

FUZZY NEURO-BDI

Haibo Liu, Guochang Gu, Jing Shen

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Abstract: The typical BDI (Belief-Desire-Intention) model of agent is not efficiently computable and the strict logic expression is not easily applicable to the domain with uncertainties. In this paper, a fuzzy neuro-BDI model is proposed. The model is a fuzzy neural network composed of five layers: input (beliefs & desires), fuzzification, commitment, fuzzy intention, and defuzzification layer. In the model, the fuzzy commitment rules and neural network are combined to form intentions from beliefs and desires. The model is demonstrated by solving dog-sheep problem, and the simulation result is satisfactory.

Key words: fuzzy neural network, BDI, agent

1. INTRODUCTION

BDI (Belief-Desire-Intention) model represents an abstraction of human deliberation based on a theory of rational actions in the human cognition process [1]. Singh [2] and Rao & Georgeff [3] have developed logic theories involving multiple worlds and logical formalisms. The logic theories greatly strengthen the usability of BDI and are viewed as critical compositions in the BDI family. Although the BDI model has become almost a norm in the area of Multi-Agent Systems (MAS) during the last 15 years, several problems still remain. One of the key problems is the gap that exists between theory and practice. The BDI model is not efficiently computable and the strict logic expression is not easily applicable to the domain with uncertainties. These uncertainties are related with the perception of the uncertain environments, the action of other agents, and the knowledge of the intelligent agents. In order to maximize the power of BDI, it is necessary to

further develop BDI into a more effective model to deal with uncertainties. There were many researches attempted to solve these kinds of problems, and mostly by using the probabilistic and stochastic methods [4][5][6]. The intelligent agent must find optimal solutions from given user's goals and the environments that the agent explores. But, based on the previous researches, it is impossible to handle Beliefs, Desires, and Intentions represented with fuzzy linguistic values that comprise of uncertainties in human language and knowledge. To solve this problem, we propose a fuzzy neuro-BDI (FNBDI) model in this paper. The model is a fuzzy neural network composed of five layers: input (beliefs & desires), fuzzification, commitment, fuzzy intention, and defuzzification layer. In FNBDI, the fuzzy commitment rules and neural network are combined to form intentions from beliefs and desires.

2. MODELING

The proposed fuzzy neuro-BDI model shown in Figure 1 is a fuzzy neural network composed of five layers as follows.

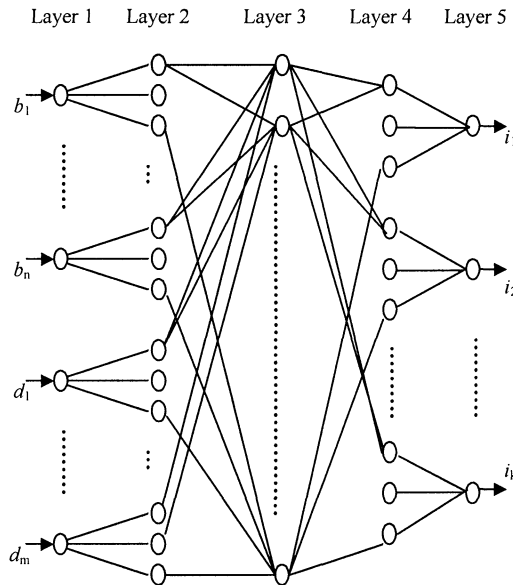


Figure 1. Fuzzy neuro-BDI model

Layer 1: Input layer. The nodes in first layer just directly transmit input values to next layer. The input variables include beliefs and desires (\mathbf{B} , \mathbf{D}). $\mathbf{B}=(b_1, b_2, \dots, b_n)$ denotes the input vector of beliefs that is the information an agent has about the world, other agent, and itself. $\mathbf{D}=(d_1, d_2, \dots, d_m)$ denotes

the input vector of desires that are the states that agent wish to bring to meet the demands of human user. There are $N_1=n+m$ nodes in this layer. The output of layer 1 is

$$\mathbf{O}^1=(o_1^1, o_2^1, \dots, o_{N_1}^1)=(b_1, b_2, \dots, b_n, d_1, d_2, \dots, d_m) \quad (1)$$

