Journal of Uncertain Systems
Vol.7, No.3, pp.179-180, 2013
Online at: www.jus.org.uk

$1 - x$ is the Only Polynomial Fuzzy Negation Operation

Jaime Nava*
Hewlett Packard, El Paso, Texas, USA

Received 1 April 2012; Revised 16 March 2013

Abstract

In fuzzy logic, our degrees of belief in different statements are usually described by numbers from the interval $[0,1]$. In principle, we may have the same degree of belief in a statement A but different degrees of belief in the negation $\neg A$. Ideally, we should be able to elicit the expert's degree of belief in different propositional combinations of their statements. In practice, this is not always feasible, so it is reasonable to estimate the degree of belief $d(\neg A)$ in $\neg A$ based on the degree of belief $d(A)$ in A. The operation that transforms the degree $a = d(A)$ into an estimate for $d(\neg A)$ is known as *fuzzy negation* and denoted by $f_\neg(a)$. The most widely used negation operation is $f_\neg(a) = 1 - a$. Since $\neg\neg A$ is equivalent to A, it is reasonable to require that by following this procedure, we assign to $\neg\neg A$ the same degree as to the original statement A, i.e., that $f_\neg(f_\neg(a)) = a$ for all a. We prove that $f_\neg(a) = 1 - a$ is the only polynomial operation which satisfies this property.

Keywords: fuzzy logic, fuzzy negation, polynomial operation

1 Background on Fuzzy Logic

Why fuzzy logic. In many applications, it is important to use *expert knowledge*. Experts often describe their knowledge in *imprecise* ("*fuzzy*") properties like "small". For example, for a specific size, an expert may be not fully confident whether this size is small.

To describe such imprecise properties, *fuzzy logic* was invented. In fuzzy logic, each statement is characterized by a *degree* of confidence. Usually, this degree is taken from the interval $[0,1]$, where 0 means absolutely false, 1 means absolutely true, and intermediate values represent uncertainty.

Fuzzy logic operations. In many applications conditions to expert rules are composite statements, e.g., "the car in front is close, *and* the road is slippery." To formalize such conditions we need to know the degrees of certainty in such composite statements. Ideally, we should elicit these degrees from the experts. However, in practice, we may have many propositional combinations of the original statements and therefore, it is not possible to elicit the expert's degree of certainty in all these combinations. In such situations, we need to approximate the expert's degree of certainty $d(A \,\&\, B)$ in a composite statement based only on the known degrees of certainty $d(A)$ and $d(B)$ in the corresponding statements A and B.

For conjunction, the algorithm for this estimation transforms two numbers $d(A)$ and $d(B)$ into the estimate for $d(A \,\&\, B)$. This estimate is usually denoted by $f_\&(d(A), d(B))$. The corresponding function $f_\&(a,b)$ is called an *"and"-operation* or a *t-norm*.

For disjunction, the algorithm for this estimation transforms two numbers $d(A)$ and $d(B)$ into the estimate for $d(A \vee B)$. This estimate is usually denoted by $f_\vee(d(A), d(B))$. The corresponding function $f_\vee(a,b)$ is called an *"or"-operation* or a *t-conorm*.

For negation, the algorithm for this estimation transforms one number $d(A)$ into the estimate for $d(\neg A)$. This estimate is usually denoted by $f_\neg(d(A))$. The corresponding function $f_\neg(a)$ is called a *negation* operation.

In summary, we have $d(A \,\&\, B) \approx f_\&(d(A), d(B))$, $d(A \vee B) \approx f_\vee(d(A), d(B))$, and $d(\neg A) \approx f_\neg(d(A))$. The functions $f_\&(a,b)$, $f_\vee(a,b)$, and $f_\neg(a)$ providing such estimates are called *fuzzy logic operations*.

Fuzzy logic operations must satisfy natural properties. For example, since $A \,\&\, B$ means the same as $B \,\&\, A$, the fuzzy "and"-operation $f_\&(a,b)$ must be commutative, i.e., $f_\&(a,b) = f_\&(b,a)$. Also, since $A \,\&\, (B \,\&\, C)$

*Corresponding author. Email: jenava@miners.utep.edu (J. Nava).

means the same as $(A \,\&\, B) \,\&\, C$, the fuzzy "and"-operation $f_\&(a,b)$ must be associative, i.e., $f_\&(a, f_\&(b,c)) = f_\&(f_\&(a,b),c)$. There exist complete descriptions of all the operations that satisfy such properties [1, 2].

2 New Result About Polynomial Fuzzy Negation Operations

Negation operations: usual properties. The main algebraic property of the negation operation is that $\neg(\neg A)$ means the same as A. Thus the fuzzy negation operation $f_\neg(a)$ shall satisfy the property:

$$f_\neg(f_\neg(a)) = a. \tag{1}$$

The fuzzy negation operation shall also satisfy monotonicity, that is, the more we believe in A, the less we believe in $\neg A$. We conclude that the function $f_\neg(a)$ shall be *non-increasing*.

The fuzzy negation operation shall also be *conservative*, that is, for $a = 0$ ("false") and for $a = 1$ ("true"), $f_\neg(a)$ must coincide with the truth value of "not a":

$$f_\neg(0) = 1, \quad f_\neg(1) = 0.$$

Description of all polynomial negation operations. Consider polynomial functions $f_\neg : [0,1] \to [0,1]$:

$$f_\neg(a) = c_0 + c_1 \cdot a + \cdots + c_k \cdot a^k.$$

Proposition. *The only conservative polynomial negation operation which satisfies the property (1) is*

$$f_\neg(a) = 1 - a.$$

Comment. Please note that we do not require monotonicity.

Proof. In general, we have

$$f_\neg(a) = c_k \cdot a^k + \text{ lower order terms.}$$

Thus,

$$f_\neg(f_\neg(a)) = c_k \cdot (f_\neg(a))^k + \text{ lower order terms.}$$

Substituting the above expression for $f_\neg(a)$ into this formula, we conclude that

$$f_\neg(f_\neg(a)) = c_k \cdot (c_k \cdot a^k + \text{ lower order terms})^k + \text{ lower order terms,}$$

hence

$$f_\neg(f_\neg(a)) = c_k^{k+1} \cdot a^{k^2} + \text{ lower order terms.}$$

The highest-order term in this formula is thus a term of order k^2. The requirement (1) means that $f_\neg(f_\neg(a)) = a$, so the highest order is 1; thus, $k^2 = 1$ hence $k = 1$, and therefore, the negation operation is a linear function: $f_\neg(a) = c_0 + c_1 \cdot a$. For this linear expression, the requirement that $f_\neg(a) = 1$ leads to $c_0 = 1$, and for the resulting expression $f_\neg(a) = 1 + c_1 \cdot a$, the requirement that $f_\neg(1) = 0$ leads to $c_1 = -1$. The statement is proven.

References

[1] Klir, G.J., and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[2] Nguyen, H.T., and E.A. Walker, *A First Course in Fuzzy Logic*, 3rd Edition, Boca Raton, Florida, Chapman & Hall Press, 2006.