NEVILLE'S AND LAGRANGE'S INTERPOLATING POLYNOMIAL

Learning Outcomes

At the end of this session, the students should be able to:

1. describe the Lagrangian interpolating polynomial for any given degree;
2. describe the Neville's interpolating polynomial for any given degree; and
3. create R scripts implementing these methods.

Content

I. Polynomial Interpolation
II. Lagrangian Interpolating Polynomial
III. Neville's Interpolating Polynomial

Polynomial Interpolation

Interpolation is a method of constructing new data points within the range of a discrete set of known data points. Given discrete set of known data points $(x_0, f(x_0))$, $(x_1, f(x_1)) \ldots (x_n, f(x_n))$, interpolation is finding the value of $f(x)$ at some value of x that is not given.

Polynomials are the most common choice of interpolants because they are easy to (1) evaluate, (2) differentiate, and (3) integrate. ***Polynomial interpolation*** involves finding a polynomial of order n that passes through $n+1$ the data points.

Lagrangian Interpolating Polynomial

The Lagrange Method seeks to find the interpolating polynomial which passes through the points $(x_1, y_1), \ldots, (x_n, y_n)$ finding n polynomials which follows these rules:

1. The first-order polynomial is y_1 at x_1 and equals zero at $x_2, \ldots, x_n$.
2. The second-order polynomial is y_2 at x_2 and equals zero at $x_1, x_3, \ldots, x_n$.
3. The n^{th}-order polynomial is y_n at x_n and equals zero at $x_1, \ldots, x_{n-1}$.

To do this, we set the following:

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

$$L_i(x) = \prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

Therefore, the ***first-order polynomial*** is:

$$f_1(x) = f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0}$$

And the ***second-order polynomial*** is:

$$f_2(x) = f(x_0)\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f(x_2)\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

The polynomial can also be evaluated at the $n-1$ th order, thus it is imperative that the data is sorted according to the x-coordinate.

Example

We are to estimate $ln(2)$, which has a true value of 0.6931. We are doing this to check if the Lagrangian interpolating polynomial will fair in approximating when we rely on other true values.

x	$f(x)$
1	$ln(1) = 0$
3	$ln(3) = 1.0986$
4	$ln(4) = 1.3863$
5	$ln(5) = 1.6094$

For us to have an interpolating polynomial for four data points, we use a third-order interpolating polynomial.

$$f_3(x) = \sum_{i=0}^{3} L_i(x) f(x_i)$$

$$f_3(x) = \sum_{i=0}^{3} \prod_{j=0, j\neq i}^{3} \frac{x-x_j}{x_i-x_j} f(x_i)$$

$$\begin{aligned} f_3(x) = &\left(\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_2}{x_0-x_2}\right)\left(\frac{x-x_3}{x_0-x_3}\right)f(x_0) + \left(\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_2}{x_1-x_2}\right)\left(\frac{x-x_3}{x_1-x_3}\right)f(x_1) + \\ &\left(\frac{x-x_0}{x_2-x_0}\right)\left(\frac{x-x_1}{x_2-x_1}\right)\left(\frac{x-x_3}{x_2-x_3}\right)f(x_2) + \left(\frac{x-x_0}{x_3-x_0}\right)\left(\frac{x-x_1}{x_3-x_1}\right)\left(\frac{x-x_2}{x_3-x_2}\right)f(x_3) \end{aligned}$$

$$\begin{aligned} f_3(x) = &\left(\tfrac{x-3}{1-3}\right)\left(\tfrac{x-4}{1-4}\right)\left(\tfrac{x-5}{1-5}\right)0 + \left(\tfrac{x-1}{3-1}\right)\left(\tfrac{x-4}{3-4}\right)\left(\tfrac{x-5}{3-5}\right)1.0986 + \\ &\left(\tfrac{x-1}{4-1}\right)\left(\tfrac{x-3}{4-3}\right)\left(\tfrac{x-5}{4-5}\right)1.3863 + \left(\tfrac{x-1}{5-1}\right)\left(\tfrac{x-3}{5-3}\right)\left(\tfrac{x-4}{5-4}\right)1.6094 \end{aligned}$$

$$f_3(2) = 0.66395$$

Neville's Interpolating Polynomial

In the Neville's Method, the evaluation is automatically done. This means that the sought value of the function x should be automatically given for computation to proceed. Also, data should be arranged according to their closeness to x.

