CREATING AND DECOMPOSING FUNCTIONS USING FUZZY FUNCTIONS

József Dániel Dombi and József Dombi
Institute of Informatics, University of Szeged, Árpád tér 2., Szeged, Hungary

Keywords: Fuzzy System, Approximation, Membership function, Sigmoid function, Conjunctive operator, Aggregation operator, Dombi operator, Pliant concept.

Abstract: In this paper we will present a new approach for composing and decomposing functions. This technology is based on the Pliant concept, which is a subclass of the fuzzy concept. We will create an effect by using the conjunction of the sigmoid function. After, we will make a proper transformation of an effect in order to define the neutral value. Then, by repeatedly applying this method, we can create effects. Aggregating the effects, we can compose the desired function. This tool is also capable of function decomposition as well, and can be used to solve a variety of real-time problems. Two advantages of our solution are that the parameters of our algorithm have a semantic meaning, and that it is also possible to use the values of the parameter in any kind of learning procedure.

1 INTRODUCTION

Functions have a very important role in science and technology and in our everyday lives. They can be represented in terms of their coordinates or by using some mathematical expression. Usually if the coordinates are given then it is important to know what kind of expression approximately describes it, because sometimes interpolation or extrapolation questions have to be addressed. In science and technology we can usually get samples to determine the relationship between the input and output values, which is called curve fitting, and usually we do not require an exact fit, but only an approximation. One way to approximate a function with coordinates is via an interpolation process. We can regard interpolation as a specific kind of curve fitting, where the function must go through the data points. It is also possible to use neural networks to find a function approximation. Curve fitting can be done by minimising the error function that measures the misfit between the function for any given value of the parameters and the data points. One simple and widely used error function is the sum of the squares of the errors, so in effect we have to minimise the 'energy function'.
However, every type of curve fitting method has its drawbacks and this one is no different. The main problem is how to choose the order n of the polynomial and this will turn out to be a problem of model comparison or model selection. These methods are not accurate enough. The parameters that we get after optimisation give us no direct information about the behaviour of the function. It would be useful if we could modify a certain part of the function by varying the parameter. And it would be good if we could characterise a function by its behaviour. Using classical function construction procedures, it is not so easy to find a parametrical mathematical expression which corresponds to the natural language description of the function, but it would be useful in fields like economics and marketing.

In this article we will present a solution that solves some of these problems. We will introduce positive and negative effects, whose mathematical description can be realised by using continuous-valued logic. Here we will use a special one called the Pliant concept, which uses the Dombi operator. After an aggregative procedure we get the derived function. Instead of the membership function we shall use soft inequalities and soft intervals which are called distending functions. All of the parameters introduced have a definite meaning. It can be proved that certain function classes may be uniformly approximated.In the following section we will concentrate on a certain structure called the Pliant concept for the construction of the necessary operators.

Dániel Dombi J. and Dombi J. (2010).
CREATING AND DECOMPOSING FUNCTIONS USING FUZZY FUNCTIONS.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 400-403
DOI: 10.5220/0002901204000403

2 BASIC OPERATOR OF THE PLIANT CONCEPT

Pliant conjunctive and disjunctive operators belong to the strict t-norm and t-conorm classes.

2.1 Conjunctive and Disjunctive Operators

Definition 1. *$c(x,y)$ and $d(x,y)$ are strict monotone logical operators if the following hold true:*

1. *Continuity*
2. *Monotonously increasing*
3. *Compatibility with logic*
4. *Associativity*
5. *Archimedian, which means*

$$c(x,x) < x < d(x,x), \text{where } x\varepsilon(0,1)$$

Definition 2. *(Aczél) The operators with all the properties except property 3 can be written in the following way:*

$$c(x,y) = f_c^{-1}\left(f_c(x) + f_c(y)\right)$$
$$d(x,y) = f_d^{-1}\left(f_d(x) + f_d(y)\right),$$

where $f_c(x)$ and $f_d(x)$ are the generator functions of the operator, which is defined up to a multiplicative constant.

