Research Article

Fuzzy Least Squares Approximation Using Fuzzy Polynomial

Kun Liu 1 and Xiaobin Guo 2

1School of Mathematics and Statistics, Longdong University, Qingyang 745000, China
2College of Mathematics and Statistics, Northwest Normal University, Lanzhou 730070, China

Correspondence should be addressed to Kun Liu; liukunws@163.com

Received 8 March 2021; Revised 2 July 2021; Accepted 26 July 2021; Published 2 September 2021

1. Introduction

There are many systems in which part or all parameters may be uncertain and can be represented and computed by the fuzzy numbers in real world. So, both theory and applications of fuzzy mathematics have been paid more attention by many scholars in the past decades, see [1–24].

In production practice, it is often necessary to conduct statistics and analysis on a large number of data. Due to equipment or technology and other reasons, many measurement or experimental data are uncertain. Sometimes, this property can be expressed and processed by means of fuzzy numbers. Therefore, the problem of numerical approximation based on the uncertainty of fuzzy numbers has attracted more and more attention of scholars in the field of fuzzy mathematics in recent years. Since a large number of data exist as an uncertain property and need a function relation to reflect the laws between different variables, the fuzzy interpolation and approximation problem has been getting more and more important. However, the research work to fuzzy interpolation and approximation was little in the past two decades. In 1990, Lowen [25] firstly introduced the fuzzy Lagrange interpolation theorem. In 1994, fuzzy interpolation was investigated by Kaleva [18] again. Later, a few researchers proposed other ways to construct fuzzy approximation functions. For instance, in 2005, Allahviranloo et al. [5] studied the method of fuzzy number approximation. In 2006, Allahviranloo et al. [1] researched numerical approximation of fuzzy function by using the fuzzy polynomials.

In this paper, we propose the fuzzy least squares problem and find a polynomial function to approximate the given data. The structure of this paper is organized as follows.

In Section 2, we introduce the basic concepts of LR fuzzy numbers. In Section 3, we propose fuzzy least squares approximation problem and present a model to solve fuzzy least squares approximate. In Section 4, we obtain a method which is made of three steps to approximate a fuzzy function. Two numerical examples are given in Section 5.

2. Preliminaries

There are several definitions for the concept of fuzzy numbers (see [10, 25–28]).

Definition 1. A fuzzy number is a fuzzy set like $u: R \rightarrow I = [0, 1]$ which satisfies

(1) $u$ is upper semicontinuous
(2) \( u \) is fuzzy convex, i.e., \( u(\lambda x + (1 - \lambda) y) \geq \min[u(x), u(y)] \) for all \( x, y \in R, \lambda \in [0, 1] \)

(3) \( u \) is normal, i.e., there exists \( x_0 \in R \) such that \( u(x_0) = 1 \)

(4) \( \text{supp} u = \{ x \in R | u(x) > 0 \} \) is the support of \( u \), and its closure \( \text{cl}(\text{supp} u) \) is compact

Let \( E^1 \) be the set of all fuzzy numbers on \( R \).

**Definition 2.** A fuzzy number \( \bar{M} \) is said to be a LR fuzzy number if

\[
\mu_{\bar{M}}(x) = \begin{cases} 
L\left(\frac{m-x}{\alpha}\right), & x \leq m, \alpha > 0, \\
R\left(\frac{x-m}{\beta}\right), & x \geq m, \beta > 0,
\end{cases}
\]

where \( m \) and \( \alpha \) and \( \beta \) are called the mean value and left and right spreads of \( \bar{M} \), respectively. The function \( L(\cdot) \), which is called left shape function, satisfies

1. \( L(x) = L(-x) \)
2. \( L(0) = 1 \) and \( L(1) = 0 \)
3. \( L(x) \) is nonincreasing on \([0, \infty)\)

The definition of a right shape function \( R(\cdot) \) is similar to that of \( L(\cdot) \).

Clearly, two LR fuzzy numbers, \( \bar{M} = (m, \alpha, \beta)_{LR} \) and \( \bar{N} = (n, \gamma, \delta)_{LR} \), are said to be equal if and only if \( m = n, \alpha = \gamma, \) and \( \beta = \delta \). Also, \( \bar{M} = (m, \alpha, \beta)_{LR} \) is positive (negative) if and only if \( m - \alpha > 0 \) (\( m + \beta < 0 \)).

