Newton's Divided Difference

Learning Outcomes

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

1. discuss the concept of interpolation;
2. compute for the polynomial that will fit a set of data points using Newton's Divided Difference; and
3. create an R script that will implement Newton's Divided Difference.

Content

Polynomial Interpolation

- ➔ used to determine the n^{th} order polynomial that fits $n+1$ data points
- ➔ the polynomial should pass through all the points of the discrete set of known points.
- ➔ the polynomial provides a formula to compute intermediate values
- ➔ for a set of $n+1$ data points, there is one and only one polynomial of order n that will fit all the points, the general form of this polynomial is as follows:

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_n x^n$$

Newton's Divided Difference

Linear Interpolation

- ➔ simplest form of interpolation
- ➔ connects two points using a straight line (first order polynomial)
- ➔ to find the first order polynomial that will fit the two data points $(x_0, f(x_0))$ and $(x_1, f(x_1))$, similar triangles can be used

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

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

- ➔ the $f_1(x)$ denotes that this is a first order interpolating polynomial
- ➔ the term $[f(x_1) - f(x_0)]/(x_1 - x_0)$ is a finite-divided-difference

Quadratic Interpolation

- ➔ connects three points using a parabola (second order polynomial)

➔ the form of the second order polynomial that will be used is as follows:

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

$$b_0 = f(x_0)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$b_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

General Form

The general form of the n^{th} order polynomial that will fit a set of $n + 1$ data points is as follows:

$$f_n(x) = b_0 + b_1(x - x_0) + \ldots + b_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

To evaluate the coefficients, $n + 1$ data points and the following equations will be used:

$$b_0 = f(x_0)$$

$$b_1 = f[x_1, x_0]$$

$$b_2 = f[x_2, x_1, x_0]$$

$$b_n = f[x_n, x_{n-1}, \ldots, x_1, x_0]$$

where the bracketed function evaluations are finite divided differences. The first finite divided difference is expressed as:

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

The second divided difference is expressed as:

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}$$

Generally, the nth finite divided difference is expressed as:

$$f[x_n, x_{n-1}, \ldots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \ldots x_1] - f[x_{n-1}, x_{n-1}, \ldots, x_0]}{x_n - x_0}$$

Example: Given four data points, the third order polynomial can be obtained by computing for the following coefficients.

i	x	f(x)	First Order	Second Order	Third Order
0	x_0	$f(x_0)$	$f[x_1, x_0]$	$f[x_2, x_1, x_0]$	$f[x_3, x_2, x_1, x_0]$
1	x_1	$f(x_1)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	
2	x_2	$f(x_2)$	$f[x_3, x_2]$		
3	x_3	$f(x_3)$			

Figure 1: Graphical depiction of the recursive nature of finite divided differences

The interpolating polynomial is expressed as:

$$f_3(x) = f(x_0) + (x - x_0)f[x_1, x_0] + (x - x_0)(x - x_1)f[x_2, x_1, x_0] + (x - x_0)(x - x_1)(x - x_2)f[x_3, x_2, x_1, x_0]$$

Notes

- ➔ It is not necessary for the data points to be equally spaced.
- ➔ It is also not necessary for the abscissa values to be in ascending order

Example

Given the following values of log_{10}, estimate the value of $log_{10}10$ by using the interpolating polynomial.

x	f(x)
8	0.9031
9	0.9542
11	1.0414
12	1.0792

Solution:

x	f(x)	First Order	Second Order	Third Order
8	0.9031	$f[x_1, x_0] = 0.0511$	$f[x_2, x_1, x_0] = -0.0025$	$f[x_3, x_2, x_1, x_0] = 0.0001416667$
9	0.9542	$f[x_2, x_1] = 0.0436$	$f[x_3, x_2, x_1] = -0.001933$	
11	1.0414	$f[x_3, x_2] = 0.0378$		
12	1.0792			

Newton's Interpolating Polynomial:

$$f_3(x) = 0.9031 + (x - 8)(0.0511) + (x - 8)(x - 9)(-0.0025) + (x - 8)(x - 9)(x - 11)(0.0001416667)$$

Evaluating $f_3(10)$ will yield 1.000017 (The true value of $log_{10}10$ is 1).

