

INTERPOLATION OF FUZZY DATA

E. Khodaparast

Chabahar maritime university, the department of basic science

Published online: 08 August 2017

ABSTRACT

Considering the many applications of mathematical functions in different ways, it is essential to have a defining function. In this study, we used Fuzzy Lagrangian interpolation and natural fuzzy spline polynomials to interpolate the fuzzy data.

In the current world and in the field of science and technology, interpolation issues are also of a fuzzy type, it has many scientific applications in developmental work, medical issues, imaging, engineering software and graphics. Therefore, in this article we intend to investigate Interpolation of fuzzy data in order to apply fuzzy interpolation, by optimizing fuzzy polynomials, also achieve the best approximation of a fuzzy function.

Keyword: Fuzzy polynomial, Data Fuzzy, Lagrange Fuzzy

1. Interpolation of fuzzy data

Suppose $n + 1$ point's $x_0 < x_1 < \dots < x_n$ of \mathbb{R} and fuzzy numbers $(\mu_i)_{i=0}^n$ belongs to S , a function $T: \mathbb{R} \rightarrow S$ with the following features is built:

- 1) For each $i = 0, 1, \dots, n$, and $T(x_i) = \mu_i$
- 2) The function T is continuous.
- 3) If for each of $i = 0, 1, \dots, n$ we have $\mu_i = 1_{y_i}$, where $y_i \in \mathbb{R}$, and if t is a unique multiplicity of at most n , from all points $(x_i, y_i)_{i=0}^n$ then we have for each $x \in \mathbb{R}$: $T(x) = 1_{t(x)} 0$

Author Correspondence, e-mail: e.khodaparast@cmu.ac.ir

doi: <http://dx.doi.org/10.4314/jfas.v9i2s.806>



To solve this problem Leuven [R.Lowen, 1990] polynomial interpolation of Lagrange Fuzzy raised and demonstrates that h is continuous with respect to m.

Kaleva has equated the Lagrange fuzzy polynomial of Lowen [Kaleva, 1994] as following:

If $P(x)^\alpha$ is the set of $-\alpha$, the alignment of P (x), then:

In which $P_{d_0 d_1 \dots d_n}$ is a polynomial Lagrange interpolation for the interpolation of the data $(x_i, d_i), i = 0, 1, \dots, n$

2- Investigation of Fuzzy Lagrangian Interpolation Polynomial Calculation method

Suppose that Lagrange's Fuzzy Interpolation Algorithm (Kaleva):

$$P_{d_0 d_1 \dots d_n}(x) = \sum_{i=0}^n d_i L_i(x)$$

And in fuzzy mode, if we want to interpolate the points $(x_i \in \mathbb{R}, u_i \in S) \cdot i = 0, 1, \dots, n$:

$$P(x)^\alpha = \sum_{i=0}^n L_i(x) u_i^\alpha,$$

Since the above equation is for each α , so:

$$P(x) = \sum_{i=0}^n L_i(x) u_i .$$

The function f is Continuous and for each $x \in (x_i, x_{i+1})$, and each $\alpha \in [0, 1]$

Always

$$\text{len } P(x)^\alpha \geq \min\{\text{len } P(x_i)^\alpha, \text{len } P(x_{i+1})^\alpha\} ,$$

Here, *len* represents the length of the interval, [Kaleva, 1994].

If $u_i = (m_i, l_i, r_i)$, then $P(x) = (m(x), l(x), r(x))$ can be obtained from the following equation s:

$$r(x) = \sum_{L_i(x) \geq 0} L_i(x)r_i - \sum_{L_i(x) < 0} L_i(x)l_i,$$

Here $P(x)$ and u_i are triangular fuzzy numbers.

Since the polynomial interpolation between points can approximate the information in a way similar to the fuzzy polynomial interpolation is also true.

Example: Suppose $u_i = (m_i, l_i, r_i)$ is a triangular fuzzy number and the backup of the fuzzy number u_i is the closed interval $[m_i - l_i, m_i + r_i]$

Consider data from the following table:

The data in the table above is information that civil engineers and geologists have to design a 30-meter bridge structure with regard to geographic information as well as soil resistance and technical issues with the vector coordinates at first, ten meters first, ten meters second and bridge ending they are in their possession.