**Definition 3.** For arbitrary LR fuzzy numbers, \( \bar{M} = (m, \alpha, \beta)_{LR} \) and \( \bar{N} = (n, \gamma, \delta)_{LR} \), we have the following.

1. **Addition:**
   \[
   \bar{M} + \bar{N} = (m + n, \alpha + \gamma, \beta + \delta)_{LR},
   \]

2. **Subtraction:**
   \[
   \bar{M} - \bar{N} = (m - n, \alpha + \delta, \beta + \gamma)_{LR},
   \]

3. **Scalar multiplication:**
   \[
   \lambda \bar{M} = (\lambda m, \lambda \alpha, \lambda \beta)_{LR}, \lambda \geq 0,
   \]
   \[
   \lambda \bar{M} = (\lambda m, -\lambda \beta, -\lambda \alpha)_{LR}, \lambda < 0.
   \]

**3. Fuzzy Least Squares Approximation**

**Definition 4.** A fuzzy number value function \( f(x) = (f(x), f'(x), f''(x)) \) is said to be a LR fuzzy number value function if

\[
\mu_f(f(x)) = \begin{cases} 
L\left(\frac{m-f(x)}{f''(x)}\right), & f(x) \leq m, f'(x) > 0, \\
R\left(\frac{f(x)-m}{f''(x)}\right), & f(x) \geq m, f'(x) > 0,
\end{cases}
\]

where \( m \) and \( f'(x) \) and \( f''(x) \) are called the mean value and left and right spreads of \( f(x) \), respectively. The function \( L(\cdot) \), which is called left shape function, satisfies

1. \( L(f(x)) = L(-f(x)) \)
2. \( L(0) = 1 \) and \( L(1) = 0 \)
3. \( L(f(x)) \) is nonincreasing on \([0, \infty)\)

The definition of a right shape function \( R(\cdot) \) is similar to that of \( L(\cdot) \).

**Lemma 1** (see [25, 29–31]). Least squares principle.

Least squares method is using curve \( y = \varphi(x) \) to approximate data \( (x_i, y_i) \) (\( i = 1, 2, \ldots, m \)). The standard is that you want to choose \( \varphi(x) \), and it makes the difference between the function \( \varphi(x_i) \) at \( x_i \) and the measured data \( y_i \) very small. If you want to do that, you have to minimize the sum of squares of deviation, which is \( \sum_{i=1}^{m} [\varphi(x_i) - y_i]^2 \) a minimum.

Minimum square error:

\[
\delta_1 = \sqrt{\sum_{i=1}^{m} w_i (\varphi(x_i) - y_i)^2}. \tag{6}
\]

Maximum deviation:

\[
\delta_2 = \max_{1 \leq i \leq m} |\varphi(x_i) - y_i|. \tag{7}
\]

**3.1. LR Fuzzy Least Squares Approximation Problem.** Consider the following set of data for \( x_i \) and \( f(x_i) \), where \( x_i \in R, x_1 < x_2 < \ldots < x_m \), \( f(x_i) \) are LR numbers, and \( f(x_i) = (f(x_i), f'(x_i), f''(x_i)) \).

\[
\begin{array}{cccc}
\bar{x}_1 & \bar{x}_2 & \bar{x}_3 & \ldots & \bar{x}_m \\
f(x_1) & f(x_2) & f(x_3) & \ldots & f(x_m)
\end{array}
\]

Select a function class in \( C[a, b] \), i.e.,

\[
S = \text{Span}\{\varphi_0(x), \varphi_1(x), \ldots, \varphi_n(x)\}. \tag{8}
\]

Meanwhile, let \( X = \{x_1, x_2, \ldots, x_m\} \) be a set of \( m \) points in \( R \), and \( m > n \). We suppose

\[
\bar{p}(x) = \left( p(x), p'(x), p''(x) \right) = \left( \sum_{j=0}^{n} a_j \varphi_j(x) \right) \sum_{j=0}^{n} b_j \varphi_j(x),
\]

\[
\sum_{j=0}^{n} c_j \varphi_j(x) \right).
\]

We seek \( p(x), p'(x), p''(x) \), making
where $w_i$ ($i = 1, 2, \ldots, m$) are the weight function, and $w_i > 0$.

