# Function Construction with Fuzzy Operators

**József Dombi**

Árpád tér 2, H-6720 Szeged, Hungary, email: dombi@inf.u-szeged.hu

**József D. Dombi**

Árpád tér 2, H-6720 Szeged, Hungary, email: dombijd@inf.u-szeged.hu

*Abstract: In this paper we present a new approach to compose and decompose functions. This technology is based on pliant concept. We use Sigmoid and proper transformations of Conjunction to create an effect. We aggregate this effect to compose the function. This tool is also capable for function decomposition.*

*Keywords: Pliant concept, Sigmoid, Conjunction, Aggregation, Dombi operator*

## 1 Introduction

Numerical analysis is the area of mathematics and computer science that creates, analyzes and implements algorithms for solving numerically the problems of continuous mathematics. One of the main subfields of numerical analysis is interpolation. Interpolation is a method of constucting new data points from a discrete set of known data points. In science one often has a number of data points, as obtained by sampling, and tries to construct a function, which closely fits those data points (also called training points), which is called curve fitting. Interpolation is a specific case of curve fitting, in which the function must go exactly through data points. There are lots of interpolations for example: linear, polynomial, spline and trignometric. Usually we fit data using a polynominal function of the form

$$y(x,w) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j \qquad (1)$$

where *M* is the order of the polynomial, and $x^j$ denotes *x* raised to the power of *j*. The polynomial coeddicients $w_0,\ldots w_M$ are collectively denoted by the vector w. Note that, although the polynomial function y(x,w) is a nonlinear function of x, it is a linear function of the coefficients w. The values of the coefficients will be determined by fitting the polynomial to the training points. This can be done by minimizing the error function that measures the misfit between the function for any given value of w and the data points. One simple choice of error function which is widely used is given by the sum of the squares of the errors between the predictions $y(x_n,w)$ for each data point $x_n$ and the corresponding target values $t_n$, so that we minimize

$$E(w) = \frac{1}{2}\sum (y(x_n, w) - t_n)^2 \tag{2}$$

We simply note that it is a nonnegative quantity that would be zero if and only if the function y(x,w) were to pass exactly through each training data point. The geometrical interpretation of the sum-of-squares error function is illustrated in Figure 1.

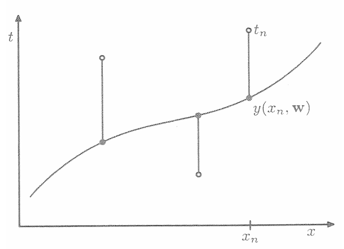


Figure 1
The error function

We can solve the curve fitting problem by choosing the value w for which E(w) is as small as possible. It is also a problem of choosing the order *M* of the polynomial and as we shall see this will turn out to be an example of an important concept called model comparison or model selection. In Figure 2, we show four examples of the results of fitting polynomials having orders M = 0, 1, 3, and 9 to the data set, which is a sinus function with noise.

The problems of the interpolation is that we do not understand the meaning of the parameters. In our approach we drop the traditional concept. We define effect as a basic unit, which has a meaning. In Section 2 we define basic units of the effect and show how to create effect. In Section 3 we define how to compose and decompose functions. At the end we present some recent results.

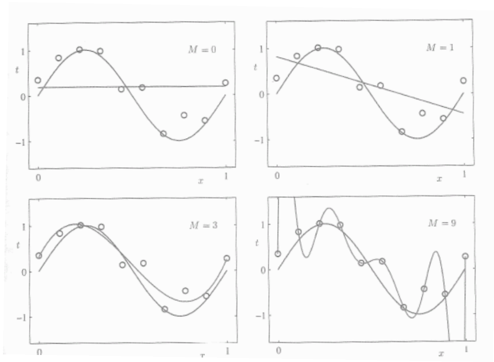

Figure 2
Plots of polynomials having various orders *M*

# 2 Operations

In this section we present how to compose effect.

## 2.1 Sigmoid

A sigmoid function is a mathematical function that produces a sigmoid curve. We defined it by the following formula:

$$\sigma_a^{\lambda}(x) = \frac{1}{1+e^{-\lambda(x-a)}} \tag{3}$$

where λ is the sharpness of the function and *a* is the offset. It is easy to see that

- $\sigma_a^{\lambda}(a) = a$,
- $\sigma_a^{-\lambda}(x) = 1 - \sigma_a^{\lambda}(x)$.