They want to approximate the soil resistance at points of ten and two meters and determine the best point to determine the location of the bridge's foundations.

Using Lagrange's fuzzy interpolation polynomials, we will be able to interpolate the function and approximate the information between the points, for example:

$$P(2 / 6) = (1 / 9, 2 / 45, 1 / 37)$$

3. Fuzzy spline node to node

Family of splines of the 1-degree on the nodes of, $i = 0, 1, \dots, n$ we show with S_l .

In other words $S \in S_l$ if

- 1) $S \in C^{l-2}[x_0, x_n]$
- 2) The function S for each $i = 0, 1, \dots, n$ on the interval $[x_i, x_{i+1}]$ is a polynomial of degree l .

Similar to state of polynomial interpolation, we can also define the fuzzy interpolation spline of degree l as follows:

If $f_S(x)^\alpha$ is the set of α and the alignment of $f_S(x)$, then:

Where the $S_{d_0 d_1 \dots d_n}$ is a spline function of degree l that interpolates the points of $(x_i, d_i)_{i=0,1,\dots,n}$
 $S_i \in S_l$ interpolate The points $j = 0, 1, \dots, n$, (x_j, f_j) where in:

$$f_j = \delta_{ij} = \begin{cases} 1 & j = i, \\ 0 & j \neq i. \end{cases}$$

In this case:

We have the similar to Lagrangian mode for the fuzzy state:

Since is set for each α so:

The fuzzy spline of the node to the node in the special case for $l = 2$ is linear and for $c = 4$ is cubic.

To determine the spline, we need two additional conditions to specify it individually. In the spin of the node to node, we provide these two conditions with the derivative of the third function of $S_{d_0 d_1 \dots d_n}$ at points x_1 and x_{n-1} in node conditions, we do not consider points x_0 and x_n as nodes. In other words, we consider the fragment of the first and second $S \in S_l$ polynomials $[x_0, x_1]$ and $[x_{n-1}, x_n]$ identically, and in the same way as the pieces of $S \in S_l$ polynomials, one to the last and last That is, we consider the same $[x_{n-2}, x_{n-1}]$ and $[x_{n-1}, x_n]$.

If the function S_i is cubic interpolation node to node, then:

- 1) S_i Function on each sub-interval $[x_j, x_{j+1}]$ is not united zero.
- 2) S_i Function on each $[x_j, x_{j+1}]$ has the same sign.
- 3) S_i Function on each point of x_j change sign for each $j \neq i$.

If f_s is an interpolate fuzzy cubic spline with the condition of the node to node, then for each $x \in (x_i, x_{i+1})$ and each $\alpha \in [0,1]$

If $u_i = (m_i, l_i, r_i)$, then $f_s(x) = (m(x), l(x), r(x))$ can be obtained from the following equation s :

$$l(x) = \sum_{S_i(x) \geq 0} S_i(x)l_i - \sum_{S_i(x) < 0} S_i(x)r_i,$$

Where in, $f_s(x)$ and u_i are triangular fuzzy numbers.

3. Natural fuzzy spline

The S function: $S: [x_0, x_n] \rightarrow R$ is a natural spline of the odd degree of the $l = 2m - 1$ and $m \geq 2$, provided that it has the following characteristics (Abbasbandy, Babolian, 1998)

- 1) $S \in C^{l-1}[x_0, x_n]$
- 2) The function $S(x)$ is a polynomial of degree l on $[x_i, x_{i+1}]$ for each $i = 0, 1, \dots, n - 1$
- 3) $S^v(x_0) = S^v(x_n) = 0$; $v = m, \dots, 2m - 2$

This family of splines with $S_l(x_0, x_n)$ is displayed. $S_i \in S_l(x_0, x_n)$ Interpolates points $j =$

$0, 1, \dots, n$ (x_j, f_j) and $S_{y_0 y_1 \dots y_n}(x) = \sum_{i=0}^n S_i(x)y_i$

In which $S_{y_0 y_1 \dots y_n}(x) \in S_l(x_0, x_n)$ interpolates the point's (x_i, x_n) $i = 0, 1, \dots, n$

So, in fuzzy mode, if we want to interpolate the node-to-node similar to $(x_i \in R, u_i \in S)$, $i = 0, 1, \dots, n$ with a normal spline, we will have:

Since is set for each α , so:

Where in $F(x)$ is the natural fuzzy spline function of the odd degree.

