Discrete (Set) Derivatives and "Algebraic" Fuzzy Logic Operations

Bernadette Bouchon-Meunier
LIP6, Pole IA, UPMC, 75252 Paris, France, bouchon@poleia.lip6.fr

Hung T. Nguyen
Mathematics, New Mexico State University, Las Cruces, NM 88003, USA, hunguyen@nmsu.edu

Vladik Kreinovich
Computer Science, University of Texas, El Paso, TX 79968, USA, vladik@cs.utep.edu

***Abstract*—We propose a new way to generalize logical operations from the discrete classical logic to a continuous fuzzy logic; namely, we propose to define derivatives for discrete case, and then to use these derivatives to derive the continuous operations. We show that this natural approach leads to "algebraic" fuzzy operations $a \cdot b$ and $a + b - a \cdot b$.**

I. Introduction

The traditional fuzzy logic, in which the set of truth values is the interval $[0, 1]$, is a natural generalization of the classical (2-valued) logic in which the set of truth values $\{0, 1\} = \{F, T\}$ consists of only two elements:

- $0 = F =$"false", and
- $1 = T =$"true".

There are many different ways of generalizing logical operations from the (discrete) 2-valued set $\{0, 1\}$ to the continuous interval $[0, 1]$; see, e.g., [2], [7]. Most of these generalizations come from selecting some reasonable properties of a 2-valued logic and requiring that these same properties hold for the continuous logic as well. For example, if we require that the property $A \,\&\, A = A$ be preserved for an "and"-operation (t-norm) $f_{\&} : [0, 1] \times [0, 1] \to [0, 1]$, then this requirement uniquely determines the operation $f_{\&}(a, b) = \min(a, b)$.

In this traditional approach, we generalize the properties of the discrete-valued logic to the continuous case. Thus, the generalization goes in the direction from the discrete case to the continuous case.

In this paper, we propose a new approach to such a generalization, an approach in which the generalization goes (in some reasonable sense) in *both* directions:

- in the usual direction, from discrete to continuous, and also
- in the opposite direction: from the continuous to the discrete case.

Specifically, we do the following:

- We start with the notion which is natural to define in the continuous case but which is difficult to define for the original discrete logic – the notion of the derivative.
- Then, we define a natural generalization of this continuous notion to the discrete case. This is the step where we generalize from the continuous to the discrete case.
- We use the definition of the "generalized" derivative to determine the derivatives of the standard 2-valued logical operations.
- Next, we generalize in the usual direction: namely, we require that the continuous generalizations of these logical operations have exactly the same values of derivatives.

From the continuous viewpoint, this approach makes perfect sense, because most continuous processes – in physics, in chemistry, etc. – are described by *differential equations*, when we first determine the values of the derivatives and then then use these values to reconstruct the desired functions.

We are doing, in effect, the same thing for fuzzy logic operations: we first find the appropriate differential equations for these operations, and then we solve ("integrate") these equations to get the explicit formulas.

II. A Natural Generalization of a Derivative to the Discrete (2-Valued) Case

The derivative of a function $f(t)$ is defined as a limit

$$\frac{df}{dt} \stackrel{\text{def}}{=} \lim_{t_1 \to t, t_2 \to t, t_1 < t_2} \frac{f(t_2) - f(t_1)}{t_2 - t_1}. \tag{1}$$

The limit means that the closer the values t_1 and t_2 to t, the more accurately the ratio in the right-hand side of the formula (1) describes the desired derivative. Thus, in practice, to estimate the value of the derivative as accurately as possible, we select the values $t_1 < t_2$ to be as close to t as possible, and use the corresponding ratio as the desired estimate:

$$\widetilde{\frac{df}{dt}} = \frac{f(t_2) - f(t_1)}{t_2 - t_1}. \tag{2}$$

It is reasonable to use this same idea to define the derivative in the discrete 2-valued case, when the variable t only takes two possible values: 0 and 1. From the viewpoint of this definition, the main difference between the continuous and the discrete cases is as follows:

- In the continuous case, we can take the values t_i as close to t as possible. No matter how close to t the values t_i are, the resulting ratio (2) is only an approximation to the actual derivative, and we need the limit operation to define the exact value of the derivatives.