## 2.2 Conjunction and Dombi Operator

To construct the effect we use fuzzy operators and sigmoid function. It is important to consider that the fuzzy operator and Dombi operator fits well. The associative function equation is the following:

$$c(x, y) = f^{-1}\left(f(x) + f(y)\right) \tag{4}$$

This function $c:[0,1]\times[0,1]\rightarrow[0,1]$, which satisfied the following properties:

- Commutativity: $c(a,b)=c(b,a)$
- Monotonicity: $c(a,b)\leq c(c,d)$, if $a\leq c$ and $b\leq d$
- Associativity: $c(a,c(b,c))=c(c(a,b),c)$
- 1 is an identity element: $c(a,1)=a$

The function of Dombi operator:

$$f(x)=\frac{1-x}{x},\ f^{-1}(x)=\frac{1}{1+x} \tag{5}$$

If we use Dombi operator in the conjunction we get the following equation:

$$c(x,y)=\frac{1}{1+\frac{1-x}{x}+\frac{1-y}{y}} \tag{6}$$

We can extend this formula with power and weight:

$$c_\lambda(x,y,u,v)=\frac{1}{1+\left(u\left(\frac{1-x}{x}\right)^\lambda+v\left(\frac{1-y}{y}\right)^\lambda\right)^{\frac{1}{\lambda}}} \tag{7}$$

In this case we lose the associativity property.

## 2.4 Aggregation

Aggregation is not a logical operator. The form of an aggregation operator is:

$$a(x_1,..x_n)=\frac{1}{1+\frac{1-\upsilon_0}{\upsilon_0}\frac{\upsilon}{1-\upsilon}\prod_{i=0}^{n}\frac{1-x_i}{x_i}}, \tag{8}$$

where ν is the neutral value and $\nu_0$ is the eigenvalue of the corresponding negation. The aggregation operator satisfies the following properties:

- Defined on [0,1] interval: $\bigcup[0,1]^n\rightarrow[0,1]$,
- Identity when unary: $a(x)=x$,

- Boundary Conditions: $a(0,...,0) = 0$ and $a(1,...,1) = 1$,
- Non decreasing: $a(x_1,...,x_n) \le a(y_1,...,y_n)$ if $(x_1,...,x_n) \le (y_1,...,y_n)$.

# 3 Function Composition and Decomposition

We create a function, which is defined on [0,1] interval. To achieve this we will use Sigmoid function, Dombi operator, Conjunction operator and Aggregation operator. First we need natural effect, so we use two Sigmoid (3) function, and the conjunction operator (5):

$$eff_{a,b}^{\lambda_1,\lambda_2}(x) = \frac{1}{1 + e^{-\lambda_1(x-a)} + e^{-\lambda_2(x-b)}} \tag{9}$$

In Figure 3 you can see this effect. We define positive effect, which means that this will increase the function value and we define negative effect, which is the opposite of the positive effect. We define the positve effect on [1/2,1] and negative effect on [0,1/2]. The neutral value is 1/2, which means that there is no effect. To achieve this we need the proper transformation of (9):

