Unit-2

Interpolation and Regression:

A> INTERPOLATION:

⊛ What is interpolation?

⇒ It is the process of estimating the value of $y=f(x)$ corresponding to the value of x, between two known data points.

It is the method of constructing new data points within the range of a discrete set of known data points. It represents the values of a function for a limited number of values of the independent variable y.

⊛ What is extrapolation?

⇒ Extrapolation is the process of calculating unknown values beyond the given data points.

Applications:

i) These techniques are used to find analytic function $f(x)$ that passes through the given set of data points and to use that function to evaluate for our desired value of x.

ii) These techniques are the foundation stone of many widely used applications in the field of science, economics and engineering.

iii) Most of the modern day measurements are discrete in nature and we do not have an analytic function to represent these set of measurement. In these cases interpolation and extrapolation are used to find that function.

iv) Interpolation and extrapolation play vital role to find a general function or an approximate values required for intermediate conditions

for e.g:- Weather data center measuring the temprature for every hour of a day is interpolation. But estimating the temprature for next day is extrapolation.

Scanned with CamScanner

⊛. LAGRANGE INTERPOLATION:

Derivation: A second order polynomial can be written in the form;

$$P_2(x) = b_1(x-x_0)(x-x_1) + b_2(x-x_1)(x-x_2) + b_3(x-x_2)(x-x_0) \quad \ldots (i)$$

Let (x_0, f_0), (x_1, f_1) and (x_2, f_2) are three interpolating points.
Substituting these points in equation (i) we get,

$$P_2(x_0) = f_0 = b_2(x_0-x_1)(x_0-x_2)$$

$$P_2(x_1) = f_1 = b_3(x_1-x_2)(x_1-x_0)$$

$$P_2(x_2) = f_2 = b_1(x_2-x_0)(x_2-x_1)$$

याद गर्ने तरिका
b_1, b_2, b_3 पहिला लेख्ने
त्यसपछि $b_1 \to 0$ to 1
$b_2 \to 1$ to 2
$b_3 \to 2$ to 0

x_0 नभएको term b_2 वाला छ त्यसलाई लिएको र x लाई x_0 ले replace गरेको
Similarly for others.

From above three equations we can calculate value of b1, b2, and b3, as;

$$b_2 = \frac{f_0}{(x_0-x_1)(x_0-x_2)}$$

$$b_3 = \frac{f_1}{(x_1-x_2)(x_1-x_0)}$$

$$b_1 = \frac{f_2}{(x_2-x_0)(x_2-x_1)}$$

Substituting these values of b_1, b_2 and b_3 in equation (i) we get,

$$P_2(x) = \frac{f_2(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} + \frac{f_0(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + \frac{f_1(x-x_2)(x-x_0)}{(x_1-x_2)(x_1-x_0)}$$

Let we rearrange the terms:-

$$\boxed{P_2(x) = \frac{f_0(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + \frac{f_1(x-x_2)(x-x_0)}{(x_1-x_2)(x_1-x_0)} + \frac{f_2(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}}$$

This equation can be written as;

$$P_2(x) = f_0 l_0 + f_1 l_1 + f_2 l_2$$

$$= \sum_{i=0}^{2} f_i l_i(x) \quad \text{—— (ii)} \qquad \text{where, } l_i(x) = \prod_{j=0,\, j\neq i}^{2} \frac{(x-x_j)}{(x_i-x_j)}$$

Generalizing eqn (ii) for $n+1$ points, we get relation

$$\boxed{P_n(x) = \sum_{i=0}^{n} f_i l_i(x)} \quad \text{—— (iii)} \qquad \text{, where, } l_i(x) = \prod_{j=0,\, j\neq i}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

Eq^n (iii) is called lagrange interpolation formula.

