1998 International Computer Symposium
Workshop on Artificial Intelligence
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

A Method to Learn Weights from Incomplete Data

Xudong Luo, Honghua Dai and Chengqi Zhang
School of Mathematical and Computer Sciences
The University of New England
E-mail: {xluo,hdai,chengqi}@turing.une.edu.au

Abstract

In fuzzy logic, the weight problem is very important. Some models have been established for solving the problem. However, the acquisition of weights in fuzzy logic is still a problem yet to be solved. Based on a class of relative weighted models, this paper provides a method to acquire weights from incomplete data.

1 Introduction

Recently, many publications have addressed the problem of learning rough set rules (or rough functions) and the learning of fuzzy functions[1; 2; 6; 7; 8]. Learning of weights is an important issue in both learning of rough set rules and the learning of fuzzy functions. When a proposition is composed of multiple sub-propositions, different sub-propositions are usually assumed to carry equal importance in traditional intelligent systems. However, this is not true when experts are interviewed, or the decision process is analyzed. In fact, different sub-propositions may have different (even quite different) influences on this composing proposition. Such a case is often encountered in practical applications. In order to capture different effects, people assign a weight to each sub-proposition. The more a sub-proposition has an effect on the composing proposition, the larger the weight is assigned to the sub-proposition. The question here is how to calculate the truth value of the composing proposition from the truth value of every sub-proposition and the corresponding weight. Some authors [5; 9; 10] have developed some models to cope with the problem. However, weights are difficulty to acquire or extract from domain experts [10]. In the present paper, we discuss this issue in the context of the relative weighted fuzzy logic [5].

The rest of this paper is organized as follows. Section 2 recaptures the basic notions of the relative weighted fuzzy logic [5]. In Section 3, we present a method to learn weights from incomplete data. Finally, conclusions are outlined in Section 4.

2 Background

In this section, we review the basic notions of the relative weighted fuzzy logic [5] for the convenience. These models are novel in three aspects: (1) they include non-weighted models as their special cases, (2) the weighted conjunction and weighted disjunction can be distinguished from each other, and (3) the information from all sub-propositions can be sufficiently considered.

2.1 T-norms and T-conorms

Dubois and Prade [3] have shown that T-norms and T-conorms are the most general families of binary functions, which respectively satisfy the requirements of the conjunction and disjunction operators, which are as follows:

Definition 2.1 *If the operator* $\circ : [0,1] \times [0,1] \longrightarrow [0,1]$ *satisfies the following conditions:*

1) *commutativity:* $a \circ b = b \circ a$;
2) *associativity:* $(a \circ b) \circ c = a \circ (b \circ c)$;
3) *monotonicity: if* $a \leq b$ *and* $c \leq d$ *then* $a \circ c \leq b \circ d$;
4) *boundary:* $a \circ 1 = a$,

where $a, b, c, d \in [0,1]$, *then* $\circ$ *is said to be a T-norm on* $[0,1]$, *denoted as* $\triangle$. *If* $\circ$ *satisfies 1), 2), 3) and*

4') *boundary:* $a \circ 0 = a$,

then $\circ$ *is said to be a T-conorm on* $[0,1]$, *denoted as* $\bigtriangledown$.

For example:

Zadeh operators $(\wedge, \vee)$	$a \wedge b = \min\{a, b\}$ $a \vee b = \max\{a, b\}$
Probability operators $(\bullet, \hat{+})$	$a \bullet b = a \times b$ $a \hat{+} b = a + b - a \times b$
Einstein operators $(\overset{\bullet}{E}, \overset{+}{E})$	$a \overset{\bullet}{E} b = \frac{ab}{1+(1-a)(1-b)}$ $a \overset{+}{E} b = \frac{a+b}{1+ab}$
Boundary operators $(\odot, \oplus)$	$a \odot b = \max\{0, a + b - 1\}$ $a \oplus b = \min\{1, a + b\}$

Although defined as two-place functions, the T-norms and T-conorms can be used respectively to process more than two sub-propositions in a composite proposition. Due to the associativity of the T-norms and T-conorms, it is possible to define recursively

$$\triangle(x_1, \cdots, x_n, x_{n+1})$$

and

$$\bigtriangledown(x_1, \cdots, x_n, x_{n+1})$$

for $x_1, \cdots, x_{n+1} \in [0,1]$, as

$$\triangle(x_1, \ldots, x_n, x_{n+1}) = \triangle(\triangle(x_1, \ldots, x_n), x_{n+1})$$
$$\bigtriangledown(x_1, \cdots, x_n, x_{n+1}) = \bigtriangledown(\bigtriangledown(x_1, \cdots, x_n), x_{n+1})$$