- In the 2-valued case, there are only two possible values of t, so if we still want to use the definition (2), our only choice is to use $t_1 = 0$ an $t_2 = 1$.

In this case, $t_2 - t_1 = 1$, so (2) turns into the following definition of the discrete derivative:

$$\frac{\mathcal{D}f}{\mathcal{D}t} \stackrel{\text{def}}{=} f(1) - f(0). \tag{3}$$

This natural formula looks trivial, but it is actually very useful in discrete optimization – just like the normal derivative is useful in continuous optimization. Let us just give two references:

- In [3], it is shown how this discrete derivative can be used to design efficient gradient-type heuristic techniques for solving discrete optimization problems (which are usually difficult to solve – NP-hard, to be more precise).
- In [5], it is shown how a similar discrete derivative can help to optimize functions of *sets*, i.e., functions $f(t)$ in which the variable t can take, as values, different (crisp) subsets. We can, of course, make this problem closer to the continuous case by considering, instead of a set, its characteristic function. The values of the characteristic function are numbers $t(x)$ corresponding to different points x, so it looks almost like a normal differentiation problem. However, for each x, the characteristic function can only take two values: 0 and 1, so we face the same problem of defining a derivative in the discrete case.

A similar notion of a discrete derivative turns out to be very useful in a more general granulation framework, when a function $f(t)$ is only defined on (fuzzy) granules t, and we want to estimate the rate with which it increases (i.e., its derivative); see, e.g., [1].

Let us show how, with this discrete differentiation operation, we can use the above methodology to select the fuzzy logic operations.

III. First Example: Selecting Fuzzy Negation

Let us start with the simplest fuzzy logic operation: negation $f_\neg(x)$. In 2-valued logic, $f_\neg(0) = 1$ and $f_\neg(1) = 0$. To generalize this operation to the interval $[0,1]$, we must find the derivative of the discrete negation.

In accordance with the formula (3), this derivative is equal to $f_\neg(1) - f_\neg(0) = 0 - 1 = -1$. We want to use this same expression for the derivative of a continuous function $f_\neg(x)$.

Thus, we must find a function $f_\neg(x)$ for which $f_\neg(0) = 1$ and $f_\neg(0) = 1$ (since it is a generalization of a classical negation) and for which

$$\frac{df_\neg}{dx} = -1.$$

One can easily see that with the given boundary conditions, this differential equation has only one solution: $f_\neg(x) = 1 - x$.

Thus, our approach leads to the standard negation operation.

IV. Second Example: Selecting Fuzzy "And" (T-Norm)

Selecting a fuzzy "and"-operation $f_\&(a,b)$ is somewhat more difficult than selecting a fuzzy negation, because, instead of selecting a function of a single variable, we must select a function of *two* variables. Let us show how it can be done.

Our main idea is that once we fix the value of one of the variables, e.g., the first variable a, then $f_\&(a,b)$ becomes a function of only one variable: the variable b. Thus, if we can use the above approach to find, for each a from the interval $[0,1]$, the function $b \to f_\&(a,b)$, we will be able to uniquely reconstruct the values $f_\&(a,b)$ for every a and b – and hence, the desired function of two variables.

Let us therefore fix a and consider the function $g_a(b) = f_\&(a,b)$ of one variable. To reconstruct this auxiliary function, we must know its initial condition, i.e., the value $g_a(0) = f_\&(a,0)$, and its derivative

$$\frac{dg_a(b)}{db} = \frac{\partial f_\&(a,b)}{\partial b}.$$

In accordance with our methodology, as the value of this derivative, we take the discrete derivative

$$\frac{\mathcal{D}g_a(b)}{\mathcal{D}b} = g_a(1) - g_a(0) = f_\&(a,1) - f_\&(a,0). \tag{4}$$

Thus, to apply this idea, we must know, for each value a, the values $f_\&(a,1)$ and $f_\&(a,0)$. In other words, we need to know the values of another two auxiliary functions of one variable. These values can also be determined by the above procedure.

