

# Simple Linear Interpolation Explains All Usual Choices in Fuzzy Techniques: Membership Functions, t-Norms, t-Conorms, and Defuzzification

Vladik Kreinovich, Jonathan Quijas,  
Esthela Gallardo, Caio De Sa Lopes  
Department of Computer Science  
University of Texas at El Paso  
500 W. University  
El Paso, TX 79968, USA  
vladik@utep.edu, jkuijas@miners.utep.edu  
egallardo5@miners.utep.edu  
cdesalopes@miners.utep.edu

Olga Kosheleva  
Department of Teacher Education  
University of Texas at El Paso  
500 W. University  
El Paso, TX 79968, USA  
olgak@utep.edu

Shahnaz Shahbazova  
Azerbaijan Technical University  
Baku, Azerbaijan  
shahbazova@gmail.com

**Abstract**—Most applications of fuzzy techniques use piece-wise linear (triangular or trapezoid) membership functions, min or product t-norms, max or algebraic sum t-conorms, and centroid defuzzification. Similarly, most applications of interval-valued fuzzy techniques use piecewise-linear lower and upper membership functions. In this paper, we show that all these choices can be explained as applications of simple linear interpolation.

## I. FORMULATION OF THE PROBLEM

**Fuzzy techniques are needed.** In many application areas, we have experts whose experience we would like to capture. Often, experts can express their experience in terms of rules, but the rules they describe use imprecise (“fuzzy”) words from natural language, like “small”, “large”, etc. To formalize these rules, L. Zadeh proposed special *fuzzy techniques* [1], [6], [7].

**Fuzzy techniques: reminder.** The usual use of fuzzy techniques consists of three stages.

**First stage of fuzzy techniques: eliciting a membership function.** On the first stage, we formalize the original imprecise terms like “small”. Each such term is described by assigning, to different possible values  $x$  of the corresponding quantity, a degree  $\mu(x)$  to which this value  $x$  satisfies this property (e.g., the degree to which  $x$  is small).

Some of these values  $\mu(x)$  are obtained by asking the expert. However, there are infinitely many real numbers  $x$ , and we can only ask an expert a finite number of questions. Thus, after eliciting, from expert, several values  $\mu(x_i)$ , we need to perform some *interpolation* to estimate the degrees  $\mu(x)$  for intermediate values  $x$ . The resulting function  $\mu$  is known as the *membership function*.

**Interval-valued fuzzy degrees.** Often, an expert is unable to describe his or her degree of confidence by an exact number; a more natural approach is to use *intervals* of possible values [2], [3], [5].

In this case, for each possible value of a quantity  $x$ , instead of a single value  $\mu(x)$ , we get an *interval*  $[\underline{\mu}(x), \overline{\mu}(x)]$  of possible values. Determining such interval-valued membership function is equivalent to determining the functions  $\underline{\mu}(x)$  and  $\overline{\mu}(x)$ . These functions are known as the *lower* and the *upper* membership functions.

**Second stage of fuzzy techniques: “and”- and “or”-operations.** Once we have elicited the membership functions, we know to what extent, e.g., the given value of the body temperature corresponds to high fever. However, many expert rules involve several conditions. For example, since some medicines increase blood pressure, a medical doctor will probably:

- prescribe the most efficient medicine if the fever is high *and* blood pressure is normal, and
- prescribe a different medicine – not so efficient but without the negative side effects – if the fever is high *and* blood pressure is high.

To handle such rules, we need to be able to transform the degrees  $a = d(A)$  and  $b = d(B)$  of individual conditions  $A$  and  $B$  into a degree of confidence in the composite statement  $A \& B$ . The corresponding estimate  $f_{\&}(a, b)$  is known as an “*and*”-operation, or, alternatively, as a *t-norm*.

Similarly, we need an “*or*”-operation  $f_{\vee}(a, b)$  (also known as *t-conorm*) to estimate the expert’s degree of confidence in  $A \vee B$  and the *negation operation*  $f_{\neg}(a)$  to estimate the expert’s degree of confidence in  $\neg A$ .

**Third stage of fuzzy techniques: defuzzification.** After performing the first two stages, for the given input  $x$  and for all possible control values  $u$ , we get a degree  $\mu(u)$  to which this control value is reasonable to apply.

If we want to use this expert knowledge in an automated system, we need to transform this membership function  $\mu(u)$  into a single value  $\bar{u}$ .

**In principle, many versions of fuzzy techniques are possible.** From the purely mathematical viewpoint, we can think of many different membership functions, many different “and”- and “or”-operations, many different defuzzification procedures.