For the negation operation $N(x) = 1 - x$, T-norm $\triangle$ and T-conorm $\bigtriangledown$ are duals in the sense of the following DeMorgan's Law:

$$\triangle(a, b) = N(\bigtriangledown(N(a), N(b)))$$
$$\bigtriangledown(a, b) = N(\triangle(N(a), N(b)))$$

Therefore, T-norms and T-conorms should be given in the form of pairs which satisfy DeMorgan's Law.

2.2 Constraints on Weighted Fuzzy Logic

Consider the weighted conjunction

$$A = A_1(w_1) \wedge \cdots \wedge A_n(w_n) \quad (1)$$

and the weighted disjunction

$$B = A_1(w_1) \vee \cdots \vee A_n(w_n) \quad (2)$$

Obviously, the fuzzy truth values of A and B should be determined by functions which combine the fuzzy truth value with the corresponding weight, of each sub-proposition. Formally, we have

$$T(A) = f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n))$$
$$T(B) = f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n))$$

where $f_\wedge$ and $f_\vee$ are functions from $[0,1]^n$ to $[0,1]$, and g is a function from $[0,1]^2$ to $[0,1]$. For the negation of a weighted sub-proposition A_i with weight w_i, its truth value should also be determined by a function from $[0,1]^2$ to $[0,1]$:

$$T(\neg A_1) = f_\neg(T(A_1), w_1)$$

Now two questions arise: (i) What are the constraints on $f_\wedge$, $f_\vee$, $f_\neg$ and g? (ii) How do we construct $f_\wedge$, $f_\vee$, $f_\neg$ and g? In the following, we will answer the first question. The second question will be answered in the next subsection.

Constraint 1 *When* $w_1 = \cdots = w_n = \frac{1}{n}$,

$$f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = T(A_1) \triangle \cdots \triangle T(A_n)$$
$$f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = T(A_1) \bigtriangledown \cdots \bigtriangledown T(A_n)$$

This constraint reveals that a weighted model degenerates into a non-weighted model in fuzzy logic when each sub-proposition takes an equal weight, i.e. carries equal importance. In other words, their corresponding boundary conditions should fulfill the *AND* and *OR* operations in fuzzy logic without weights.

Constraint 2 *Thereāre*

$T(A_1), w_1, \cdots, T(A_n), w_n$ *in* $[0,1]$

such that

$$f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) \neq f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n))$$

This constraint reveals that the function $f_\wedge$ is not equivalent to the function $f_\vee$ since there is a difference between conjunction and disjunction.

Constraint 3 *Let* $w_i = \max\{w_1, \cdots, w_n\}$*, then*

$$T(A_i) = 0 \Rightarrow f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = 0$$
$$T(A_i) = 1 \Rightarrow f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = 1$$

This constraint means that, in a weighted conjunction, if the most important sub-proposition is false, the conjunction proposition should be false; in a weighted disjunction, if the most important sub-proposition is true, the disjunction proposition should be true. This constraint is a direct extension of the corresponding properties in a non-weighted model in fuzzy logic.

Constraint 4

$$f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = 1 - f_\vee(f_\neg(T(A_1), w_1), \cdots, f_\neg(T(A_n), w_n))$$
$$f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = 1 - f_\wedge(f_\neg(T(A_1), w_1), \cdots, f_\neg(T(A_n), w_n))$$

This constraint states that the weighted conjunction and weighted disjunction should satisfy DeMorgan's law.

Constraint 5 $f_\neg(T(A_i), w_i) + g(T(A_i), w_i) = 1$

In the above constraint, $f_\neg(T(A_i), w_i)$ can be viewed as the truth value of $\neg A_i$, and $g(T(A_i), w_i)$ can be viewed as the truth value of the sub-proposition A_i with the weight w_i. So this constraint is an extension of the corresponding property in fuzzy logic as well as in classical logic.

2.3 Relative Weighted Models

We define the relative weighted model as follows:

Definition 2.2 *Let* $(\triangle, \triangledown)$ *be a pair of T-norms and T-conorms which satisfy DeMorgan's Law. And let* $w = \max\{w_1, w_2, \cdots, w_n\}$.