Therefore, $\bar{p}(x) = (p(x), p'(x), p''(x))$ is the solution of fuzzy least squares approximation.

**Lemma 2** (see [25]). The canonical system $Ga = d$ has an unique solution $(a_0, a_1, \ldots, a_n)$, and the corresponding function $p(x) = \sum_{j=0}^n a_j \phi_j(x)$ is satisfied with equation (11); therefore, $p(x)$ is the least squares solution of the data set $(x_i, f(x_i))$.

**Theorem 1.** The three canonical systems $Ga = d, Gb = e$, and $Gc = f$ have a unique solution $(a_0, a_1, \ldots, a_n)$, $(b_0, b_1, \ldots, b_n)$, $(c_0, c_1, \ldots, c_n)$, separately, and the corresponding function $p(x) = \sum_{j=0}^n a_j \phi_j(x), p'(x) = \sum_{j=0}^n b_j \phi_j(x)$, and $p''(x) = \sum_{j=0}^n c_j \phi_j(x)$ are satisfied with equations (11)–(13); therefore, $\bar{p}(x)$ are the least squares solutions of the data set $(x_i, f(x_i))$.

**Proof.** Suppose $\bar{p}(x) = (p(x), p'(x), p''(x)) = (a_0 \phi_0(x) + a_1 \phi_1(x) + \cdots + a_n \phi_n(x), b_0 \phi_0(x) + b_1 \phi_1(x) + \cdots + b_n \phi_n(x), c_0 \phi_0(x) + c_1 \phi_1(x) + \cdots + c_n \phi_n(x))$; then,

$$p(x) = a_0 \phi_0(x) + a_1 \phi_1(x) + \cdots + a_n \phi_n(x),$$

$$p'(x) = b_0 \phi_0(x) + b_1 \phi_1(x) + \cdots + b_n \phi_n(x),$$

$$p''(x) = c_0 \phi_0(x) + c_1 \phi_1(x) + \cdots + c_n \phi_n(x).$$

Based on least squares solution, $p(x) = a_0 \phi_0(x) + a_1 \phi_1(x) + \cdots + a_n \phi_n(x)$ is satisfied with equation (11). If we want to get the least squares solution, we must resolve the coefficient $a_0, a_1, \ldots, a_n$.

In fact, we will solve the coefficient $a_0, a_1, \ldots, a_n$, and it can transform into solve the minimum points $G(a_0, a_1, \ldots, a_n)$ of the multivariate functions $G(a_0, a_1, \ldots, a_n)$:

$$G(a_0, a_1, \ldots, a_n) = \sum_{i=1}^m w_i \left( f(x_i) - \sum_{j=0}^n a_j \phi_j(x_i) \right)^2,$$

and we can obtain $(\partial G/\partial a_k) = 0, k = 0, 1, \ldots, n$:

$$\frac{\partial G}{\partial a_k} = 2 \sum_{i=1}^m w_i \left( f(x_i) - \sum_{j=0}^n a_j \phi_j(x_i) \right) \phi_k(x_i) = 0,$$

$$\sum_{j=0}^n a_j \sum_{i=1}^m w_i \phi_j(x_i) \phi_k(x_i) = \sum_{i=1}^m w_i f(x_i) \phi_k(x_i),$$

for $k = 0, 1, \ldots, n$.

According to the inner product symbol of the discrete point set, equation (16) can be written as

$$\sum_{j=0}^n (\phi_j, \phi_k) a_j = (f, \phi_k), \quad (k = 0, 1, \ldots, n).$$

The matrix form can be expressed as $Ga = d$, i.e.,

$$\begin{pmatrix} (\phi_0, \phi_0) & (\phi_0, \phi_1) & \cdots & (\phi_0, \phi_n) \\ (\phi_1, \phi_0) & (\phi_1, \phi_1) & \cdots & (\phi_1, \phi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\phi_n, \phi_0) & (\phi_n, \phi_1) & \cdots & (\phi_n, \phi_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} (f, \phi_0) \\ (f, \phi_1) \\ \vdots \\ (f, \phi_n) \end{pmatrix},$$

when $\phi_0(x), \phi_1(x), \ldots, \phi_n(x)$ are linearly independent, due to the Gram matrix, $\det(G_{mn}) \neq 0$; therefore, there is unique solution $a_0, a_1, \ldots, a_n$. It can be seen the least squares solution exists. So, $a_0, a_1, \ldots, a_n$ are the unique solution with equation (18). It proved that a unique function can be obtained $p(x) = a_0 \phi_0(x) + a_1 \phi_1(x) + \cdots + a_n \phi_n(x)$.