⊗. Algorithm and C program for Lagrange interpolation [V.Imp] ✓

Algorithm:

1. Start
2. Read number of points, say n.
3. Read the value at which value is needed, say x.
4. Read given data points.
5. Calculate values of Li as below;

```
for i=1 to n
for j=1 to n
        if (j!=i)
        L[i]=L[i]*((x-x[j])/(x[i]-x[j]))
        End if
End for.
End for.
```

6. Calculate interpolated value at x as below;

```
For i=1 to n
V=V+fx [i] *Lx[i]
End for.
```

7. Print the interpolation value v at x.
8. Terminate.

C Program:

```c
#include<stdio.h>
#include<conio.h>
int main() { int n,i,j;
        float x,l,v=0,ax[10],fx[10],L[10];
        printf("Enter the number of points \n");
        scanf("%d",&n);
        printf ("Enter the value of x \n");
        scanf ("%f", &x);
    for (i=0; i<n; i++)
    { printf("Enter the value of x and fx at i=%d \n", i);
      scanf ("%f %f", &ax[i], &fx[i]);
    }
```

```
for (i=0; i<n; i++)
 { l=1.0; //to make float.
   for (j=0; j<n; j++)
    {   if (j!=i)
         l=l*((a-ax[j])/(ax[i]-ax[j]));
    }
    L[i]=l;
 }

 for (i=0; i<n; i++)
  {
      v=v+fx[i]*L[i];
  }
 printf("Interpolated value=%f",v);
 getch();
 return 0;
}
```

Output:

Enter the number of points
4
Enter the value of x
1
Enter the value of x and fx at i=0
-1   -8
Enter the value of x and fx at i=1
0   3
Enter the value of x and fx at i=2.
2   1
Enter the value of x and fx at i=3.
3   12
Interpolation value=2.00.

Scanned with CamScanner

Example 1: The upward velocity of a rocket is given as a function of time in Table below:

Time (t)	Velocity (v)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

Determine the value of velocity at $t=16$ seconds using a first order Lagrange polynomial.

Solution:-

For first order polynomial interpolation, the velocity is given by,

$$P_1(x) = \sum_{i=0}^{1} l_i(x) f_i$$

$$= l_0(x) f_0 + l_1(x) f_1$$

Since we want to find the velocity at $t=16$, and we are using first order polynomial, we need to choose the two data points that are closest to $t=16$ that also bracket $t=16$. These two points are $t_0=15$ and $t_1=20$. This gives

$$l_0(x) = \prod_{\substack{j=0\\ j\neq 0}}^{1} \frac{x-x_j}{x_0-x_j} = \frac{x-x_1}{x_0-x_1} = \frac{16-20}{15-20} = \frac{4}{5} = 0.8$$

$$l_1(x) = \prod_{\substack{j=0\\ j\neq 1}}^{1} \frac{x-x_j}{x_1-x_j} = \frac{x-x_0}{x_1-x_0} = \frac{16-15}{20-15} = 0.2$$

Thus,

$$P_1(16) = l_0(x) f_0 + l_1(x) f_1$$

$$= 0.8 \times 362.78 + 0.2 \times 517.35$$

$$= 393.69$$

Example 2: The value of e^x is given in table below:

X	e^x
0	1
1	2.7183
2	7.3891
3	20.0855

Scanned with CamScanner

Determine the value of $e^{1.2}$ by using second order polynomial interpolation using Lagrange polynomial interpolation.

Solution:

For second order polynomial interpolation, the polynomial is given by,

$$P_2(x) = \sum_{i=0}^{2} l_i(x) f_i$$

$$= l_0(x) f_0 + l_1(x) f_1 + l_2(x) f_2.$$

Since we want to find value of $e^{1.2}$, and we are using second order polynomial, we need to choose the three data points that are closest to $x=1.2$ that also bracket $x=1.2$. The three data points are $x_0=0$, $x_1=1$ and $x_2=2$. This gives

$$l_0(x) = \prod_{j=0,\, j\neq 0}^{2} \frac{x-x_j}{x_0-x_j} = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(1.2-1)(1.2-2)}{(0-1)(0-2)} = -0.08$$

$$l_1(x) = \prod_{j=0,\, j\neq 1}^{2} \frac{x-x_j}{x_1-x_j} = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(1.2-0)(1.2-2)}{(1-0)(1-2)} = 0.96$$

$$l_2(x) = \prod_{j=0,\, j\neq 2}^{2} \frac{x-x_j}{x_2-x_j} = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(1.2-0)(1.2-1)}{(2-0)(2-1)} = 0.12$$