Layer 2: Fuzzification layer. Each fuzzy variable of the second layer appearing in the premise part is represented with a condition node. Each of the outputs of the condition node is connected to commitment nodes in the third layer to constitute a condition specified in some rules. Note that the output links must be emitted from proper linguistic terms as specified in fuzzy rules. This layer performs the first inference step to compute matching degrees. The output of this layer is

$$\mathbf{O}^2=(o_1^2, o_2^2, \dots, o_{N_2}^2)=((\mu_{11}(b_1), \mu_{12}(b_1), \dots, \mu_{1T_1}(b_1)), (\mu_{21}(b_2), \mu_{22}(b_2), \dots, \mu_{2T_2}(b_2)), \dots, (\mu_{(n+m)1}(d_m), \mu_{(n+m)2}(d_m), \dots, \mu_{(n+m)T(n+m)}(d_m))) \quad (2)$$

where μ_{pq} is the membership degree of the q th linguistic term in the p th variable for \mathbf{O}^1 , T_p is the number of linguistic term for the p th variable of \mathbf{O}^1 .

There are $N_2=\sum_{p=1}^{n+m} T_p$ nodes in this layer. Several functions such as the triangular function, trapezoidal function, and so on can be adopted as the membership functions μ of linguistic terms.

Layer 3: Commitment layer. Each neuron in this layer corresponds to a single fuzzy commitment rule. The links in this layer are used to perform precondition matching of fuzzy logic rules. Hence, the rule nodes should perform the fuzzy AND operation. In our model, the commitment rules are defined by domain expert's fuzzy knowledge previously as form of

R_h : IF b_1 IS B_1 AND b_2 IS B_2 AND ... AND b_n IS B_n AND d_1 IS D_1 AND d_2 IS D_2 AND ... AND d_m IS D_m THEN g_1 IS G_1 AND g_2 IS G_2 AND ... AND g_l IS G_l

where B_x , D_y , and G_z are linguistic terms of the variable b_x , d_y and g_z corresponding to belief, desire, and subgoal, respectively. There are

$N_3=\prod_{p=1}^{n+m} T_p$ nodes in layer 3. The output of this layer is

$$\mathbf{O}^3=(o_1^3, o_2^3, \dots, o_{N_3}^3) \quad (3)$$

where o_h^3 is the firing strength of fuzzy commitment rule R_h :

$$o_h^3=\mu_{1h} \times \mu_{2h} \times \dots \times \mu_{(n+m)h} \quad (4)$$

where μ_{jh} is the membership degree of the corresponding linguistic term of the j th variable in \mathbf{O}^1 , the linguistic term appears in the conditions of R_h . The \times in Eq.(4) is a fuzzy product operator which can be defined as a various forms such as $\min()$. Note that the output links must be emitted to proper linguistic terms as specified in fuzzy rules.

Layer 4: Fuzzy intention layer. There are $N_4 = \sum_{p=1}^{N_3} S_p$ nodes in this layer, where $N_3 = k$ is the number of nodes in layer 5, S_p is the number of linguistic term for the p th variable of intentions. The neurons in this layer have weighted connections with the neurons in layer 3. The weight vector can be denoted by matrix W with $N_4 \times N_3$ elements that are adjusted during training.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N_3} \\ w_{21} & w_{22} & \cdots & w_{2N_3} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N_4,1} & w_{N_4,2} & \cdots & w_{N_4,N_3} \end{bmatrix} \quad (5)$$

Note that if there is no connection between the i th nodes in layer 4 and the j th nodes in layer 3 then $w_{ij} = 0$. The output of layer 4 is

$$O^4 = (o_1^4, o_2^4, \dots, o_{N_4}^4), \text{ where } o_i^4 = \sum_{j=1}^{N_3} o_j^3 w_{ij} \quad (6)$$

where Σ is a fuzzy sum operator which can be defined as a various forms such as $\max()$.