**How to select an appropriate technique?** To select an appropriate combination of fuzzy techniques, a natural idea is to try different combinations and to see which combination leads to the best result.

**Interesting empirical fact: in many cases, the same few combinations of fuzzy techniques lead to the best results.** Interestingly, it turns out that in many practical applications, the same small number of combinations turn out to be most efficient. Specifically, in these applications, we use:

- *trapezoid* membership functions that start with 0, linearly got to 1, stay at 1, and then linearly decrease back to 0; in particular, when the stay-at-1 interval degenerates into a single point, we get *triangular* membership functions;
- “and”-operations  $f_{\&}(a, b) = \min(a, b)$  (known as *min*) or  $f_{\&}(a, b) = a \cdot b$  (known as *algebraic product*);
- “or”-operations  $f_{\vee}(a, b) = \max(a, b)$  (known as *max*) or  $f_{\vee}(a, b) = a + b - a \cdot b$  (known as *algebraic sum*);
- negation operation  $f_{\neg}(a) = 1 - a$ ; and
- *centroid defuzzification*

$$\bar{u} = \frac{\int u \cdot \mu(u) du}{\int \mu(u) du}.$$

In the interval-valued case, similarly, in many case, both lower and upper membership functions are trapezoidal.

**What we do in this paper.** In this paper, we show that all these choices can be explained in a very simple way – by the use of the simplest (linear) interpolation.

**How this paper is structured.** In the following sections, we will explain these derivations for all three stages

**Our result is also useful for teaching fuzzy techniques.** Our explanation not only helps us understand why all these choices are made, it makes it easier to *teach* the basic fuzzy techniques – since now all these techniques can be derived from the same linear interpolation idea.

## II. INTERPOLATION: WHY WE NEED IT, AND WHY LINEAR INTERPOLATION IS THE SIMPLEST

**Need for interpolation.** In the above text, we have mentioned that we need interpolation to elicit a membership function from an expert.

Similarly, we can elicit the expert’s degrees of confidence  $a_k$ ,  $b_k$ , and  $c_k$  in different statements  $A_k$ ,  $B_k$ , and  $A_k \& B_k$ . Then, we should get an “and”-operation  $f_{\&}(a, b)$  for which  $f_{\&}(a_k, b_k) = c_k$ . This will enable us to find the values  $f_{\&}(a, b)$  for a finite number of pairs  $(a_k, b_k)$ . To estimate the values  $f_{\&}(a, b)$  for all other pairs, we need to use interpolation.

A similar interpolation is needed to determine the “or”-operation  $f_{\vee}(a, b)$ , the negation operation  $f_{\neg}(a, b)$ , and the appropriate defuzzification.

**In the 1-D case, linear interpolation is the simplest.** Interpolation means that we find a curve that goes through given points, or, equivalently, a function that attains known values at given points.

The simplest possible functions are constants. The simplest possible non-constant functions are linear functions. When we use linear functions to interpolate, we get linear interpolation – which is, thus, the simplest possible interpolation.

To be more precise, if for two values  $x_1 < x_2$ , we know the values  $y_1 = f(x_1)$  and  $y_2 = f(x_2)$ , then these two values uniquely determine a linear function:

$$f(x) = f(x_1) + \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1). \quad (1)$$

**What we do in this paper: reminder.** In this paper, we show that this simplest (linear) interpolation explains all usual choices of fuzzy techniques.

## III. SIMPLE LINEAR INTERPOLATION EXPLAINS THE USUAL CHOICE OF MEMBERSHIP FUNCTIONS

**Main idea.** Usually, for each property like “small”,

- first, there are some values which are definitely not small (e.g., negative ones),
- then some values which are small to some extent;
- then, we have an interval of values which are definitely small;
- this is followed by values which are somewhat small; and
- finally, we get values which are absolutely not small.

Let us denote the threshold separating these regions by  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$ . In terms of these thresholds, we have the following:

- for all the values  $x \leq t_1$ , the desired property is not satisfied, so we get  $\mu(x) = 0$ ;
- for the values  $x$  between  $t_1$  and  $t_2$ , the property is satisfied to some degree, so we have  $0 < \mu(x) < 1$ ;
- for the values  $x$  between  $t_2$  and  $t_3$ , the property is absolutely satisfied, so we have  $\mu(x) = 1$ ;
- for the values  $x$  between  $t_3$  and  $t_4$ , the property is satisfied to some degree, so we have  $0 < \mu(x) < 1$ ;
- finally, for  $x \geq t_4$ , the desired property is not satisfied, so we get  $\mu(x) = 0$ .