Now,

$$P_2(1.2) = l_0(x) f_0 + l_1(x) f_1 + l_2(x) f_2.$$

$$= -0.08*1 + 0.96*2.7183 + 0.12*7.3891$$

$$= 3.41626.$$

Scanned with CamScanner

⊛. NEWTON'S DIVIDED DIFFERENCE INTERPOLATION:

Derivation: Let us consider a polynomial of degree n of the form given below;

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}) \quad \text{(i)}$$

To construct the polynomial we need to find coefficients a_0, a_1, a_2 and a_n. Let us suppose $(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_n, f(x_n))$ are given interpolating points.

Now, at $x = x_0$ eqn (i) becomes

$$p(x_0) = f(x_0) = a_0$$

$$\Rightarrow a_0 = f(x_0)$$

Similarly, at $x = x_1$ eqn (i) becomes

x replaced by x_1 since we are using function $f(x_1)$. Similarly for others

$$p(x_1) = f(x_1) = a_0 + a_1(x_1 - x_0)$$

$$\Rightarrow a_1 = \frac{f(x_1) - a_0}{x_1 - x_0}$$

since $a_0 = f(x_0)$

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

at $x = x_2$ eqn (i) becomes

$$p(x_2) = f(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$$

$$\Rightarrow f(x_2) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0} \cdot (x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$$

On further solving we get

$$\Rightarrow a_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

यो याद गर्नै नत solve गरेर निकाल्दा हेरे राम्रो जान्छ

Note that a_0, a_1, and a_2 are finite divided differences. a_0, a_1 and a_2 are the first, second and third finite divided differences, respectively. We denote the first divided difference by;

$$f[x_0] = f(x_0)$$

The second divided difference by,

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Scanned with CamScanner

and the third divided difference by,

$$f[x_2,x_1,x_0] = \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2-x_0}$$

This leads us to writing the general form of the Newton's divided difference polynomial for $n+1$ data points, (x_0, y_0), $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), (x_n, y_n)$, as:

$$P_n(x) = a_0 + a_1(x-x_0) + \ldots + a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}).$$

where,
$$a_0 = f[x_0]$$
$$a_1 = f[x_1, x_0]$$
$$a_2 = f[x_2, x_1, x_0]$$
$$\vdots$$
$$a_{n-1} = f[x_{n-1}, x_{n-2}, \ldots, x_0]$$
$$a_n = f[x_n, x_{n-1}, \ldots, x_0]$$

where the definition of the mth divided difference is:

$$a_m = f[x_m, \ldots, x_0]$$
$$= \frac{f[x_m, \ldots, x_1] - f[x_{m-1}, \ldots, x_0]}{x_m - x_0}.$$

Now the eqn (i) can be written as:

$$P_n(x) = f[x_0] + \sum_{i=1}^{n} f[x_0, x_1, \ldots, x_n] \prod_{j=0}^{i=1} (x - x_j) \quad \text{(ii)}$$

This eqn (ii) is called Newton's divided difference interpolation polynomial.

__Algorithm:__

1. Start
2. Read number of points, say n;
3. Read the value at which interpolated value is needed, say x.
4. Read given data points.
5. Calculate the first divided difference as;

   for $i=0$ to $n-1$.

   $dd[i] = fx[i]$

   End for.

Scanned with CamScanner

6. Calculate second to nth divided difference as;
   For $i=0$. to $i+1$

   for $j=n-1$ to $i+1$

   $dd[j] = \dfrac{dd[j]-dd[j-1]}{x[j]-x[j-1-i]}$

   End for

   End for.