As a result, we arrive at the following procedure:

- First, we determine the values of the function $f_\&(a,0)$:
  - The initial condition comes from the requirement that this function should be an extension of the classical "and" operation. So, we have $f_\&(0,0) = 0$.
  - The discrete derivative of this function is equal to $f_\&(1,0) - f_\&(0,0) = 0 - 0 = 0$.
  - We extrapolate the function to arbitrary values of $a \in [0,1]$ by assuming that the same expression holds for the actual derivative

    $$\frac{\partial f_\&(a,0)}{\partial a}.$$

    Thus, we have a function whose initial value is 0, and whose derivative is 0, so the function is identically 0: $f_\&(a,0) = 0$ for all a.
- Second, we determine the values of the function $f_\&(a,1)$:
  - The initial condition comes from the requirement that this function should be an extension of the classical "and" operation. So, we have $f_\&(0,1) = 0$.

- The discrete derivative of this function is equal to $f_\&(1,1) - f_\&(0,1) = 1 - 0 = 1$.
- We extrapolate the function to arbitrary values of $a \in [0,1]$ by assuming that the same expression holds for the actual derivative
$$\frac{\partial f_\&(a,1)}{\partial a}.$$
Thus, we have a function whose initial value is 0, and whose derivative is 1, so the function is equal to $f_\&(a,1) = 0 + 1 \cdot a = a$ for all a.

- Finally, we determine the values of the function $f_\&(a,b)$ for arbitrary a and b. We do that by determining, for each $a \in [0,1]$, the function $g_a(b) = f_\&(a,b)$ of one variable b.
  - The initial condition for the function $g_a(b)$ comes from the already known fact that $f_\&(a,0) = 0$.
  - The discrete derivative of this function is equal to $f_\&(a,1) - f_\&(a,0) = a - 0 = a$.
  - We extrapolate the function to arbitrary values of $b \in [0,1]$ by assuming that the same expression holds for the actual derivative
$$\frac{\partial f_\&(a,b)}{\partial b}.$$
Thus, we have a function whose initial value is 0, and whose derivative is a, so the function is equal to $f_\&(a,b) = 0 + a \cdot b = a \cdot b$ for all a and b.

So, we get the "algebraic product", one of the t-norms originally proposed by L. Zadeh in his pioneer paper [8].

Comment. Instead of looking for a function $g_a(b) \stackrel{\text{def}}{=} f_\&(a,b)$, we could, alternatively, start by looking for a function $h_b(a) \stackrel{\text{def}}{=} f_\&(a,b)$. One can easily show that we would have gotten the same result.

V. Third Example: Selecting Fuzzy "Or" (T-Conorm)

A similar procedure can determine the fuzzy "or" operation (a t-conorm):

- First, we determine the values of the function $f_\vee(a,0)$.
  - The initial condition comes from the requirement that this function should be an extension of the classical "or" operation. So, we have $f_\vee(0,0) = 0$.
  - The discrete derivative of this function is equal to $f_\vee(1,0) - f_\vee(0,0) = 1 - 0 = 1$.
  - We extrapolate the function to arbitrary values of $a \in [0,1]$ by assuming that the same expression holds for the actual derivative
$$\frac{\partial f_\vee(a,0)}{\partial a}.$$
Thus, we have a function whose initial value is 0, and whose derivative is 1, so the function is equal to $f_\vee(a,0) = 0 + 1 \cdot a = a$ for all a.

- Second, we determine the values of the function $f_\vee(a,1)$.
  - The initial condition comes from the requirement that this function should be an extension of the classical "or" operation. So, we have $f_\vee(0,1) = 0$.
  - The discrete derivative of this function is equal to $f_\vee(1,1) - f_\vee(0,1) = 1 - 1 = 0$.
  - We extrapolate the function to arbitrary values of $a \in [0,1]$ by assuming that the same expression holds for the actual derivative
$$\frac{\partial f_\vee(a,1)}{\partial a}.$$
Thus, we have a function whose initial value is 1, and whose derivative is 0, so the function is equal to $f_\vee(a,1) = 1$ for all a.