In this description, we know the values  $\mu(x)$  for  $x \leq t_1$ , for  $x \in [t_2, t_3]$ , and for  $x \geq t_4$ . What we need is estimate the values  $\mu(x)$  for  $x \in [t_1, t_2]$  and  $x \in [t_3, t_4]$ .

**Let us use linear interpolation.** To find the values  $\mu(x)$  for  $x \in [t_1, t_2]$ , we can apply linear interpolation to the known

values  $\mu(t_1) = 0$  and  $\mu(t_2) = 1$ . As a result, we get a linear increasing function on this interval.

Similarly, to find the values  $\mu(x)$  for  $x \in [t_3, t_4]$ , we can apply linear interpolation to the known values  $\mu(t_3) = 1$  and  $\mu(t_4) = 0$ . As a result, we get a linear decreasing function on this interval.

As a result of this interpolation, we get a trapezoid membership function; see, e.g., [4].

**Conclusion.** Simple linear interpolation explains the usual choice of a trapezoid membership function.

#### IV. SIMPLE LINEAR INTERPOLATION EXPLAINS THE USUAL CHOICE OF T-NORMS

**Main idea.** What do we know about “and”?

Intuitively, if one of the component statements  $A$  is false, then the composite statement  $A \& B$  is also false. In other words, if  $a = d(A) = 0$ , then  $f_{\&}(a, b) = f_{\&}(0, b) = 0$ .

Also, if  $A$  is absolutely true, then our belief in  $A \& B$  is equivalent to our degree of belief in  $B$ . In other words, if  $a = d(A) = 1$ , then  $f_{\&}(1, b) = b$ .

We thus know the values  $f_{\&}(a, b)$  for the cases when  $a$  is equal to 0 or 1. We would like to use this information to estimate the value  $f_{\&}(a, b)$  for all possible pairs  $(a, b)$ .

**Let us use linear interpolation.** Let us fix  $b$  and consider a function  $F_b(a) \stackrel{\text{def}}{=} f_{\&}(a, b)$  that maps  $a$  into the value  $f_{\&}(a, b)$ . We know that:

- for  $a = 0$ , we have  $F_b(0) = 0$ , and that
- for  $a = 1$ , we have  $F_b(1) = b$ .

We can use linear interpolation to find the value  $F_b(a)$  for all  $a \in [0, 1]$ . The general formula for linear interpolation leads to  $F_b(a) = a \cdot b$ , i.e., to the algebraic product  $f_{\&}(a, b) = a \cdot b$ .

*Comment.* Please note that while the resulting operation is commutative and associative, we did not require commutativity or associativity: all we required was linear interpolation.

**What if we additionally require that  $A \& A$  is equivalent to  $A$ .** Another intuitive property of “and” is that for every statement  $A$ , “ $A$  and  $A$ ” means the same as  $A$ . In terms of the “and”-operation, this means that for every  $a = d(A)$ , we have  $f_{\&}(a, a) = a$ .

Now, for each  $b$ , in addition to knowing the values  $f_{\&}(a, b)$  for the values  $a = 0$  and  $a = 1$ , we also know its value for  $a = b$ .

**What if we also require that  $A \& A$  is equivalent to  $A$ : let us use linear interpolation.** Let us fix  $b$  and consider a function  $F_b(a) = f_{\&}(a, b)$  that maps  $a$  into the value  $f_{\&}(a, b)$ .

We know that  $F_b(0) = f_{\&}(0, b) = 0$  and that  $F_b(b) = f_{\&}(b, b) = b$ . Thus, on the interval  $[0, b]$ , linear interpolation leads to  $F_b(a) = a$ , i.e., to  $f_{\&}(a, b) = a$ .

Let us now apply linear interpolation to the interval  $[b, 1]$ . From  $F_b(b) = b$  and  $F_b(1) = f_{\&}(1, b) = b$ , we conclude that for the values  $a$  from this interval, we have  $F_b(a) = b$ , i.e., we have  $f_{\&}(a, b) = b$ .

These two cases lead us to the following conclusion:

- when  $a \leq b$ , then  $f_{\&}(a, b) = a$ ;
- when  $b \leq a$ , then  $f_{\&}(a, b) = b$ .

These two cases can be described by a single formula  $f_{\&}(a, b) = \min(a, b)$ . This is exactly the min-operation.

**Conclusion.** Simple linear interpolation explains the usual two choice of an “and”-operation: algebraic product and min.