Let us prove the latter part. For any $p^*(x) = \sum_{j=0}^n a_j^* \phi_j(x) \in S$, then

$$\sum_{i=1}^m w_i [f(x_i) - p^*(x_i)]^2 = \sum_{i=1}^m w_i [f(x_i) - p(x_i) + p(x_i) - p^*(x_i)]^2$$

$$= \sum_{i=1}^m w_i [f(x_i) - p(x_i)]^2 + 2 \sum_{i=1}^m w_i [f(x_i) - p(x_i)] [p(x_i) - p^*(x_i)]$$

$$+ \sum_{i=1}^m w_i [f(x_i) - p^*(x_i)]^2.$$

From equation (16), we know...
\[
\sum_{i=1}^{m} w_i [f(x_i) - p(x_i)] [p(x_i) - p^*(x_i)] = \sum_{i=1}^{m} w_i [f(x_i) - p(x_i)] \left[ \sum_{j=0}^{n} (a_j - a_j^*) \varphi_j(x_i) \right] \\
= \sum_{j=0}^{n} (a_j - a_j^*) \left[ \sum_{i=1}^{m} w_i [f(x_i) - p(x_i)] \varphi_j(x_i) \right] = 0,
\]

and thus,

\[
\sum_{i=1}^{m} w_i [f(x_i) - p^*(x_i)]^2 = \sum_{i=1}^{m} w_i [f(x_i) - p(x_i)]^2 + \sum_{i=1}^{m} w_i [p(x_i) - p^*(x_i)]^2 \\
\geq \sum_{i=1}^{m} w_i [f(x_i) - p(x_i)]^2,
\]
i.e.,

\[
\sum_{i=1}^{m} w_i [f(x_i) - p(x_i)]^2 = \min_{p(x) \in \mathcal{S}} \sum_{i=1}^{m} w_i [f(x_i) - p^*(x_i)]^2.
\]  \hfill (22)

Therefore, \( p(x) \) is the least squares solution of data set \((x_i, f(x_i)) (i = 1, 2, \ldots, m) \), and it is satisfied with equation (10).

In the same way, \( Gb = e, Gc = g \) can solve the \( b_0, b_1, \ldots, b_m, c_0, c_1, \ldots, c_n \) which are the unique solutions of \( p'(x) = b_0 \varphi_0 (x) + b_1 \varphi_1 (x) + \cdots + b_m \varphi_m (x) \), \( p''(x) = c_0 \varphi_0 (x) + c_1 \varphi_1 (x) + \cdots + c_n \varphi_n (x) \), and \( p'(x), p''(x) \) is the least squares solution of data set \((x_i, f'(x_i)), (x_i, f''(x_i)) (i = 1, 2, \ldots, m) \), and they are also satisfied with equation (10).

At last, we can get the fuzzy least squares approximation:

\[
\bar{p}(x) = (p(x), p'(x), p''(x))
\]  \hfill (23)

Minimum square error:

\[
\delta_1 = \left[ \sum_{i=1}^{m} w_i [f(x_i) - p(x_i)]^2 \right]^{1/2}, \quad \delta_1' = \left[ \sum_{i=1}^{m} w_i [f'(x_i) - p'(x_i)]^2 \right]^{1/2}, \quad \delta_1'' = \left[ \sum_{i=1}^{m} w_i [f''(x_i) - p''(x_i)]^2 \right]^{1/2}.
\]  \hfill (24)