If $u_i = (m_i, l_i, r_i)$ then we can get the $F(x) = (m(x), l(x), r(x))$ from following equation s.

$$r(x) = \sum_{S_i(x) \geq 0} S_i(x)r_i - \sum_{S_i(x) < 0} S_i(x)l_i,$$

In which $F(x)$ and u_i are triangular fuzzy numbers.

Consider the tabular data of the example above, if using natural cubic spline ($l = 3$) approximate the function at around points, we have:

$$F(2 / 6) = (2 / 37, 2 / 05, 1 / 47)$$

4. The best approximation of a fuzzy function

"Assume that $n + 1$ is the non-fuzzy point x_1, x_2, \dots, x_n in R , and we have a fuzzy value for each of these points. Is it possible to construct a fuzzy function that contains the fuzzy numbers that are true in the given data, and in the conditions smoothity is also true? [2]"

Therefore, we are looking for a maximal polynomial of degree $n \cdot \tilde{P}_n(x)$. Here we have m distinct points and for each point we have a fuzzy value and we try to find a maximally n fuzzy polynomial of degree n , which is $n \leq m-1$. Of course, n is not necessarily equal to $m-1$, and in some cases m is very large; for such issues a general approximation polynomial is presented in [Abbasbandy, 2006]. In this chapter, we obtain the best polynomial approximation of a fuzzy function by linear programming.

A fuzzy polynomial of maximally degree n is a function $\tilde{P}: R \rightarrow TF(R)$, in which:

In which $P_m(x) \in \prod_n$ and \prod_n are the families of all true real polynomials of maximum degree n . Also, $\bar{P}(x)$ and $\underline{P}(x)$ are positive integer polynomials of maximum degree n . A set of all polynomials with a maximum of n fuzzy nets $P(x)$ is represented by $\widetilde{\prod}_n$.

Let the set of m points belong to R with $X = \{x_1, x_2, \dots, x_n\}$, whose fuzzy value is a function $\tilde{f}: R \rightarrow TF(R)$. For each of $i = 0, 1, 2 \dots m$ These points are:

A polynomial $\widetilde{P}^*(x)$ of maximum n degree is the best approximation of the function f on the set of $X = \{x_1, x_2, \dots, x_n\}$, if we have:

Now assume that $\tilde{\beta} = (\beta_m, \beta_l, \beta_r)$ is a triangular fuzzy number in which $\beta_m \geq 0$ we set the maximum fuzzy approximation error to $\tilde{\beta}$:

Where in:

The above statement can also be written:

We now model them as limitations of linear programming issues.

For each $i = 0, 1, 2, \dots, m$ we have:

$$\beta_r \geq \bar{P}(x_i) + f_{i_l}$$

So we have three linear programming problems that we need to should address them:

β_m

$$\left\{ \begin{array}{l} \min \beta_r \\ s. t: \beta_r - \sum_{j=1}^n \bar{a}_j x_i^j \geq f_{i_l} \quad i = 0, 1, 2, \dots, m \quad (3) \\ \beta_r \geq 0 \end{array} \right.$$

By linear programming (1), (2) and (3), $P_m(x)$, $\underline{P}(x)$ and $\bar{P}(x)$ are obtained, respectively, in the form of definite and true approximations of fuzzy information (x_i, f_{i_m}) , (x_i, f_{i_r}) and (x_i, f_{i_l}) .

Therefore, the best approximation \tilde{f} at the point x is $\tilde{P}(x) = (P_m(x), \underline{P}(x), \bar{P}(x))$ and the error of this approximation on X ; $\tilde{\beta} = (\beta_m, \beta_l, \beta_r)$.

But in some cases, fuzzy value may be a negative function. Therefore, in order to always have a non-negative fuzzy value, we present the best approximation as follows:

The existence and uniqueness of the best approximation of a fuzzy function

Suppose $m \geq n + 1$ to get the answer to the linear programming problems (2) and (3), we need to set $\beta_l = \beta_r = 0$.