#### V. SIMPLE LINEAR INTERPOLATION EXPLAINS THE USUAL CHOICE OF T-CONORMS

**Main idea.** What do we know about “or”?

Intuitively, if one of the component statements  $A$  is absolutely true, then the composite statement  $A \vee B$  is also absolutely true. In other words, if  $a = d(A) = 1$ , then  $f_{\vee}(a, b) = f_{\vee}(1, b) = 1$ .

Also, if  $A$  is absolutely false, then our belief in  $A \vee B$  is equivalent to our degree of belief in  $B$ . In other words, if  $a = d(A) = 0$ , then  $f_{\vee}(0, b) = b$ .

We thus know the values  $f_{\vee}(a, b)$  for the cases when  $a$  is equal to 0 or 1. We would like to use this information to estimate the value  $f_{\vee}(a, b)$  for all possible pairs  $(a, b)$ .

**Let us use linear interpolation.** Let us fix  $b$  and consider a function  $G_b(a) \stackrel{\text{def}}{=} f_{\vee}(a, b)$  that maps  $a$  into the value  $f_{\vee}(a, b)$ . We know that:

- for  $a = 0$ , we have  $G_b(0) = b$ , and that
- for  $a = 1$ , we have  $G_b(1) = 1$ .

We can use linear interpolation to find the value  $F_b(a)$  for all  $a \in [0, 1]$ . The general formula for linear interpolation leads to  $G_b(a) = b + a \cdot (1 - b)$ , i.e., to the algebraic sum  $f_{\vee}(a, b) = a + b - a \cdot b$ .

*Comment.* Please note that, similarly to the case of “and”-operations, while the resulting operation is commutative and associative, we did not require commutativity or associativity: all we required was linear interpolation.

**What if we additionally require that  $A \vee A$  is equivalent to  $A$ .** Another intuitive property of “or” is that for every statement  $A$ , “ $A$  or  $A$ ” means the same as  $A$ . In terms of the “or”-operation, this means that for every  $a = d(A)$ , we have  $f_{\vee}(a, a) = a$ .

Now, for each  $b$ , in addition to knowing the values  $f_{\vee}(a, b)$  for the values  $a = 0$  and  $a = 1$ , we also know its value for  $a = b$ .

**What if we also require that  $A \vee A$  is equivalent to  $A$ : let us use linear interpolation.** Let us fix  $b$  and consider a function  $G_b(a) = f_{\vee}(a, b)$  that maps  $a$  into the value  $f_{\vee}(a, b)$ .

We know that  $G_b(0) = f_\vee(0, b) = b$  and that  $G_b(b) = f_\vee(b, b) = b$ . Thus, on the interval  $[0, b]$ , linear interpolation leads to  $F_b(a) = b$ , i.e., to  $f_\vee(a, b) = b$ .

Let us now apply linear interpolation to the interval  $[b, 1]$ . From  $G_b(b) = b$  and  $G_b(1) = f_\vee(1, b) = 1$ , we conclude that for the values  $a$  from this interval, we have  $G_b(a) = a$ , i.e., we have  $f_\vee(a, b) = a$ .

These two cases lead us to the following conclusion:

- when  $a \leq b$ , then  $f_\vee(a, b) = b$ ;
- when  $b \leq a$ , then  $f_\vee(a, b) = a$ .

These two cases can be described by a single formula  $f_\vee(a, b) = \max(a, b)$ . This is exactly the max-operation.

**Conclusion.** Simple linear interpolation explains the usual two choice of an “or”-operation: algebraic sum and max.

## VI. SIMPLE LINEAR INTERPOLATION EXPLAINS THE USUAL CHOICE OF NEGATION OPERATIONS

**Formulation of the problem.** For the classical 2-valued logic, with two truth values ‘ (“true”) and 0 (“false”), the negation operation is easy to define:

- the negation of “false” is “true”:  $f_-(0) = 1$ , and
- the negation of “true” is “false”:  $f_-(1) = 0$ .

We want to extend this operation from the 2-valued set  $\{0, 1\}$  to the whole interval  $[0, 1]$ .

**Let us use linear interpolation.** In this case, in terms of the formula (1), we have  $x_1 = 0$ ,  $x_2 = 1$ ,  $y_1 = 1$ , and  $y_2 = 0$ . Thus, the formula for linear interpolation leads to

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

This is exactly the most frequently used negation operation in fuzzy logic.