Maximum deviation:

\[
\delta_2 = \max_{1 \leq i \leq m} |f(x_i) - p(x_i)|, \quad \delta_2' = \max_{1 \leq i \leq m} |f'(x_i) - p'(x_i)|, \quad \delta_2'' = \max_{1 \leq i \leq m} |f''(x_i) - p''(x_i)|.
\]  \hfill (25)

4. Solving Fuzzy Least Squares Approximation

In order to solve the fuzzy least squares approximation polynomials, we use the least squares principle to approximate each polynomial, where

\[
\bar{f}(x_i) = (f(x_i), f'(x_i), f''(x_i)).
\]

In particular, we generally choose the basis in \( S \) to be \( \{1, x, x^2, \ldots, x^n\} \), and the weight coefficient is \( w_i = 1 (i = 1, 2, \ldots, m) \).

Solve the fuzzy least squares approximation polynomial \( \bar{p}(x) \), which satisfies

\[
\bar{p}(x_i) = \bar{f}(x_i), \quad i = 0, 1, \ldots, m.
\]  \hfill (26)

Because of \( \bar{p}(x) = \bar{f}(x), \bar{p}(x) \) can be converted to solve three polynomials by least squares approximation, that is, \( p(x), p'(x), p''(x) \). We can divide it into three steps:

Step 1: solve \( p(x) \).

\[
\begin{array}{cccccccc}
\bar{x}_1 & x_1 & x_2 & \cdots & x_m \\
\bar{f}(x_1) & f(x_1) & f(x_2) & \cdots & f(x_m) \\
\end{array}
\]

At first, let

\[
p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.
\]  \hfill (27)

Then, calculate

\[
\sum_{i=1}^{m} w_i x_i, \sum_{i=1}^{m} x_i^2, \ldots, \sum_{i=1}^{m} x_i^n, \sum_{i=1}^{m} x_i^{n+1}, \ldots, \sum_{i=1}^{m} x_i^{2n},
\]

\[
\sum_{i=1}^{m} f(x_i), \sum_{i=1}^{m} x_i f(x_i), \sum_{i=1}^{m} x_i^2 f(x_i), \ldots, \sum_{i=1}^{m} x_i^n f(x_i).
\]  \hfill (28)

Last, by solving normal equation,
we get the solution $a_0, a_1, \ldots, a_n$, i.e.,
\[ p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n. \]
Step 2: solve $p'(x)$.
\[
\begin{pmatrix} f'(x_1) & f'(x_2) & \cdots & f'(x_m) \\
\end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m f'(x_i) \end{pmatrix}
\]
At first, let
\[
\begin{pmatrix} \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 & \cdots & \sum_{i=1}^m x_i^n \\
\sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i^3 & \cdots & \sum_{i=1}^m x_i^{n+1} \\
\vdots & \ddots & \vdots & \vdots \\
\sum_{i=1}^m x_i^n & \sum_{i=1}^m x_i^{n+1} & \cdots & \sum_{i=1}^m x_i^{2n} \\
\end{pmatrix} \begin{pmatrix} a_0 \\
a_1 \\
\vdots \\
a_n \\
\end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m f(x_i) \sum_{i=1}^m x_i f(x_i) \sum_{i=1}^m x_i^2 f(x_i) \sum_{i=1}^m x_i^3 f(x_i) \sum_{i=1}^m x_i^4 f(x_i) \vdots \sum_{i=1}^m x_i^n f(x_i) \end{pmatrix},
\]
(29)

\[ p'(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n. \]
Then, calculate
\[
\sum_{i=1}^m f'(x_i), \sum_{i=1}^m x_i f'(x_i), \sum_{i=1}^m x_i^2 f'(x_i), \cdots, \sum_{i=1}^m x_i^n f'(x_i).
\]
(31)

Last, by solving normal equation,
\[
\begin{pmatrix} \sum_{i=1}^m 1 \\
\sum_{i=1}^m x_i \\
\sum_{i=1}^m x_i^2 \\
\vdots \\
\sum_{i=1}^m x_i^n \\
\end{pmatrix} \begin{pmatrix} b_0 \\
b_1 \\
b_2 \\
\vdots \\
b_n \\
\end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m f'(x_i) \end{pmatrix},
\]
(32)

\[ p'(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n. \]
Then, calculate
\[
\sum_{i=1}^m f'(x_i), \sum_{i=1}^m x_i f'(x_i), \sum_{i=1}^m x_i^2 f'(x_i), \cdots, \sum_{i=1}^m x_i^n f'(x_i).
\]
(34)