In an article by Dombi (Dombi, 1982b) there are sets of logical operators that have the above properties (1).

Definition 3. *An operator belongs to the Pliant system if*

$$f_c(x) f_d(x) = 1 \qquad (1)$$

A special case of the Pliant system is the Dombi operator.

2.2 Distending Function

In pliant logic we use a soft inequality and we call it the distending function.

Definition 4. *The distending function:*

$$\delta_a^{\lambda}(x) = f^{-1}\left(e^{-\lambda(x-a)}\right) \qquad \lambda\varepsilon R, a\varepsilon R \qquad (2)$$

Here f is the generator function of the logical connectives, λ is responsible for the sharpness and a is the threshold value.

The semantic meaning of $\delta_a^{(\lambda)}$ is

$$truth(a <_{\lambda} x) = \delta_a^{(\lambda)}(x)$$

The distending function in the Dombi operator case is the sigmoid (logistic) function.

2.3 Modeling Interval

In fuzzy logic the membership function is not an open interval, but in most cases it is a soft interval. We have to give a mathematical description of

$$truth(a <_{\lambda_1} x <_{\lambda_2} b)$$

Definition 5. *The distending interval falls within in the Dombi operator case:*

$$\sigma_{a,b}^{\lambda_1,\lambda_2}(x) = \frac{1}{1 + \frac{1-\nu_0}{\nu_0}\frac{1}{A}\left(A_1 e^{-\lambda_1(x-a)} + A_2 e^{-\lambda_2(b-x)}\right)},$$

where

$$A = 1 - e^{-(\lambda_1+\lambda_2)(b-a)}$$
$$A_1 = 1 - e^{-\lambda_2(b-a)}$$
$$A_2 = 1 - e^{-\lambda_1(b-a)}$$

The following properties hold true for the distending interval.

$$\delta_{a,b}(x) = \lim_{\lambda_1\to\infty,\lambda_2\to\infty} \delta_{a,b}^{\lambda_1,\lambda_2}(x) = \begin{cases} 0 & \text{if} \quad x < a \\ 0.5 & \text{if} \quad x = a \\ 1 & \text{if} \quad a < x < b \\ 0.5 & \text{if} \quad x = b \\ 0 & \text{if} \quad b < x \end{cases}$$

2.4 Aggregation Operator

The aggregation concept was first introduced in (Dombi, 1982a), which is also called the uninorm. Several articles discuss uninorms (J. Fodor and Rybalov, 1997), (Li and Shi, 2000), (D. Dubois, 2000), (Yager., 2001), but here we will just focus on the aggregation concept. Using Aczél's theorem (Aczél, 1966) about an associative equation we get:

$$a(x,y) = f_a^{-1}\left(f_a(x) + f_a(y)\right) \qquad (3)$$

Applying the Pliant concept and using the Dombi's generator function with multiple variables, we have:

$$a(x_1, x_2 \ldots x_n) = \frac{\prod_{i=1}^{n} x_i}{\prod_{i=1}^{n} x_i + \prod_{i=1}^{n}(1-x_i)} \qquad (4)$$

This operator can be found in (Dombi, 1982a). Nowadays it is called the 3π operator for obvious reasons. The main properties of the aggregation operator are:

1. $a(x, n(x)) = \nu_*$
2. $a(x, \nu_*) = x$

where ν_* is the fixpoint of the negation; that is, $n(\nu_*) = \nu_*$.

The ν_* value plays an important role in the definition of the aggregation operator (Dombi, 2008). If $x, y < \nu_*$ then the aggregation can be viewed as a conjunction and if $\nu_* < x, y$ then the aggregation can be viewed as a disjunction. If ν_* is close to 0 then the operation is disjunctive, and if ν_* is close to 1 then the operation is conjunctive.

