is restrictive and contributes to computational complexity. The requirement of conditional probability tables is often prohibitive.

A less constrained type of belief network is the *fuzzy belief network* (FBN) [5] that uses fuzzy beliefs in place of probabilities and lets the knowledgeable user insert parameters that determine the extent of the fuzzy influence exerted along each link. The influence in each direction is determined by a *gain* function for that direction, which is a logistic type function with a parameter that determines the extent of increase or decrease of the fuzzy belief. From a set of observation nodes whose fuzzy beliefs are given, the nodes one link away are updated first, then the nodes two links away from the observations nodes are updated, and so forth. The update at a node is determined by the fuzzy belief of the influencing nodes and the strength of the influences determined by the gain functions. The fuzzy influences can go in either direction along a link and so each direction has its own gain function.

Fuzzy Petri nets are special Petri nets that allow rule-based decision-making systems to be represented and executed [6]. An example of a fuzzy Petri net (FPN) is shown in Figure 1. The circular nodes represent conditions (propositions) of which each has a token that designates its fuzzy truth (a value from 0 to 1). The bars represent *synapses* where a threshold value is set for comparison with the fuzzy truth and fuzzy truths that meet or exceed the threshold are propagated to the destination nodal condition(s). Thus the synapses are gates that open or close, that is, fire or don't fire, to transmit the fuzzy implication or not.

For example, let Condition A in Figure 1 have a fuzzy truth of 0.77 and Condition B have a fuzzy truth of 0.68. Both A and B are required to fire the synapse with threshold T1, so the AND logic yields the minimum of 0.77 and 0.68. Thus 0.68 must meet or exceed the threshold T1 in order for the synapse to fire and imply the fuzzy truth of Condition E.

Now we demonstrate OR logic. If Condition G already has a fuzzy truth of 0.66 from Conditions E and F via the synapse at T2 and if Condition J fires the synapse at T5 with a fuzzy truth of 0.81, then the fuzzy truth at G is the maximum of 0.66 and 0.81. Thus a fuzzy Petri net implements rules with fuzzy AND and fuzzy OR logic. A fuzzy Petri net allows more control than does a fuzzy rule system with no synaptic thresholds.

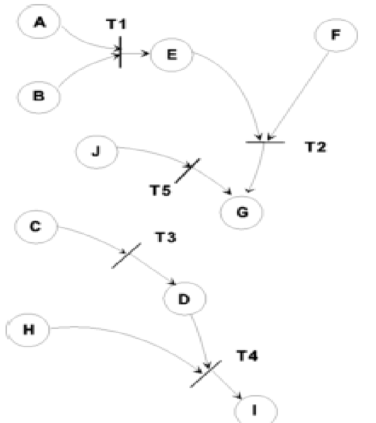


Figure 1. An FPN for fuzzy reasoning

2. Fuzzy belief Petri net architecture

The *fuzzy belief Petri net* (FBPN) is developed here to take advantage of the fuzzy belief network's simple axiomatic structure and utilization of an expert's experience in assigning gain parameters, but yet model the synaptic thresholds of the fuzzy Petri net that allow more flexibility. Thus we combine the bi-directional fuzzy propagation of the FBN with the situation modeling flexibility of the FPN, so the FBPN is therefore more flexible and powerful than either of its two predecessors.

Figure 2 presents an example of a fuzzy belief Petri net. It is indeed a fuzzy Petri net, but it has the extra power allowed by being bi-directional with synaptic thresholds for each direction and the gain functions of which there is also a different one in each direction. This allows more control over the propagation of fuzzy beliefs than either a fuzzy belief network or a fuzzy Petri net.

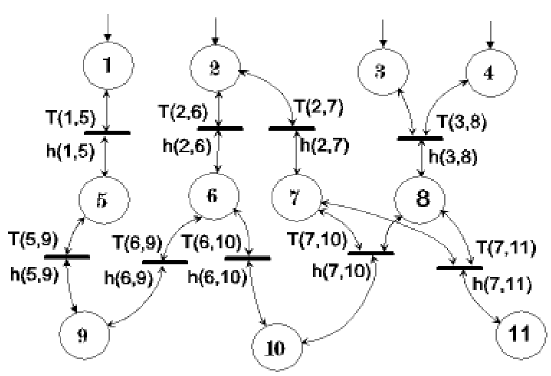


Figure 2. An example of fuzzy belief Petri net.

Fuzzy values from Nodes 3 and 4 in Figure 2 are ANDed at the synapse with threshold T(3,8). Node 2 of Figure 2 provides influences to both Node 6 and Node 7. In the opposite direction, Nodes 6 and 7 each influence Node 2 using OR logic. For Node 10, the ANDed influences of Nodes 7 and 8 is ORed with the influence of Node 6.