Layer 5: Defuzzification layer. There are k nodes in this layer, i.e., $N_5 = k$, and the output intention vector is denoted as $I = (i_1, i_2, \dots, i_k)$ which can be calculated according to Eq.(7).

$$i_p = \text{Centroid}(Y_p, \rho_p), \quad p = 1..k \quad (7)$$

where the function $\text{Centroid}()$ denotes the centroid defuzzification process, Y_p and ρ_p are output domain and membership degree of the p th variable of intention. Without loss of generality, we assume that the output domain $Y = \{Y_1, Y_2, \dots, Y_p\} \in \mathbf{R}^k$.

A various learning algorithm may be applied to adjust the weights W and modify the input/output membership functions μ and ρ .

3. SIMULATION

We consider the dog-sheep problem [7] in a 9×9 grid. The dog can occupy one of the 81 positions and so can the sheep. Therefore, there are 6561 states in the system. The aim of the dog is to make the sheep go to the pen that is the (0,0) position for the sheep. $x(t)$ is state vector given by $x(t) = [x_s(t) \ y_s(t) \ x_d(t) \ y_d(t)]$ which stand for the coordinate of the sheep position and the dog position at time instance t . For the 9×9 dog sheep problem, we have $\forall t \ x_s(t) \in \{0..8\}$, $y_s(t) \in \{0..8\}$, $x_d(t) \in \{0..8\}$, $y_d(t) \in \{0..8\}$. However, the dog and the sheep cannot have the same location on the grid as their initial positions. There is a cost of one unit for each step (horizontal or

vertical or diagonal) of a dog as well as of a sheep. The desire of the dog is to move the sheep to the pen (x_p, y_p) i.e. to the $(0,0)$ coordinate, with the least cost c . We say the sheep is in an equilibrium state when given a time instant t' the condition, $\forall t \geq t'$, if $x_d(t)=x_d(t')$ and $y_d(t)=y_d(t')$ then $x_s(t)=x_s(t')$ and $y_s(t)=y_s(t')$, is satisfied. The dog can only move when sheep is in an equilibrium state. The dog can only move one step in one time instant and that step can be in vertical or diagonal direction. The sheep can also only move one step in horizontal, vertical or diagonal direction. The sheep moves based on the rules proposed Kachroo *et al.* [7] but the coordinates in the conditions should be modified to fit the scale of 9×9 grid. The dog moves based on the intentions formed by the FNBDI.

For this dog-sheep problem, we constructed an FNBDI with 7 inputs $(x_s(t), y_s(t), x_d(t), y_d(t), x_p=0, y_p=0, c=0)$ and 2 outputs $(x_d(t+1), y_d(t+1))$. Figure 2 shows the fuzzy sets of fuzzy input-output variables. For the desire variables (x_p, y_p, c) are invariants, we only have 81 fuzzy commitment rules, part of which are shown in Table 1.

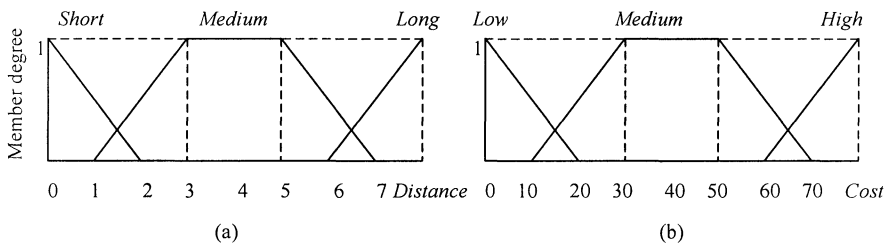


Figure 2: (a) Fuzzy sets for variable $x_s(t), y_s(t), x_d(t), y_d(t), x_p, y_p, x_d(t+1)$, and $y_d(t+1)$; (b) Fuzzy sets for variable c .