- Finally, we determine the values of the function $f_\vee(a,b)$ for arbitrary a and b. We do that by determining, for each $a \in [0,1]$, the function $g_a(b) = f_\vee(a,b)$ of one variable b.
  - The initial condition for the function $g_a(b)$ comes from the already known fact that $f_\vee(a,0) = 0$.
  - The discrete derivative of this function is equal to $f_\vee(a,1) - f_\vee(a,0) = 1 - a$.
  - We extrapolate the function to arbitrary values of $b \in [0,1]$ by assuming that the same expression holds for the actual derivative
$$\frac{\partial f_\vee(a,b)}{\partial b}.$$
Thus, we have a function whose initial value is a, and whose derivative is $1-a$, so the function is equal to $f_\vee(a,b) = a + (1-a) \cdot b = a + b - a \cdot b$ for all a and b.

So, we get the "algebraic sum", one of the t-conorms originally proposed by L. Zadeh in his pioneer paper [8].

Comment. The fact that these operations surface is no accidental: An alternative justification of the use of this operation comes, e.g., from requiring that it should be, on average, the least sensitive to possible uncertainty in determining the exact values of $d(A)$ and $d(B)$; see [6]. Since a natural interpretation of "sensitivity" is via derivatives, there is a clear relationship between these results.

VI. General Result

Let us now formulate, in precise terms, a general procedure for extrapolating as function $f(x_1, \ldots, x_n)$ of n propotional variables $x_i \in \{0,1\}$ into a function of n fuzzy variables $x_i \in [0,1]$:

- For a function of a single variable ($n = 1$), the reconstruction is straightforward:
  - we know the value $f(0)$, and
  - we require that for every value x_1, we have
$$\frac{\partial f}{\partial x_1} = \frac{\mathcal{D} f}{\mathcal{D} x_1} = f(1) - f(0); \quad (5)$$
  - thus, we get $f(x_1) = f(0) + (f(1) - f(0)) \cdot x_1$, i.e., $f(x_1) = f(0) \cdot (1 - x_1) + f(1) \cdot x_1$.
- Let us now assume that we already know how to extrapolate functions of $n - 1$ variables. Then, we can determine the values of $f(x_1, \ldots, x_n)$ for all $x_1, \ldots, x_n$, if we find, for every $x_1, \ldots, x_{n-1}$, the function
$$g_{x_1,\ldots,x_{n-1}}(x_n) \stackrel{\text{def}}{=} f(x_1, \ldots, x_{n-1}, x_n) \quad (6)$$
as follows:
  - The initial value $g_{x_1,\ldots,x_{n-1}}(0) = f(x_1, \ldots, x_{n-1}, 0)$ of the auxiliary function g is a function of $n - 1$ variables $x_1, \ldots, x_{n-1}$, so, according to our assumption, we already know how to use extrapolation to find these values.
  - We require that for every value x_n, we have
$$\frac{\partial f}{\partial x_n} = \frac{\mathcal{D} g}{\mathcal{D} x_n} = f(x_1, \ldots, x_{n-1}, 1) - f(x_1, \ldots, x_{n-1}, 0).$$
Here, $f(x_1, \ldots, x_{n-1}, 1)$ is also a function of $n-1$ variables, so, according to our assumption, we already know how to use extrapolation to find these values.
  - Thus, we get
$$f(x_1, \ldots, x_{n-1}, x_n) = f(x_1, \ldots, x_{n-1}, 0) + f(x_1, \ldots, x_{n-1}, 1) - f(x_1, \ldots, x_{n-1}, 0) \cdot x_n,$$
i.e.,
$$f(x_1, \ldots, x_{n-1}, x_n) = f(x_1, \ldots, x_{n-1}, 0) \cdot (1 - x_n) + f(x_1, \ldots, x_{n-1}, 1) \cdot x_n.$$

The result of applying this procedure is easy to describe:

Proposition. *As a result of applying the above procedure, we get the following multi-linear function*
$$f(x_1, \ldots, x_n) = \sum_{\varepsilon_1 \in \{0,1\}} \cdots \sum_{\varepsilon_n \in \{0,1\}} f(\varepsilon_1, \ldots, \varepsilon_n) \cdot x_1^{(\varepsilon_1)} \cdot \ldots \cdot x_n^{(\varepsilon_n)}, \quad (7)$$
where $x^{(1)}$ means x, and $x^{(0)}$ means $1 - x$.