When a node in the network is observed with a fuzzy belief, if the belief exceeds its threshold, it will be transformed via a gain function and then propagated to another node. Section 3 and Section 4 will describe the transformation and propagation respectively.

3. Fuzzy belief transformation

In Equation (1), h(r, s) transforms the fuzzy belief f(r) into the fuzzy belief f(s).

$$h(r,s): f(r) \rightarrow f(s), \quad f(s) = [h(r,s)](f(r)) \qquad (1)$$

The gain functions have two forms, depending upon whether the influence is positive or negative. For the positive and negative cases, respectively, we have Equations (2) and (3), where $\alpha(r,s)$ is a given parameter in the gain function h(r,s) with indices r and s.

$$h(r,s)= 1 / [1 + \exp\{-\alpha(r,s)[f(r) - 0.5]\}] \qquad (2)$$
$$h(r,s)=1- \{1 / [1 + \exp\{-\alpha(r,s)[f(r) - 0.5]\}]\} \qquad (3)$$

Figure 3 shows Equation (2) and the one of Equation (3) is just this one turned upside down. In Equation (2), the positive influence of a link is boosted when the fuzzy belief of the antecedent is strong, and is suppressed when it is weak. Equation (3) is for the opposite situation where the negative of the influence is boosted when the fuzzy belief of the antecedent is weak, and is suppressed when it is strong.

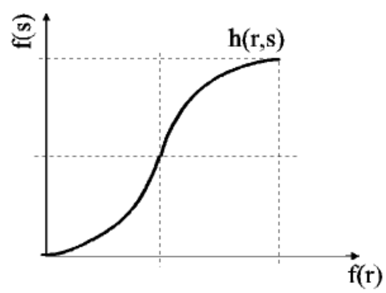


Figure 3. A Sigmoid gain function

4. Fuzzy belief propagation

We divide the connection in FBPN into 4 categories: single node with single node, single node with OR-ed multiple nodes, single node with AND-ed multiple nodes, and hybrid of the previous three categories. Each kind of connection has its own rules for influence propagation, forward and backward. Whether the forward rules or the backward rules of the connection should be used depends on the node(s) of which side of the connection is (are) observed.

In a real fuzzy belief Petri net (FBPN), the links connecting nodes can be either bidirectional or unidirectional. If a link has only one direction, then the influence only propagates on that direction of the link. Table 1 shows types of connections in FBPN and their propagation rules. The following subsections examine the types of connections in FBPN and their propagation rules one by one. Note that even all the links in the connections shown are bidirectional, unidirectional links can also be used with eliminating rules for propagation on any missing direction.

4.1 Single node with single node propagation.

This *single node with single node* (SS) inferencing is the simplest in the four categories. A model of it is shown in Figure 4a of Table 1, where T_{1-2} is the threshold of the transition bar for influence from Node 1 to Node 2, and h_{1-2} is the fuzzy propagation function from Node 1 to Node 2. Similarly, T_{2-1} and h_{2-1} are defined. Suppose Node 1 is observed with belief f1 (shown in Figure 4b), if f_1 is larger than T_{1-2} , the influence of Node 1 will go through the transition bar, propagate via h_{12} and reach Node 2 . So the belief of Node 2, f_{12}, can be updated with the value calculated in Equation (4). In the case that Node 2 is observed (shown in Figure 4c), the belief propagation is similar but in the other direction (Equation (5)).

$$f_2 = f_1 * h_{1-2}, \quad \text{if } f_1 \$ T_{12} \qquad (4)$$
$$0, \quad \text{Otherwise}$$
$$f_1 = f_2 * h_{2-1}, \quad \text{if } f_2 \$ T_{2-1} \qquad (5)$$
$$0, \quad \text{Otherwise}$$

4.2 Single node with multiple nodes propagation.

Single node with multiple node inferencing are divided into two categories based on how those multiple nodes influences that single node. If multiple nodes influence a single node independently, then any one of them can influence that single node when observed without the others observed at the same time. The inferencing is called *single node with OR-ed multiple nodes* (SOM). A model of this category is shown in Figure 5a (only two nodes are OR-ed for simplicity). If those multiple nodes must jointly influence that single node, and can only influence that single node when every one them is observed, then the inferencing is called *single node with AND-ed multiple nodes* (SAM). Figure 6a shows this situation. For simplicity, only two nodes are AND-ed in the figure.