$$P(x) = \frac{1}{2}(1 + eff_{a,b}^{\lambda_1,\lambda_2}(x)) \tag{10}$$

$$N(x) = \frac{1}{2}(1 - eff_{a,b}^{\lambda_1,\lambda_2}(x)) \tag{11}$$

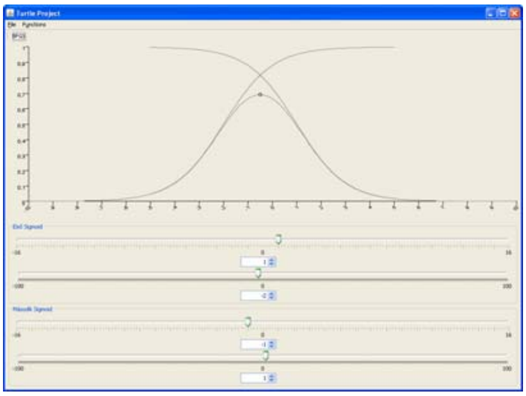

Figure 3
Conjunction of two Sigmoid function

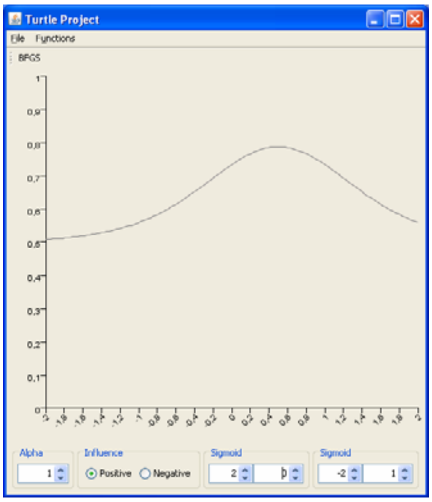


Figure 4
Positive Effect

In Figure 4 you can see that this function has four parameters. These parameters have semantic meanings: *a* is the starting point of the effect, *b* is the end point of the effect and lambdas mean the sharpness of the effect.

The last step of function creations is to aggregate these effects as you can see in Figure 5.

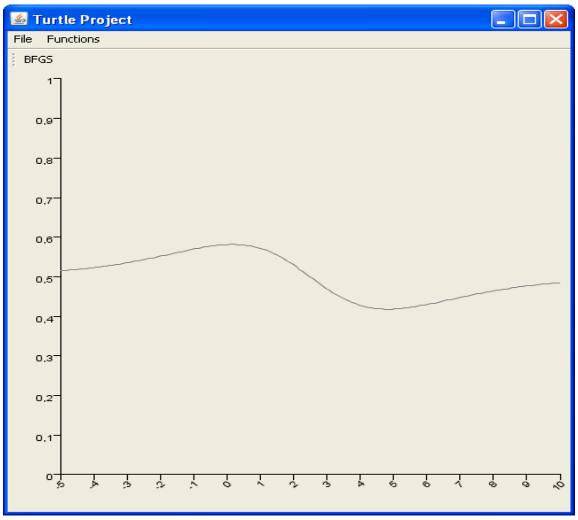


Figure 5
Aggregation of effects

In function decomposition we know real values. Our task is to divide real values into effects. It is an optimization problem. The critical point of the optimization is the initial values of the effects. If we define good initial points we have a high chance to find the global optimum.

In our case it is not impossible. We can define the number of the effect and each effect we can determine the following properties:

- Start/end point of the effect
- Sharpness of the effect.

# 4 Result

We create a Java program to create and analyze our approach. In our system we can create any arbitrary funcion as you can see in Figure 6.

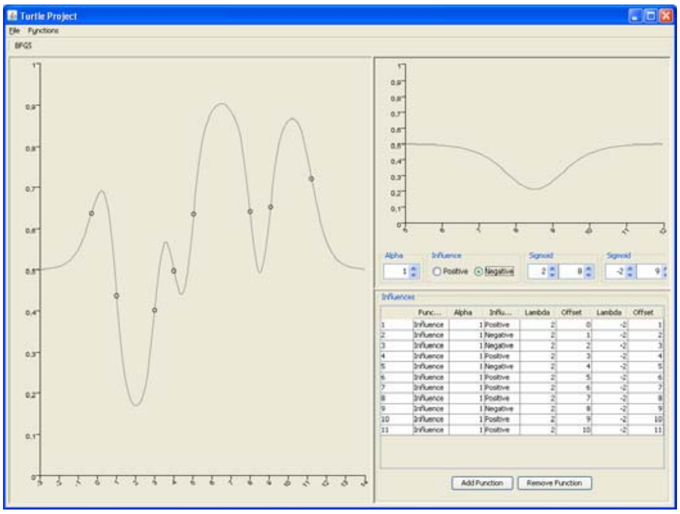


Figure 6
Our Java system

The global optimization is under development in our Java system, but we tried in Matlab and the solution calculates very fast the parameters and the error of the result is acceptable.

**Conclusions**

In this paper we develop a new tool, that is useful for interpolation. Our approach has good properties. Our model is based on fuzzy logic. We create effects and aggregate them. Our method is also capable to decompose a function. We are working on an global optimization algorithm that is working in our System.

**References**

[1] Dombi J.: A General Class of Fuzzy Operators, the De Morgan Class of Fuzzy Operators and Fuzziness Measures Induced by Fuzzy Operators, Fuzzy Sets and Systems Vol. 8, 1982, pp. 149-163

[2] Dombi J.: Basic Concept for a Theory of Evaluation: the Aggregation Operator, Europian Journal of Operations Research Vol. 10, 1982, pp. 282-293

[3] Detyniecki M..: Fundamentals on Aggregation Operators, asturias, 2001