This result is easy to prove by induction over n. This results shows, in particular, that although the procedure depends on which of the variables we select first, the resulting function $f(x_1, \ldots, x_n)$ does not depend on the order in which the variables x_i are selected.

All three above results are a particular case of this general proposition. Indeed, for negation, for which $f(0) = 1$ and $f(1) = 0$, this proposition implies that
$$f(x_1) = f(0) \cdot x_1^{(0)} + f(1) \cdot x_1^{(1)} = f(0) \cdot (1 - x_1) + f(1) \cdot x_1 = 1 - x_1. \quad (8)$$
For an "or" operation, for which $f(0,0) = 0$ and $f(0,1) = f(1,0) = f(1,1) = 1$, we get
$$f(x_1, x_2) = f(0,0) \cdot (1 - x_1) \cdot (1 - x_2) + f(0,1) \cdot (1 - x_1) \cdot x_2 + f(1,0) \cdot x_1 \cdot (1 - x_2) + f(1,1) \cdot x_1 \cdot x_2 = (1 - x_1) \cdot x_2 + x_1 \cdot (1 - x_2) + x_1 \cdot x_2 = x_1 + x_2 - x_1 \cdot x_2. \quad (9)$$
We can apply this result to other classical logic operations as well. For *implication*, for which $f_{\to}(1,0) = 0$ and $f_{\to}(0,0) = f_{\to}(0,1) = f_{\to}(1,1) = 1$, we get $f_{\to}(a,b) = 1 - a + a \cdot b$; this expression can be represented in terms of other "algebraic" operations is we assume that $f_{\to}(a,b) = f_{\vee}(f_{\neg}(a), b)$ (i.e., that we have an S-implication).

For "*exclusive or*" $f_{\oplus}(a,b)$, we get $f_{\oplus}(a,b) = a + b - 2 \cdot a \cdot b$; for *equivalence* $f_{\equiv}(a,b)$, we get $f_{\equiv}(a,b) = 1 - a - b + 2 \cdot a \cdot b$. It is worth mentioning that, in contrast to the expression for implication, expressions for "exclusive or" and for equivalence cannot be expressed as compositions of the algebraic operations "and", "or", and "not" [4].

Acknowledgments

This work was supported in part by NASA grants NCC5-209 and NCC 2-1232, NSF grants CDA-9522207 and 9710940 Mexico/Conacyt, Air Force Office of Scientific Research grant F49620-00-1-0365, and by Grant No. W-00016 from the U.S.-Czech Science and Technology Joint Fund.

The authors are thankful to Witold Pedrycz for useful discussions, and to anonymous referees for their valuable comments.

References

[1] S. Dick and A. Kandel, "The linguistic gradient of a fuzzy logic rulebase", *Proc. FUZZ-IEEE'2000*, Vol. 2, pp. 911–916.

[2] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, 1995.

[3] V. Kreinovich and G. Mints (eds.), *Problems of reducing the exhaustive search*, American Math. Soc., Providence, RI, 1997.

[4] V. Kreinovich, H. T. Nguyen, and E. A. Walker, "Maximum entropy (MaxEnt) method in expert systems and intelligent control: new possibilities and limitations", In: K. M. Hanson and R. N. Silver (eds.), *Maximum Entropy and Bayesian Methods*, Kluwer, Dordrecht, 1996, pp. 93–100.

[5] H. T. Nguyen and V. Kreinovich, "How to Divide a Territory? A New Simple Differential Formalism for Optimization of Set Functions", *Int'l J. Intelligent Systems*, 1999, Vol. 14, No. 3, pp. 223–251.

[6] H. T. Nguyen, V. Kreinovich, and D. Tolbert, "A measure of average sensitivity for fuzzy logics", *Int'l J. Uncertainty, Fuzziness, and Knowledge-Based Systems*, 1994, Vol. 2, No. 4, pp. 361–375.

[7] H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, FL, 1999.

[8] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.