Last, by solving normal equation,
\[
\begin{pmatrix} \sum_{i=1}^m 1 \\
\sum_{i=1}^m x_i \\
\sum_{i=1}^m x_i^2 \\
\vdots \\
\sum_{i=1}^m x_i^n \\
\end{pmatrix} \begin{pmatrix} c_0 \\
c_1 \\
c_2 \\
\vdots \\
c_n \\
\end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m f'(x_i) \end{pmatrix},
\]
(35)

we get the solution $c_0, c_1, \ldots, c_n$, i.e.,
\[ p'(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n. \]

From the above steps, we can get the fuzzy least squares approximation polynomial:

Remark 1. Sometimes the fitting function can be of hyperbolic,
\[ p(x) = ae^{bt}, \]  

or exponential type, 

\[ p(x) = \frac{t}{at + b} \]  

(38)

By means of variable substitution, 

\[ Y = \frac{1}{y'}, \quad X = \frac{1}{t'} \]  

(39)

or 

\[ Y = \ln(y), \quad X = \frac{1}{t'} \]  

(40)

we transform the fitting problem into a polynomial fitting of low order, 

\[ S(X) = A + BX, \]  

(41)

to complete it.

5. Numerical Examples

Example 1. Consider the following data.

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(x_i) )</td>
<td>(1, 1, 3) (2, 1, 5) (5, 3, 13)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 1: solve \( p(x) \).

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(x_i) )</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Let \( p(x) = a_0 + a_1x + a_2x^2 \):

\[
\sum_{i=1}^{3} x_i = 3, \\
\sum_{i=1}^{3} x_i^2 = 5, \\
\sum_{i=1}^{3} x_i^3 = 9, \\
\sum_{i=1}^{3} x_i^4 = 17, \\
\sum_{i=1}^{3} f(x_i) = 8, \\
\sum_{i=1}^{3} x_i f(x_i) = 12, \\
\sum_{i=1}^{3} x_i^2 f(x_i) = 22.
\]  

Then,

\[
\begin{pmatrix}
3 & 3 & 5 \\
3 & 5 & 9 \\
5 & 9 & 17
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
a_2
\end{pmatrix} =
\begin{pmatrix}
8 \\
12 \\
5
\end{pmatrix}.
\]

(43)

We get the solution \( a_0 = 1, a_1 = 0, a_2 = 1 \), i.e., 
\( p(x) = a_0 + a_1x + a_2x^2 \).

Step 2: solve \( p'(x) \).

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f'(x_i) )</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Let \( p'(x) = b_0 + b_1x + b_2x^2 \):

\[
\sum_{i=1}^{3} f'(x_i) = 5, \\
\sum_{i=1}^{3} x_i f'(x_i) = 7, \\
\sum_{i=1}^{3} x_i^2 f'(x_i) = 13.
\]  

(44)

Then,

\[
\begin{pmatrix}
3 & 3 & 5 \\
3 & 5 & 9 \\
5 & 9 & 17
\end{pmatrix}
\begin{pmatrix}
b_0 \\
b_1 \\
b_2
\end{pmatrix} =
\begin{pmatrix}
5 \\
7 \\
13
\end{pmatrix}.
\]

(45)

We get the solution \( b_0 = 1, b_1 = -1, b_2 = 1 \), i.e., 
\( p'(x) = 1 - x + x^2 \).

Step 3: solve \( p''(x) \).

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f''(x_i) )</td>
<td>3</td>
<td>5</td>
<td>13</td>
</tr>
</tbody>
</table>

Let \( p''(x) = c_0 + c_1x + c_2x^2 \):

\[
\sum_{i=1}^{3} f''(x_i) = 21, \\
\sum_{i=1}^{3} x_i f''(x_i) = 31, \\
\sum_{i=1}^{3} x_i^2 f''(x_i) = 57.
\]  

(46)

Then,

\[
\begin{pmatrix}
3 & 3 & 5 \\
3 & 5 & 9 \\
5 & 9 & 17
\end{pmatrix}
\begin{pmatrix}
c_0 \\
c_1 \\
c_2
\end{pmatrix} =
\begin{pmatrix}
21 \\
31 \\
57
\end{pmatrix}.
\]

(47)

We get the solution \( c_0 = 3, c_1 = -1, c_2 = 3 \), i.e., 
\( p''(x) = 3 - x + 3x^2 \).