1. *The truth value,* $T(A)$*, of weighted conjunction (1) is given by*

$$T(A) = (\frac{w_1}{w} \times T(A_1)) \triangle \cdots \triangle (\frac{w_n}{w} \times T(A_n)) \quad (3)$$

2. The truth value, $T(B)$, of weighted disjunction (5) is given by

$$T(B) = (\frac{w_1}{w} \times T(A_1)) \triangledown \cdots \triangledown (\frac{w_n}{w} \times T(A_n)) \quad (4)$$

3. The truth value, $T(\neg A_i)$, of the negation of weighted sub-proposition A_i is given by

$$T(\neg A_i) = 1 - \frac{w_i}{w} \times T(A_i) \quad (5)$$

The idea behind this definition is that after weighting w_i on A_i, its truth value is updated to $\frac{w_i}{w} \times T(A_i)$ from $T(A_i)$.

In [5], it is proved that this relative weighted model satisfies Constraints 1–5.

3 Learning Weight

For the formula (3) for calculating the truth value of weighted conjunction, suppose that we get m $(< n)$ instances

$$\begin{cases} T^{(1)}(A) = (\frac{w_1}{w} \times T^{(1)}(A_1)) \triangle \cdots \triangle (\frac{w_n}{w} \times T^{(1)}(A_n)) \\ T^{(2)}(A) = (\frac{w_1}{w} \times T^{(2)}(A_1)) \triangle \cdots \triangle (\frac{w_n}{w} \times T^{(2)}(A_n)) \\ \vdots \\ T^{(m)}(A) = (\frac{w_1}{w} \times T^{(m)}(A_1)) \triangle \cdots \triangle (\frac{w_n}{w} \times T^{(m)}(A_n)) \end{cases} \quad (6)$$

Additionally, we have the following equations

$$\begin{cases} \sum_1^n w_i = 1 \\ w = \max\{w_1, \cdots, w_n\} \end{cases} \quad (7)$$

Thus, if m=n-1, principally we can solve $w_1, \cdots, w_n$ from the equation arrays (6) and (7). This is because the number of equations is equal to the number of variables. If the number of equations is less than the number of variables, we cannot. However, we might give a reasonable approximate solution. In this section, we will discuss this problem.

For any instance $(T^{(i)}(A), T^{(i)}(A_1), \cdots, T^{(i)}(A_n))$, and the solution $(w_1, \cdots, w_n)$, clearly by (3) the following equation should be satisfied:

$$T^{(i)}(A) - (\frac{w_1}{w} \times T^{(i)}(A_1)) \Delta \cdots \Delta (\frac{w_n}{w} \times T^{(i)}(A_n)) = 0$$

If our approximate solution $(w_1, \cdots, w_n)$ cannot make the above equation satisfied, at least we should wish the left side of the above equation is minimum. Or, the function

$$f(w_1, \cdots, w_n)$$

$$= \sum_1^m (T^{(i)}(A) - (\frac{w_1}{w} \times T^{(i)}(A_1)) \Delta \cdots \Delta (\frac{w_n}{w} \times T^{(i)}(A_n)))^2 \quad (8)$$

should take the *minimum.* In other words, the problem of finding weight values turns to be a problem to find the minimum of the function.

Now we will solve the minimum problem. Firstly, because (6) and (7) we can use $w_{m+1}, \cdots, w_n$ to represent $w_1, \cdots, w_m$, that is,

$$\forall 1 \le j \le m, w_j = g_j(w_{m+1}, \cdots, w_n)$$

where g_j denotes a real function. Substitute these functions into (8), we obtain

$$f(w_1, \cdots, w_n)$$

$$= \sum_1^m (T^{(i)}(A) - (\frac{g_1(w_{m+1}, \cdots, w_n)}{w} \times T^{(i)}(A_1)) \Delta \cdots$$
$$\Delta(\frac{g_m(w_{m+1}, \cdots, w_n)}{w} \times T^{(i)}(A_m)) \Delta \cdots$$
$$\Delta(\frac{w_{m+1}}{w} \times T^{(i)}(A_{m+1})) \Delta \cdots \Delta (\frac{w_n}{w} \times T^{(i)}(A_n)))^2$$

Clearly, it is a function with the variables $w_{m+1}, \cdots, w_n$. So, for the convenience, we denote it as

$$g(w_{m+1}, \cdots, w_n)$$

$$= \sum_1^m (T^{(i)}(A) - (\frac{g_1(w_{m+1}, \cdots, w_n)}{w} \times T^{(i)}(A_1)) \Delta \cdots$$
$$\Delta(\frac{g_m(w_m, \cdots, w_n)}{w} \times T^{(i)}(A_m))$$
$$\Delta(\frac{w_{m+1}}{w} \times T^{(i)}(A_{m+1})) \Delta \cdots \Delta (\frac{w_n}{w} \times T^{(i)}(A_n)))^2 \quad (9)$$

Thus, the minimum problem of the function f turns to be that of the function g.