**Conclusion.** Simple linear interpolation explains the usual choice of negation operation.

## VII. SIMPLE LINEAR INTERPOLATION EXPLAINS THE USUAL CHOICE OF DEFUZZIFICATION

**Main idea.** The desired control  $\bar{u}$  should be close to reasonable control values  $u$ :  $\bar{u} \approx u$ . In principle, we have different possible control values  $u$ .

Let us start with a simplified situation in which we have finitely many reasonable values  $u_1, \dots, u_k$ , and all these values are equally reasonable. In this case, we want to find the values  $\bar{u}$  for which  $\bar{u} \approx u_1, \bar{u} \approx u_2, \dots, \bar{u} \approx u_k$ .

Since the values  $u_i$  are different, we cannot get the exact equality in all  $k$  cases, i.e., we cannot get all  $k$  approximation errors  $e_k \stackrel{\text{def}}{=} \bar{u} - u_k$  to 0. What we want is to make the vector  $e \stackrel{\text{def}}{=} (e_1, \dots, e_k)$  consisting of all these errors to be as close to the ideal point  $(0, \dots, 0)$  as possible.

The distance between the vector  $e$  and the 0 point is equal to  $\sqrt{e_1^2 + \dots + e_k^2}$ . Minimizing this distance is equivalent to minimizing its square, i.e., the sum

$$e_1^2 + \dots + e_k^2 = (\bar{u} - u_1)^2 + \dots + (\bar{u} - u_k)^2.$$

This is the usual *Least Squares* method.

In the continuous case, when all the value  $u$  are possible, in the limit, we get an integral instead of the sum  $\int (\bar{u} - u)^2 du$ .

This method works well if all the values  $u$  are equally possible. In reality, different values  $u$  have different degrees of possibility  $\mu(u)$ .

- If the value  $u$  is fully possible, i.e., if  $\mu(u) = 1$ , then we should keep the corresponding term  $(\bar{u} - u)^2$  in the sum.
- If the value  $u$  is completely impossible, i.e., if  $\mu(u) = 0$ , then we should not consider this term at all.

In general, a natural idea is, instead of simply adding the squares, to first multiply each square by a weight  $w(\mu(u))$  depending on the degree of possibility, so that:

- when  $\mu(u) = 1$ , we get full weight  $w(\mu(u)) = w(1) = 1$ ;
- when  $\mu(u) = 0$ , we get zero weight  $w(\mu(u)) = w(0) = 0$ ; and
- for values  $\mu(u)$  intermediate between 0 and 1, we should use intermediate values of weight.

In the discrete case, we thus minimize the sum

$$\sum_i w(\mu(i)) \cdot (\bar{u} - u_i)^2;$$

in the continuous case, we minimize the corresponding integral

$$\int w(\mu(u)) \cdot (\bar{u} - u)^2 du. \quad (2)$$

**Let us use linear interpolation.** We know that  $w(0) = 0$  and  $w(1) = 1$ . We want to estimate the values  $w(\mu)$  for all  $\mu \in [0, 1]$ . Linear interpolation leads to  $w(u) = u$ .

Thus, we should select  $\bar{u}$  for which the integral

$$\int \mu(u) \cdot (\bar{u} - u)^2 du \quad (3)$$

attains the smallest possible value.

Differentiating the expression (3) with respect to  $\bar{u}$  and equating the derivative to 0, we conclude that

$$\int 2\mu(u) \cdot (\bar{u} - u) du = 0. \quad (4)$$

Dividing both sides of this equality by 2 and moving all the terms not depending on  $\bar{u}$  to the right-hand side, we conclude that

$$\bar{u} \cdot \int \mu(u) du = \int u \cdot \mu(u) du,$$

and thus, that

$$\bar{u} = \frac{\int u \cdot \mu(u) du}{\int \mu(u) du}.$$

This is exactly the centroid defuzzification which is efficiently used in many practical applications. ion operation in fuzzy logic.

**Conclusion.** Simple linear interpolation explains the usual choice of defuzzification.

#### ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721.

#### REFERENCES

- [1] G. Klir and B. Yuan, "Fuzzy Sets and Fuzzy Logic", Prentice Hall, Upper Saddle River, New Jersey, 1995.
- [2] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, Upper Saddle River, 2001.
- [3] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*, IEEE Press and Wiley, New York, 2010.
- [4] H. T. Nguyen and V. Kreinovich, *Applications of Continuous Mathematics to Computer Science*, Kluwer, Dordrecht, 1997.
- [5] H. T. Nguyen, V. Kreinovich, and Q. Zuo, "Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control", *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS)*, 1997, Vol. 5, No. 3, pp. 317–358.
- [6] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2006.
- [7] L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.