Therefore, the fuzzy least squares approximation polynomial is
\[ \bar{f}(x) = \bar{p}(x) = (p(x), p'(x), p''(x)) = (1 + x^2, 1 - x + x^2, 3 - x + 3x^2). \] (48)

The allowable minimum square error and maximum deviation:
\[ \delta_i = 0, \]
\[ \delta_i^f = 0, \]
\[ \delta_i^p = 0, \]
\[ \delta_i^\prime = 0, \]
\[ \delta_i^{\prime \prime} = 0. \] (49)

**Example 2.** Consider the following data.

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(x_i) )</td>
<td>(1, 1, 3)</td>
<td>(2, 1, 5)</td>
<td>(5, 3, 13)</td>
<td></td>
</tr>
<tr>
<td>( \bar{f}(x_i) )</td>
<td>(1, 1, 3)</td>
<td>(2, 1, 5)</td>
<td>(5, 3, 13)</td>
<td></td>
</tr>
</tbody>
</table>

Let \( p(x) = a_0 + a_1 x \):
\[ \sum_{i=1}^{3} x_i = 7, \]
\[ \sum_{i=1}^{3} x_i^2 = 21, \]
\[ \sum_{i=1}^{3} f(x_i) = 13.7, \]
\[ \sum_{i=1}^{3} x_i f(x_i) = 42, \]
and then,
\[ \begin{pmatrix} 3 & 7 \\ 7 & 21 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 13.7 \\ 42 \end{pmatrix}. \] (50)

We get the solution \( a_0 = -0.45, a_1 = 2.15, \) i.e., \( p(x) = -0.45 + 2.15x. \)

**Step 1:** solve \( p'(x). \)

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f'(x_i) )</td>
<td>0.9</td>
<td>2.1</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Let \( p'(x) = b_0 + b_1 x; \)
\[ \sum_{i=1}^{3} f'(x_i) = 6.8, \]
\[ \sum_{i=1}^{3} x_i f'(x_i) = 20.3. \] (52)

Then,
\[ \begin{pmatrix} 3 & 7 \\ 7 & 21 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} 6.8 \\ 20.3 \end{pmatrix}. \] (53)

We get the solution \( b_0 = 0.05, b_1 = 0.95, \) i.e., \( p'(x) = 0.05 + 0.95x. \)

**Step 2:** solve \( p''(x). \)

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f''(x_i) )</td>
<td>2.8</td>
<td>6.2</td>
<td>11.5</td>
</tr>
</tbody>
</table>

Let \( p''(x) = c_0 + c_1 x; \)
\[ \sum_{i=1}^{3} f''(x_i) = 20.5, \]
\[ \sum_{i=1}^{3} x_i f''(x_i) = 61.2. \] (54)

Then,
\[ \begin{pmatrix} 3 & 7 \\ 7 & 21 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 20.5 \\ 61.2 \end{pmatrix}. \] (55)

We get the solution \( b_0 = 0.15, b_1 = 2.86, \) i.e., \( p''(x) = 0.15 + 2.86x. \)

Therefore, the fuzzy least squares approximation polynomial is
\[ \bar{f}(x) = \bar{p}(x) = (p(x), p'(x), p''(x)) = (-0.45 + 2.15x, 0.05 + 0.95x, 0.15 + 2.86x). \] (56)

The allowable minimum square error and maximum deviation:
\[ \delta_i = 0.0641, \]
\[ \delta_i^f = 0.1539, \]
\[ \delta_i^p = 0.0131, \]
\[ \delta_i^\prime = 0.1, \]
\[ \delta_i^{\prime \prime} = 0.1, \]
\[ \delta_i^p = 0.21. \] (57)

**Remark 2.** Sometimes, we can choose different types of fitting functions for the same set of data. There is only a criterion to decide which function has the best fitting effect, that is, whose error is the minimum.
References