Table 1: Part of fuzzy commitment rules (*S: Short, M: Medium, L: Long or Low, H: High*)

Rule	$x_s(t)$	$y_s(t)$	$x_d(t)$	$y_d(t)$	x_p	y_p	c	$x_d(t+1)$	$y_d(t+1)$
1	S	S	S	S	S	S	L	S	S
2	S	S	S	M	S	S	L	S	S
3	S	S	S	L	S	S	L	S	M
4	S	S	M	S	S	S	L	S	S
...		
81	L	L	L	L	S	S	L	L	L

We make use of a mixture of back propagation and least mean square estimation to adjust the connection weights and modify the input/output membership functions. The training data is generating by the dynamic programming (DP) solution used in [7]. We trained the FNBDI using the ANFIS command in MATLAB by specifying the training epoch number to 20 and setting the default values to other training options. Then we replaced the dynamic programming solution with our trained FNBDI solution to form

the moving intentions of the dog. As a simulation result, Figure 3 shows pursuit-evasion tracks of the dog and the sheep. Note that the tracks produced by FNBDI solution superposes that produced by DP, which indicate that the FNBDI solution can solve dog-sheep problem satisfactorily.

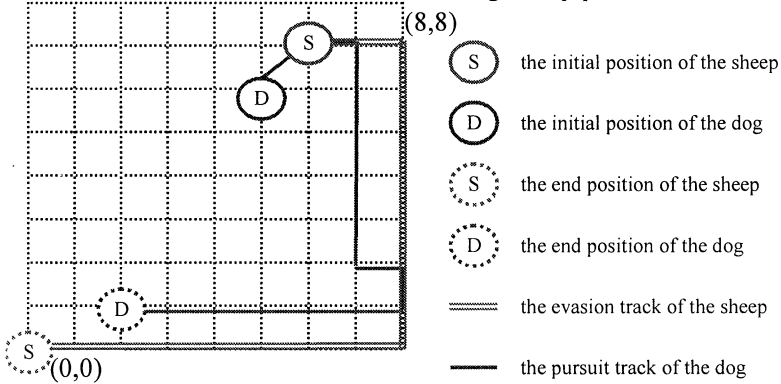


Figure 3: The pursuit-evasion tracks of the dog and the sheep.

4. CONCLUSIONS

In this paper we modeled an FNBDI and demonstrated how to apply it and how about the application by solving dog-sheep problem. The FNBDI extends the typical modal-logic-based BDI with fuzzy inference capability that is very useful for dealing with large-scale problems with uncertainties.

REFERENCES

1. Bratman M E. Intentions, Plans, and Practical Reason. Cambridge: Harvard University Press, 1987
2. Singh M P, Asher N M. Towards a formal theory of intentions. Proceedings of the European Workshop JELIA-90, Amsterdam, Netherlands, 1991: 472-486
3. Rao A S, Georgeff M P. Modelling Rational Agents within a BDI-Architecture. Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, USA, 1991:473-484
4. Poole D. The Independent Choice Logic for Modelling Multiple Agents Under Uncertainty. Artificial Intelligence, 1997,94(1-2): 5-56
5. Kaelbling L P, Littman M L, Cassandra A R. Planning and Acting in Partially Observable Stochastic Domains. Artificial Intelligence, 1998,101(1-2): 99-134
6. Xuan P, Lesser V R. Handling Uncertainty in Multi-Agent Commitments. UMASS Computer Science Technical Report, University of Massachusetts Amherst, 1999
7. Kachroo P, et al. Pursuit Evasion: The Herding Non-cooperative Dynamic Game. Journal of Integrated Design & Process Science, 2002,6(1): 31-42