Secondly, we will find the critical points of the function $g(w_{m+1}, \cdots, w_n)$. That is to solve the following equation array:

$$\begin{cases} \frac{\partial g}{\partial w_{m+1}} = 0 \\ \frac{\partial g}{\partial w_{m+2}} = 0 \\ \vdots \\ \frac{\partial g}{\partial w_n} = 0 \end{cases} \quad (10)$$

Assume the set of all solution to this equation array is $SW = \{(w_{m+1}^{(i)}, \cdots, w_n^{(i)})\}$.

Finally, we select a critical point $(w_{m+1}^{(0)}, \cdots, w_n^{(0)})$ from the solution set SW so that the function g has a minimum at the critical point. According to mathematical theory of calculus [4], if the the following form satisfies

$$\sum_{m \le i,j \le n} \frac{\partial^2 g}{\partial w_i \partial w_j} u_i u_j > 0 \quad (11)$$

where $(u_{m+1}, \cdots, u_n)$ is a unit vector, i.e. $u_{m+1}^2 + \cdots + u_n^2 = 1$, and all derivatives are evaluated at the critical point, then g has a minimum at the critical point.

The learning weight problem of only weighted conjunction is involved in the discussion above for the sake of space. For weighted disjunction, the discussion is similar.

4 Conclusion

The relative weighted models presented in [5] are novel approaches capable of coping with weights in fuzzy logic in a sound and efficient manner. This paper, further, proposes a method to acquire weights from an incomplete date set. The key idea behind this approach is establishing a function related to all data, and then the problem to find a reasonable approximate solution is turn into the problem to find the minimum point of the function. Future work on this method will focus on the acceptance test to this function in various practical applications. We are going to apply it to some intelligent systems for finding the optimization form of this function.

Acknowledgement

This research is supported by the large grant from the Australian Research Council (A49530850) and the UNE research grant 7059 881 499.

References

[1] Honghua Dai. Field learning. In *Proceedings of the 19th Australian Computer Science Conference*, pages 55–63. Melbourne, Australia, Jan 31–Feb 2, 1996.

[2] Honghua Dai and Victor Ciesielski. Learning of inexact rules by the FISH-NET algorithm from low quality data. In *Proceedings of 7th Australian Joint Conference on Artificial Intelligence*, pages 108–115. World Scientific, Nov, 1994.

[3] D. Dubois & H. Prade (1984) Criteria Aggregation and Ranking of Alternatives in the Framework of Fuzzy Set Theory. *TIMS/Studies in the Management Science*, 20, Elsevier Science Publishers, pp. 209-240.

[4] W. Kaplan (1973) *Advanced Calculus.* Addison-Wesley Publishing Company, pp. 176–188.

[5] X. Luo and C. Zhang (1997) The Weighting Issue in Fuzzy Logic. *Informatica*, 21, pp.255–265.

[6] Maciej Modrzejewski. Feature selection using rough sets theory. *In: Pavel B. Brazdil(Ed.), Machine Learning: ECML-3, Proceedings of European Conference on Machine Learning*, pages 213–226, April 5-7, 1993.

[7] Zdzislaw Pawlak. Rough classification. *International Journal of Man-Machine Studies*, 20:469–181, 1984.

[8] Zdzislaw Pawlak, S. K. M. Wong, and Wojciech Ziarro. Rough sets: Probabilistic versus deterministic approach. *International Journal of Man-Machine Studies*, 29:81–95, 1988.

[9] R. R. Yager (1988) On Ordered Weighted Averaging Aggregation Operators in Multi-Criteria Decision Making. *IEEE Trans. on Systems, Man, and Cybernetics*, 18, pp. 183–190.

[10] D. S. Yeung and E. C. C. Tsang (1997) Weighted Fuzzy Production Rules. *Fuzzy Sets and System*, 88, pp. 299–313.