7. Set $v=0$ and $p=1$.
8. Calculate interpolated value as;
   For $i=0$ to $n-1$

   For $j=0$ to $i-1$

   $p=p*(x-x_j)$

   End for

   $v=v+dd[i]*p$

   Reset $p=1$

   End for
9. Print the interpolated value v.
10. Terminate.

Note: Practice C-programs on the basis of algorithms yourself, if programs are not written here. Only important types of programs are written here.

Example: The upward velocity of a rocket is given as a function of time in table below:

time (t)	velocity (v)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

Determine the value of velocity at $t=16$ seconds with third order polynomial using Newton's divided difference polynomial method.

Solution: For a third order polynomial the velocity is given by,

$$P(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2)$$

So, let the 4 data points for a_0, a_1, a_2 and a_3 be $x_0=10$, $x_1=15$, $x_2=20$ & $x_3=22.5$

Scanned with CamScanner

Then,

$x_0 = 10,\ f(x_0) = 227.04$

$x_1 = 15,\ f(x_1) = 362.78$

$x_2 = 20,\ f(x_2) = 517.35$

$x_3 = 22.5,\ f(x_3) = 602.97$

Now, calculate values of divided differences as below:-

$$a_0 = f[x_0] = f(x_0) = 227.04$$

$$a_1 = f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{362.78 - 227.04}{15 - 10} = 27.148$$

$$a_2 = f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0}$$

$$a_3 = f[x_3, x_2, x_1, x_0] = \frac{f[x_3, x_2, x_1] - f[x_2, x_1, x_0]}{x_3 - x_0}.$$

For a_2 we know that

$$f[x_2, x_1] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{517.35 - 362.78}{20 - 15} = 30.914$$

and $f[x_1, x_0] = 27.148$ (from a_1 above that we already calculated)

Now,

$$a_2 = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{30.914 - 27.148}{20 - 10} = 0.37660.$$

Again for a_3 we know that

$$f[x_3, x_2] = \frac{f(x_3) - f(x_2)}{x_3 - x_2} = \frac{602.97 - 517.35}{22.5 - 20} = 34.248$$

$$f[x_2, x_1] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{517.35 - 362.78}{20 - 15} = 30.914$$

$$\Rightarrow f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = \frac{34.248 - 30.914}{22.5 - 15} = 0.44453$$

Therefore,

from a_2 $\therefore a_2 = f[x_2, x_1, x_0]$

$$a_3 = \frac{f[x_3, x_2, x_1] - f[x_2, x_1, x_0]}{x_3 - x_0} = \frac{0.44453 - 0.37660}{22.5 - 10}$$

$$= 5.4347 \times 10^{-3}$$

Scanned with CamScanner

Now, Third order polynomial can be written as:

$$p(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2)$$

$$= 227.04 + 27.148(x-10) + 0.3766(x-10)(x-15) + 0.0055347(x-10)(x-15)(x-20)$$

Now at $x=16$

$$p(16) = 227.04 + 27.148(16-10) + 0.3766(16-10)(16-15) + 0.0055347(16-10)(16-15)(16-20)$$

$$= 392.06.$$

⊗ Divided Difference Table [Imp]

We know that the first order divided differences can be calculated from given interpolating points as below:

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$f[x_2, x_1] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

And second order divided difference can be calculated from first order divided differences below:

$$f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0}$$

Thus we can say that higher order divided differences can be calculated from lower order divided difference recursively by using formula as;

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

Thus from this formula divided difference table for cubic polynomial is as follows;

x_0	$f(x_0) \leftarrow a_0$			
		$f[x_1, x_0] \leftarrow a_1$		
x_1	$f(x_1)$		$f[x_2, x_1, x_0] \leftarrow a_2$	
		$f[x_2, x_1]$		$f[x_3, x_2, x_1, x_0] \leftarrow a_3$
x_2	$f(x_2)$		$f[x_3, x_2, x_1]$	
		$f[x_3, x_2]$		
x_3	$f(x_3)$			

fig. Table of divided differences for a cubic polynomial.