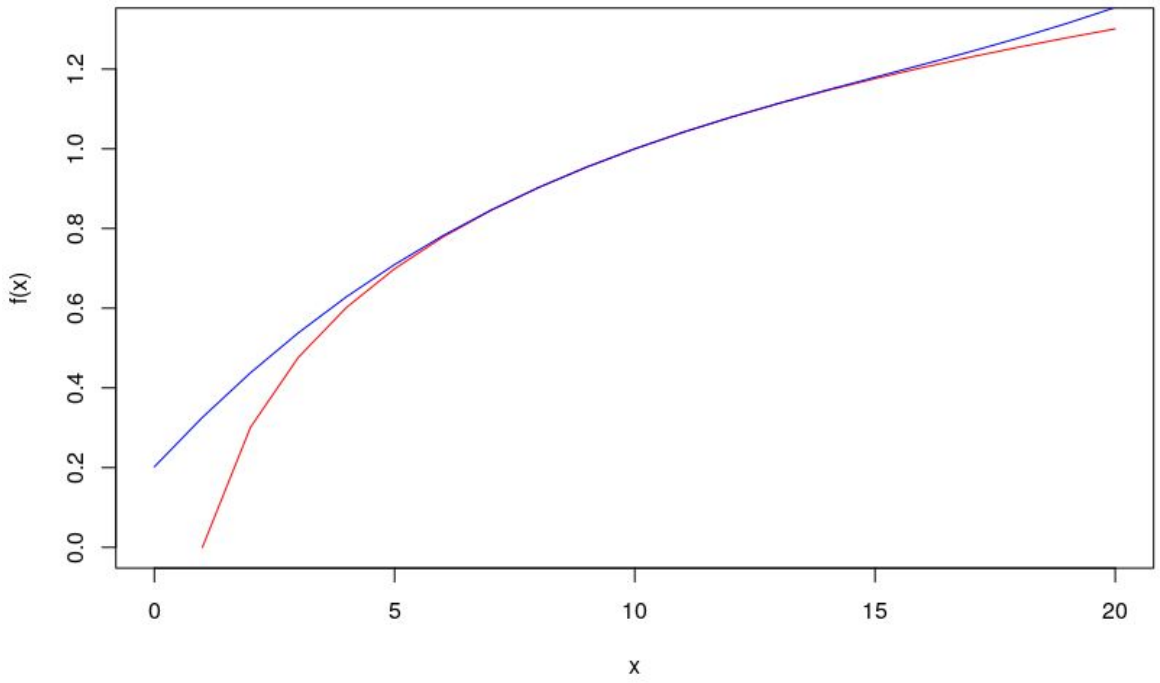

Figure 2: Plot of the $log_{10}x$ (red) and the interpolating polynomial (blue)

Learning Experiences

1. Students will manually compute for the interpolating polynomial that will fit a given data points using ***Newton's Divided Difference.***
2. Students will create their own R script that will solve for the ***interpolating polynomial*** of the given data points using Newton's Divided Difference.
3. Students will attempt to accomplish ***sample exercises for self-learning*** provided below.

Sample Exercises for Self-learning

1. *Required Competencies*: Regression, Interpolation
   The following data define the sea-level concentration of dissolved oxygen for fresh water as a function of temperature:

T°C	0	8	16	24	32	40
o, mg/L	14.621	11.843	9.870	8.418	7.305	6.413

   a. Solve for the function that will model the data using linear regression.
   b. Solve for the function that will model the data using polynomial regression of degree five.
   c. Solve for the function that will model the data using Newton's divided difference
   d. Estimate o(27°C) using the three functions.
   e. Plot the three functions and the data points in a graph.
   f. Compare the three functions using the graph.

Assessment Tool

A ***programming exercise*** on the implementation of Newton's Divided Difference written as an ***R script***.

References

[1] Chapra, S. C., & Canale, R. P. (2015). ***Numerical methods for engineers*** (7th ed.). Boston ; New Delhi: McGraw-Hill Higher Education.

[2] Rick Jason Obrero. (2015). ***Interpolation: Newton's Divided Difference***. CMSC 150 Handout 5.