In the SOM model, when Node 1 is observed (shown in Figure 5b), and its influence are propagate to Nodes 2 and 3 connected to it, it is called forward propagation. The belief of Node 1 influences each of Nodes 2 and 3 with a different fuzzy propagation function simultaneously. Equations (6) and (7) follow directly.

$$f_2 = f_1 * h_{1-2}, \quad \text{if } f_1 \$ T_{1-2} \qquad (6)$$
$$0, \quad \text{otherwise}$$
$$f_3 = f_1 * h_{1-3}, \quad \text{if } f_1 \$ T_{1-3} \qquad (7)$$
$$0, \quad \text{otherwise}$$
$$f_1 = f_{2-1} = f_2 * h_{2-1}, \quad \text{if } f_1 \$ T_{1-2} \qquad (8)$$
$$0, \quad \text{Otherwise}$$
$$f_3 = f_1 * h_{1-3}, \quad \text{if } f_1 \$ T_{1-3} \qquad (9)$$
$$0, \quad \text{Otherwise}$$

If only Node 2 is observed (shown in Figure 5c), the belief of Node 1 will be influenced and it will in turn influence the belief of Node 3. This is called backward propagation with partial observation (Equations (8, 9)). From f_{2-1}, the fuzzy belief of Node 1 is based on the influence of Node 2. Similar equations hold for the case that only Node 3 is observed. If both Node 2 and 3 are observed (shown in Figure 5d), we take the maximum of the f_{1-2} and f_{1-3} as f_1 (Equations (10,11,12)). This is called backward propagation with complete observation.

$$f_{2-1} = f_2 * h_{2-1}, \quad \text{if } f_2 \$ T_{2-1} \qquad (10)$$
$$0, \quad \text{Otherwise}$$
$$f_{3-1} = f_3 * h_{3-1}, \quad \text{if } f_3 \$ T_{3-1} \qquad (11)$$
$$0, \quad \text{Otherwise}$$
$$f_1 = \max(f_{2-1}, f_{3-1}); \quad (1) \qquad (12)$$

0, Otherw

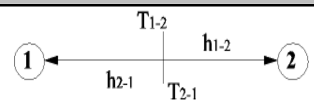
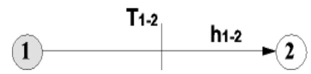
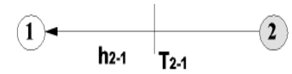
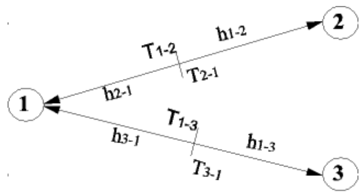
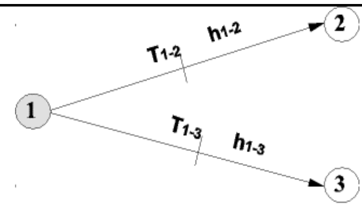
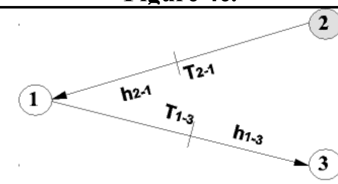
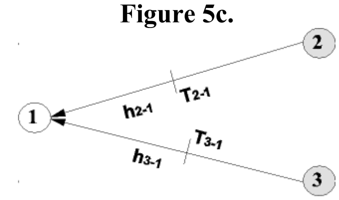
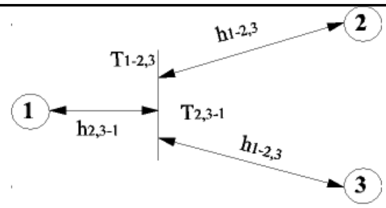
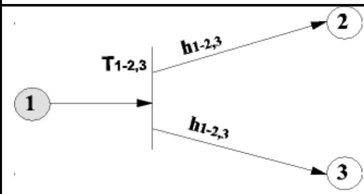
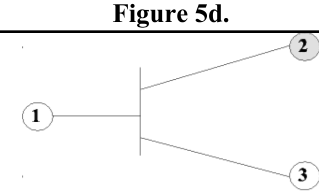
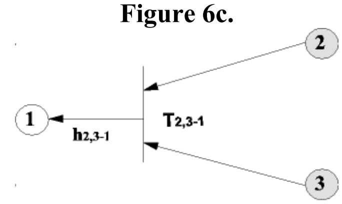
In the SAM model, when Node 1 (2) is observed (shown in Figure 6b), and its influence is propagated to Nodes 2 and 3 connected to it, it is called forward propagation. The belief of Node 1 first passes through the transition bar if it is larger than the threshold $T_{1-2,3}$, and then influences each of Nodes 2 and 3 (Equation 13).

$$f_2 = f_3 = f_1 * h_{1-2,3}, \quad \text{if } f_{1>} \$ T_{1-2} \qquad (13)$$
$$0, \quad \text{Otherwise}$$

If only Node 2 is observed (shown in Figure 5c), the belief of Node 1 will not be influenced since Node 3 is not observed. None of the beliefs of the Nodes in the model can be updated. If both Node 2 and 3 are observed (shown in Figure 5d) , which is complete observation for backward propagation, and if minimum of the f_{2-1} and f_{3-1} is larger than the threshold $T_{3,2-1}$, Equation (14) holds.

