DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Faculties of Computer Science



Bachelors of Science in Computer Science and Information

Technology

(BSc. CSIT)

Course: Numerical Method (CSC-207)

Class of 2027/Semester: III

A Lab Report On:

**Interpolation and Regression**

**Submitted to:**

Ganesh Thapa

Faculty, Department of Computer Science

**Submitted by:**

Name:

Roll No:

# LAB 2

**TITLE: Interpolation and Regression**

**OBJECTIVES:**

To study different interpolation methods used to find the value for independent variable using provided reference data.

**THEORY:**

Interpolation: The process of finding the value of f(x) for the corresponding ‘x’ value by using reference data is known as interpolation. Interpolation is primarily done to find the unknown f(x) within the range of available data. Extrapolation, a process like Interpolation is however done to find out the values of f(x) for the values of ‘x’ that lies outside the given dataset.

Interpolation can be applied to two different cases based on the characteristics of the given dataset. The processes can be identified such that, if it is applicable to equally spaced data or unequally spaced data.

**Newton’s Forward Interpolation**:

Newtons Forward Interpolation is the type of interpolation that involves finding the unknown values of f(x) for value of ‘x’ off equally spaced dataset.

Newtons Forward Interpolation formula is as follows:

Here, to note:

P = ; where x is the intended value of x, for which the f(x) is to be found

h = common height of the dataset

For

Newton’s forward difference table is given as:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

In similar manner, the values of f(x) for values of x can be found using newtons forward interpolation method. To note, this method only applies for dataset with equally spaced data.

For unequally spaced data, either **Lagrange’s interpolation method** or the newtons divided difference interpolation method can be performed

Lagrange’s Interpolations is used to find the values of f(x) for given x, in data sets which is not necessarily equally space. Lagrange’s interpolations use nth order polynomials for n+1 data points.

The polynomial can be constructed as follows:

f(x) =

where, are coefficients and can be found by substituting f

in the equation.

For x =

=

Similarly, For x =

=

From the given equations,

Along with this method, **Newton’s divided difference method** can also be used for same context, i.e. when the data are unequally spaced.

Formula for Newtons divided difference is given as:

For the values of is given by the forward difference table given as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Regression:**

Regression is defined as the process of finding relations between two variables, one being independent while the other being dependent on one other. Regression is also commonly termed curve fitting.

Interpolation helps us find the unknown values of f(x) for given values of x within the dataset, whereas regression helps us to predict the data observing its tendencies in later stages of the dataset, if the data formulates a particular type of curve.

Regression coefficients using least square method:

Here a normal equation for a linear regression is:

From the normal equations we obtain, the values of a and b which are the coefficients of the normal equations for a linear regression. Particularly, ‘b’ provides the regression coefficient.

**PRACTICAL IMPLEMENTATION**

**TASK 1: NEWTON'S FORWARD INTERPOLATION METHOD**

**ALGORITHM:**

Input: n, x[20], y[20], xp

Output: yp

1. Declare variables n, i, j as integer.
2. Declare variables x[20] and y[20] to store the x and y values.
3. Declare a 2D array diff[20][20] to store the forward differences.
4. Declare variables xp, p, h, nr, dr, and yp as floating point numbers for calculations.
5. Read n, x, y
6. Read xp
7. Calculate Differences:
   1. Calculate h = x[1] – x[0] as the difference between the second and first x values.
   2. For i from 0 to n-1:
      1. Calculate diff[i][1] as y[i+1] - y[i].
   3. For j from 2 to n:

For i from 0 to n-j:

1. Calculate diff[i][j] as diff[i+1][j-1] - diff[i][j-1].
2. Calculate Interpolation:
3. Calculate p as (xp - x[0]) / h.
4. Initialize yp with y[0].
5. For k from 1 to n-1:
6. Calculate nr \*= (p - k + 1).
   1. Calculate dr \*= k.
   2. Update yp += (nr / dr) \* diff[i][k].
7. Output Result:

Print the interpolated value yp for the given xp..

**SOURCE CODE:**

*Source Code must be handwritten*

**OUTPUT:**

*Output must be screenshot of your output and must display your name as well.*

**TASK 2: LAGRANGE'S INTERPOLATION METHOD**

**ALGORITHM:**

Input: n, x[100], y[100], xp

Output: yp

1. Declare arrays x[100] and y[100] to store the x and y values.
2. Declare variables xp and yp for calculations.
3. Declare variables n, i, j, and p.
4. Read n, x, y
5. Read xp

6. Interpolate using Lagrange's method:

a. For i from 1 to n:

i. Initialize p to 1.

ii. For j from 1 to n:

- If i is not equal to j, calculate p \*= (xp - x[j]) / (x[i] - x[j]).

iii. Update yp += p \* y[i].

7. Output Result:

- Print the interpolated value yp for the given xp.

**SOURCE CODE:**

**OUTPUT:**

**TASK 3: NEWTON'S DIVIDED DIFFERENCE INTERPOLATION**

**ALGORITHM:**

Algorithm Newton Interpolation:

Input: n, x[10], y[10], k

Output: f

1. Initialize variables:

- Declare arrays x[10], y[10], and p[10] to store x values, y values, and divided differences.

- Declare variables k, f, n, f1, f2, i, j = 1;

2. Read n, x, y

3. Read k

4. Calculate Divided Differences:

a. Use a do-while loop until n becomes 1:

i. For i from 1 to n-1:

- Calculate p[i] = (y[i+1] - y[i]) / (x[i+j] - x[i]).

- Update y[i] = p[i].

ii. Initialize f1 = 1.

iii. For i from 1 to j:

- Calculate f1 \*= (k - x[i]).

iv. Update f2 += y[1] \* f1.

v. Decrement n and increment j.

5. Calculate Interpolation:

- Update f += f2.

6. Output Result:

- Print the interpolated value f for the given k.

**SOURCE CODE:**

**OUTPUT:**

**TASK 4: LINEAR REGRESSION**

Solving the non- linear equation x3 - 2x – 5 correct upto 3 decimal places using secant method.

**ALGORITHM:**

1. Start
2. Initialize variables: n,i as integer
3. Initialize variable x[20],y[20], sumX = 0, sumX2 = 0, sumY = 0, sumXY = 0, a, b as floating point number.

4. Read Number of Data (n)

5. For i=1 to n:

Read Xi and Yi

Next i

6. Calculate Required Sum

For i=1 to n:

sumX = sumX + Xi

sumX2 = sumX2 + Xi \* Xi

sumY = sumY + Yi

sumXY = sumXY + Xi \* Yi

Next i

7. Calculate Required Constant a and b of y = a + bx:

b = (n \* sumXY - sumX \* sumY)/(n\*sumX2 - sumX \* sumX)

a = (sumY - b\*sumX)/n

8. Display value of a and b

9. Print the equation of the best fit line: y = a + bx.

**SOURCE CODE:**

**OUTPUT:**

**CONCLUSION:**

On the completion of this project, I was able to distinguish different procedures of interpolation and its case wise applications when various kind of data sets are available.

The project also gave me some insight into Regression.