We show that in the answer to the problem (2), $\beta_l = 0$ (with the same argument for (3), $\beta_r = 0$).

For this purpose, we find the point (2) so that for this point $\beta_l = 0$ (as we have noted, $\beta_l \geq 0$ was one of the conditions of problem (2)), and finally we reach the desired result.

For this purpose, we obtain the polynomial $\underline{P}(x) = \sum_{j=0}^n \underline{a}_j x_i^j$ such that every $i = 0, 1, 2, \dots, m$, we have:

Now we obtain the quantity of polynomial interpolation at all $n + 1$ points $\{(x_i, f_{i_r}) | i = 0, 1, 2, \dots, m\}$.

Suppose:

If $h > 0$, we can change this polynomial to the new $\underline{P}(x) - h$ polynomial. This new polynomial also applies in the case. The point $X(0, \underline{a}_0, \underline{a}_1, \underline{a}_2, \dots, \underline{a}_n)$ is an answer to (2) in which a_j 's are the polynomial coefficients obtained. The best approximation of a fuzzy function exists and is unique [Abbasband, 2006].

The value of the best approximation \tilde{f} at point of x_i is:

In order to make more expressive and explain how to use the method, we offer a numerical example. Using the proposed linear programming problems, we will be able to obtain the best fuzzy approximation for the following table data and approximate at the points among information and we do fuzzy interpolation;

As you can see, $m = 3$ and $n = 2$. By solving the following three linear programming problems we will have:

$$\tilde{\beta} = (\beta_m, \beta_l, \beta_r) = (0,0,0)$$

For example, at point of $x = 1.6$ we perform fuzzy interpolation and approximate the value of the function:

CONCLUSION

The problem of interpolating functions is one of the most practical issues in practical mathematics, which is proposed in various fields of science and engineering. Given the practical issues, the use of interpolation methods for data is absolutely essential.

In this study, we used fuzzy mathematics that is one of the successful works of the world of practical mathematics and we study the fuzzy interpolation.

According to the investigations carried out in this paper, we interpolated the data with the help of the fuzzy natural cubic spline. Every day, the tendency to use of fuzzy interpolation increases in various ways, so work in this area is welcoming and is rapidly growing and evolving.

We also found the best approximation of a fuzzy function in order to apply fuzzy interpolation, by optimizing the fuzzy polynomial.

REFERENCE

- [1] O. Kaleva, Interpolation of Fuzzy Data, Fuzzy Sets and Systems, 61 (1994), 63-70.
- [2] R. Lowen, A fuzzy lagrange interpolation theorem, Fuzzy Sets and Systems, 34 (1990), 33-38.
- [3] S. Abbasbandy and E. Babolian, Interpolation of Fuzzy Data by Natural Splines, The Korean J. Comput. & Appl Math., 5: 2 (1998), 457-463.
- [4] S. Abbasbandy, Interpolation of Fuzzy Data by Complete Splines, The Korean J. Comput. & Appl Math, 8: 3 (2001), 587-594.
- [5] A. M. Anile, B. Falcidieno, G. Gallo, M. Spagnuolo and S. Spinello, Modeling uncertain data with fuzzy B-splines, Fuzzy Sets and Systems, 113 (2000), 397-410.

-
- [6] G. Gallo, M. Spagnuolo and S. Spinello, Fuzzy B-splines: A surface model encapsulating uncertainty, *Graphical Models*, 62 (2000), 40-55.
- [7] S. Abbasbandy, R. Ezzati and H. Behforooz, Interpolation of fuzzy data using fuzzy splines, *internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 16 (2008) 107-15.
- [8] S. Abbasbandy, M. Amirfakhrian, A new approach to Universal approximation of the fuzzy function on a discrete set of points, *Appl. Math Modeling* 30 (2006) 1525-1534.
- [9] S. Abbasbandy, M. Amirfakhrian, Numerical approximation Of fuzzy function by fuzzy polynomials, *Appl. Math Comput* 174 (2006) 1001-1006.

How to cite this article:

Khodaparast E. Interpolation of fuzzy data. *J. Fundam. Appl. Sci.*, 2017, 9(2S), 1191-1201.