3 CONSTRUCTION OF FUNCTION BY POSITIVE AND NEGATIVE EFFECTS

Because the aggregation operator has a neutral value we have to transform the interval to $[0, \nu]$ or $[\nu, 1]$. We will define positive and negative effects using the distending interval. That is,

$$P_{a_1,a_2}^{\lambda_1,\lambda_2}(x) = \frac{1}{2}\left(1 + \gamma\sigma_{a,b}^{\lambda_1,\lambda_2}(x)\right) \tag{5}$$

$$N_{a_1,a_2}^{\lambda_1,\lambda_2}(x) = \frac{1}{2}\left(1 - \gamma\sigma_{a,b}^{\lambda_1,\lambda_2}(x)\right) \tag{6}$$

where the scaling factor $\gamma \in [0, 1]$ controls the intensity of the effect. Equations (5) and (6) have a common form if $\gamma \in [-1, 1]$, namely

$$E_{a_1,a_2}^{\lambda_1,\lambda_2}(\gamma, x) = \frac{1}{2}\left(1 + \gamma\sigma_{a,b}^{\lambda_1,\lambda_2}(x)\right) \tag{7}$$

Here if $\gamma > 0$ then we have a positive effect and if $\gamma < 0$ we have a negative effect.

Using the aggregation operator we get a function that models certain positive and negative effects.

The goodness of this construction will now be described. If the functions are integrable function in the Riemannian sense, then there exist upper or lower approximations of rectangles. Because $\delta_{a,b}(x)$ is a rectangle and the aggregation of the rectangles are rectangles, we can define an interval where $0 < a_1 < a_2 \ldots < a_n < 1$. The discretisation of an interval rectangle approximation will be sufficiently good if the a_i, a_j intervals are small enough. So our method can be applied on any function that is integrable in the Riemannian sense.

4 FUNCTION DECOMPOSITION

In the previous section we saw that we can construct a desired function using the aggregation operator and functions that model the effects. When applying this method, the reverse case may sometimes be helpful too. We will show that we can do this by using an optimisation method. We can find a wide variety of optimisation techniques. If the initial values are properly chosen it is not hard to get the global minimum by using a local search algorithm. Here we will apply the well-known BFGS method (Ruszczynski, 2001). Because we can define initial points that are not far away from the optimum, the BFGS method can find the optimal solution in a few iterations.

In our experiment we will use a function with a dense sampling procedure. In each example we will use 100 equidistant coordinates on the given interval.

Let us define a function $F : \mathbb{R} \to [0, 1]$ that we would like to approximate. Our task is then to decompose it into positive and negative effects. This can be done via our approximation method or any interpolation procedure. In order to get a good result, we shall first smooth the function $F(x)$. The algorithm has the following steps:

1. Find the the local minimum and maximum value of the function $F(x)$:

   $$F(c_i) = A_i, \text{ where } F(x) < A_i, \text{ if } x\varepsilon(c_i - \varepsilon, c_i + \varepsilon)$$

   $$F(c_j) = A_j, \text{ where } F(x) > A_j, \text{ if } x\varepsilon(c_i - \varepsilon, c_i + \varepsilon)$$

2. Define the $[a_i, b_i]$ intervals

   $$a_n = \frac{c_{n-1} + c_n}{2}, b_n = c_n + \frac{c_{n-1} + c_n}{2}$$

   where

   $$c_1 < c_2 < c_3 < \ldots < c_k$$

3. Define the initial value of λ_{i_1} and λ_{i_2}

   $$\lambda_{i_1} = 2\frac{f(c_i) - f(a_i)}{c_i - a_i} \qquad \lambda_{i_2} = 2\frac{f(b_i) - f(c_i)}{b_i - a_i}$$

   Here there is a multiplicative constant of 2, because we have to transform the distending interval up and down.