$$f_1 = \min(f_2, f_3) * h_{2,3-1}, \quad \text{if } \min(f_2, f_3) \$ T_{2,3-1} \qquad (14)$$
$$0, \quad \text{Otherwise}$$

Table 1. Propagation rules by connection type in FBPN

	Connection type	Forward propagation	Backward propagation
SS	 **Figure 4a.**	 **Figure 4b**	 **Figure 4c.**
SOM	 **Figure 5a.**	 **Figure 5b.**	 **Figure 5c.**  **Figure 5d.**
SAM	 **Figure 6a.**	 **Figure 6b.**	 **Figure 6c.**  **Figure 6d.**

5. An inference example

In Figure 2 if node 6, 7 and 8 are observed, the fuzzy beliefs of the observations propagate as follows.

For Node 6: 6 ÿ 9 ÿ 5 ÿ 1 (propagation stops at the end of the path)

6 ÿ 2

6 ÿ 10 (propagation stops before reaching the other observations 7, 8)

For Node 7: 7 ÿ 2

7,8 ÿ 10

7,8 ÿ 11

For Node 8: 8 ÿ 3,4

Node 2 receives influences from both Node 6 and Node 7. Since those two inputs are ORed, the maximum of them is assigned to Node 2. For Node 10, the ANDed influences of Node 7 and 8 is ORed with influence of Node 6 (hybrid connection). Again, the maximum of the ORed two is assigned to Node 10.

Based on the analysis above, a high level algorithm is developed.

Step 1. Initialized all nodes with fuzzy belief 0

Step 2. Update beliefs of the observed nodes with data in the observation file

Step 3. For each updated node
  for each rule in the network
    if the inverse of this rule is not the one that was used to update the node
      if the updated node is the antecedent or one of the antecedents of this rule

if the consequent of this rule is not an observed node
propagate belief according to the rule described in Section 4

Step 4. If any node was updated during step 3, go back to step3

Step 5. Write the finial belief of each node to the output file

This algorithm makes sure that the observations do not get updated and also a node does not influence the node from which it gets its belief.

A rule file is used to stores the logic rules of the FBPN, including the thresholds and gains associated with each rules. A "0" is used to separate antecedent index (indices) and consequent index. Each row is a rule and has only one consequent.

6. Conclusions and future work

Table 2 shows a summary of the FBPN properties in comparison to the FPN and FBN. The FBPN has more desirable properties than the other two.

Table 2. A summary of FBPN in comparison to FPN and FBN

	fuzzy Petri net (FPN)	fuzzy belief Petri net (FPBN)	fuzzy belief network (FBN)
Token Value	Fuzzy Truth	**Fuzzy Truth**	Fuzzy Truth
linkage Direction	Single	**Two and Single**	Two
Propagation function	No	**Yes**	Yes
Transition Type	ANDing ORing Threshold-ing	**ANDing ORing Threshold -ing**	ORing

We conclude that the fuzzy belief Petri net is a more flexible tool for aiding decision making than either the fuzzy Petri net or the fuzzy belief network. It is also more intuitive and computationally efficient than are Bayesian belief networks.

Future work will investigate methods for training the FBPN on real world data sets. It will also make comparative runs using the FPN, FBN, Bayesian networks and the FBPN to compare accuracy and computational time.

7. References

[1] E. Charniak, "Bayesian Networks without Tears," *AI magazine*, 12, 4, pp. 50-63, 1991.

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

[3] Fin V. Jensen, *Bayesian Networks and Decision Graphs*, Springer, New York, 2001.

[4] Hongjun Li, *An Introduction to Belief Networks*, Tech. Research Report, www.isr.umd.edu/CSHCN.

[5] Carl Looney and Lily Liang, "Inference via Fuzzy Belief Networks" *Proceedings of the ISCA International Conference*, San Diego, Nov. 2002.

[6] Carl Looney, "Fuzzy Petri Nets for Rule Based Decisionmaking," *IEEE Trans. Systems, Man & Cybernetics*, 18(1), Jan. 1988.

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

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

[9] Huaiqing Wang, Changjun Jiang and Shaoyi Liao, "Concurrent reasoning of fuzzy logical Petri nets based on multi-task schedule," *IEEE Trans. Fuzzy Systems* 9(3), 444-449, 2001.

[10] Y. Xiang, "Belief updating in multiply sectioned Bayesian networks without repeated local propagations," *Int. J. Approximate Reasoning* 23, 1-21, 2000.

[11] Nevin L. Zhang, "Computational properties of two exact algorithms for Bayesian networks," *Applied Intelligence* 9, 173-183, 1998.