$$P_{i,k} = \frac{(x - x_i)\,P_{i+1,k-1} + (x_{i+k} - x)\,P_{i,k-1}}{x_{i+k} - x_i}$$

For easy computation of the values, a table for this purpose is needed.

i	x_i	$\lvert x - x_i \rvert$	$P_{i,0} = f(x_i)$	$P_{i,1}$	...	$P_{i,n-2}$	$P_{i,n-1}$

Example

We are to estimate $ln(2)$, which has a true value of 0.6931. We are doing this to check if the Neville's interpolating polynomial will fair in approximating when we rely on other true values.

x	$f(x)$
1	$ln(1) = 0$
3	$ln(3) = 1.0986$
4	$ln(4) = 1.3863$
5	$ln(5) = 1.6094$

For us to have an interpolating polynomial, we need to construct the following table:

i	x_i	$\lvert x - x_i \rvert$	$P_{i,0} = f(x_i)$	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$
1	1	1	0	0.5493	0.6365	0.66395
2	3	1	1.0986	0.8109	0.7463	
3	4	2	1.3863	0.9401		
4	5	3	1.6094			

Learning Experiences

1. Students will try to manually expand the Lagrangian interpolating polynomials for any given degree.
2. Students will try to manually solve word problems using the given methods.

Sample Exercises for Self-learning

1. Create R functions named `Neville` and `Lagrange` which takes a matrix of data points.
   The functions should have the following ***parameters***:
   1. `mat`, the matrix of data points (for `Neville` and `Lagrange`);
   2. `verbose`, a Boolean with a default of TRUE which prints the table of reliant values (for `Neville`);
   3. `x`, the x-coordinate sought for (for `Neville`);

   The functions must ***return*** the following values:
   1. `f`, the interpolating polynomial being made as a function (for `Lagrange`);
   2. `fx`, the corresponding y-coordinate sought (for `Neville`).

Assessment Tool

A ***programming exercise*** that implements both the Neville's and Lagrange's Interpolating Polynomial.

References

[1] Obrero, R.J. (2015). [Handout 6] Neville's and Lagrange's Interpolating Polynomial (*CMSC 150 old handout)*

[2] Obrero, R.J. (2015). [Exercise 6] Neville's and Lagrange's Interpolating Polynomial (*CMSC 150 old handout)*

[3] Encinas, J.E.I (2012). Lagrange Interpolating Polynomial and Neville's Algorithm *(CMSC 150 old handout)*

Sample Problems

Using the code you have created, answer the following problems below in a 1/2 sheet of paper, or as attached otherwise.

1. Gather the Lagrangian and Neville's Interpolating Polynomial which explains the trend of the Philippine population from 1995 to 2015. Using the gathered polynomial, predict the population in 2004.
   a. For the computation in Lagrange's, place the final interpolating polynomial.
   b. For the computation in Neville's, include the table which was produced by the algorithm, as well as the final output, rounded into the nearest whole number.

2. Plot the Lagrangian Interpolating Polynomial in R from 1990 to 2020, as well as the coordinates of the said function in 2004.
   a. Provide appropriate titles for the graph, and let be the plot of the said coordinate be a solid filled circle of color red. Also, plot the points which make the interpolating polynomial as an x-mark of color blue.
   b. Attach the image that you have produced in Google Classroom. In a form of a comment, interpret the results that you have gathered.

Year	**Population Count**
1995	68 349 452
2000	75 505 061
2005	82 079 348
2010	87 940 171

2015	93 440 274