4. Build the initial effects of the function:

   $$E_i(x) = E_{a_i,b_i}^{\lambda_{i_1},\lambda_{i_2}}(\gamma, x)$$

   Using these effects we can create the approximated function with the help of the aggregation operator:

   $$G_{\mathbf{a},\mathbf{b}}^{\underline{\lambda}_1,\underline{\lambda}_2}(\underline{\gamma}, x) = \frac{1}{1 + \prod_{i=1}^{n} \frac{1-E_i(x)}{E_i(x)}}$$

5. Find the optimal solution of the $a_i, b_i, \underline{\gamma}, \lambda_{i_1}, \lambda_{i_2}$ values with the suggested initial values.

$$\min_{\mathbf{a},\mathbf{b},\underline{\gamma},\underline{\lambda}_1,\underline{\lambda}_2} \sum \left(G_{\mathbf{a},\mathbf{b}}^{\underline{\lambda}_1,\underline{\lambda}_2}(\underline{\gamma}, x_i) - F(x_i) \right)^2$$

6. If the difference between the original function and the approximated function is too large then extract the approximated function from the original function and repeat the procedure in order to define additional effects.

It is not easy to minimise this problem, because a minimum may not be the global minimum. However, because $G_{\mathbf{a},\mathbf{b}}^{\underline{\lambda}_1,\underline{\lambda}_2}(\underline{\gamma}, x)$ is a continuous function of its parameters and the initial values are well-chosen, we can get good results.

The results of this are shown pictorially in figures 1 and 2. Our solution has an error of less than $|0.04|$. We performed some additional tests, but the results turned out to be the same as those presented below.

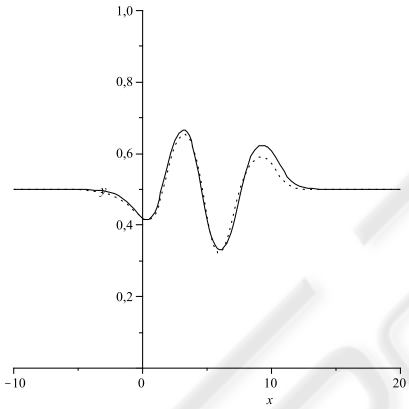

Figure 1: The function and its approximation.

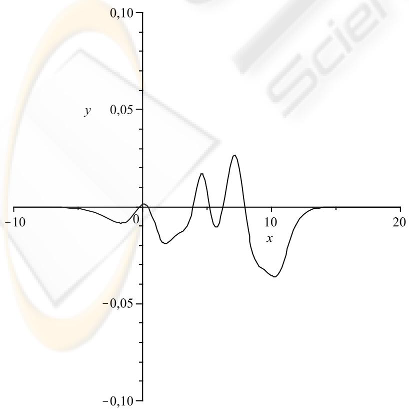

Figure 2: Plot of the difference between the function and its approximation.

5 CONCLUSIONS

In this article we developed a new type of non-linear regression method which is based on positive and negative effects and provides a natural description of the function. Our algorithm uses the BFGS method for getting accurate effects. We showed that this procedure is effective if all the data points are given. We found that this method is fast (only a few iteration steps are required for the optimisation procedure) and it is easy to use. Moreover, it is not necessary to modify the whole of the function.

REFERENCES

Aczél, J. (1966). *Lectures on Functional Equations and Applications*. Academic Press.

D. Dubois, H. P. (2000). Fundamentals of fuzzy sets (the hanbooks of fuzzy sets series). *Kluwer*.

Dombi, J. (1982a). Basic concept for a theory of evaluation: the aggregation operator. *Europian Journal of Operations Research*, 10.

Dombi, J. (1982b). A general class of fuzzy operators, the de morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems*, 8.

Dombi, J. (2008). Towards a general class of operators for fuzzy systems. *IEEE Transaction on Fuzzy Systems*, 16.

J. Fodor, R. Y. and Rybalov, A. (1997). Structure of uninorms. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 5.

Li, Y. M. and Shi, Z. K. (2000). Weak uninorm aggregation operators. *Information Sciences*, 124.

Ruszczynski, A. (2001). Nonlinear optimization. *Princeton University Press*.

Yager., R. (2001). Uninorms in fuzzy systems modeling. *Fuzzy Sets and Systems*, 122.