Example 1: Given the following data points, create the table of divided differences. Use the table to estimate the value of $f(1.8)$ by using second and third order polynomial.

	x_0	x_1	x_2	x_3
x	1	2	3	4
$f(x)$	0	7	26	63

Solution:

	$f(x)$	1st	2nd	3rd
x_0	0 ← a_0			
		7 ← a_1		
x_1	7		6 ← a_2	
		19		1 ← a_3
x_2	26		9	
		37		
x_3	63			

Rough

How 7 came?

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

i.e. $\frac{7-0}{x_1 - x_0} = \frac{7-0}{2-1}$

$= 7$

Similarly for others.

How 6 & 9 came?

$\frac{19-7}{x_2 - x_0} = \frac{12}{3-1} = \frac{12}{2} = 6$

& $\frac{37-19}{x_3 - x_1} = \frac{18}{3-1} = \frac{18}{2} = 9$

How 1 came?

$\frac{9-6}{x_3 - x_0} = \frac{3}{4-1} = \frac{3}{3} = 1$

This implies general form to calculate as: $\boxed{\frac{\text{higher} - \text{lower}}{x_? - x_0}}$

Now, Evaluating $f(1.8)$ by using second order polynomial.

We know that,

$$p(x) = a_0 + a_1 (x - x_0) + a_2 (x - x_0)(x - x_1)$$

Therefore

$$f(1.8) = P_2(1.8) = 0 + 7 \times (1.8 - 1) + 6 \times (1.8-1)(1.8-2)$$

$$= 0 + 5.6 - 0.96$$

$$= 4.64$$

Again, Evaluating by using third order polynomial

$$p(x) = a_0 + a_1 (x - x_0) + a_2 (x - x_0)(x - x_1) + a_3 (x - x_0)(x - x_1)(x - x_2)$$

Therefore,

$$f(1.8) = P_3(1.8) = 0 + 7 \times (1.8-1) + 6 \times (1.8-1)(1.8-2) + 1 \times (1.8-1)(1.8-2)(1.8-3)$$

$$= 0 + 5.6 - 0.96 + 0.192$$

$$= 4.832$$

This example shows that higher order polynomials give more accurate value than lower order polynomials.

Scanned with CamScanner

NEWTON'S FORWARD AND BACKWARD DIFFERENCE INTERPOLATION: [Imp]

Newton's – Forward Difference Interpolation:

We know that Newton's – Divided difference polynomial is given by:

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}).$$

i.e., $P_n(x) = f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) + \dots$

$$+ f[x_0,x_1,\dots,x_{n-1},x_n](x-x_0)(x-x_1)\dots(x-x_{n-1}) \quad \text{—(i)}$$

Let us introduce the notation $h = x_{i+1} - x_i$, for each $i = 0,1,\dots,n-1$. Suppose $x = x_0 + sh$ be a point where we are trying to find value of y. & let next point to x be x_k as $x_k = x_0 + kh$. Then subtracting x_k from x we get,

$$x - x_k = x_0 + sh - (x_0 + kh).$$

or, $x - x_k = sh - kh$

or, $x - x_k = (s-k)h$.

This gives:

$$\left.\begin{aligned} x-x_0 &= sh \\ x-x_1 &= (s-1)h \\ &\ \vdots \\ x-x_{n-1} &= (s-n+1)h \end{aligned}\right\}$$

Putting $0,1,\dots,n-1$ in place of k in above $x - x_k = (s-k)h$

Now the Newton's – Divided difference interpolation formula (i) becomes;

$$P_n(x) = P_n(x_0+sh) = f[x_0] + f[x_0,x_1]sh + f[x_0,x_1,x_2]sh^2(s-1) + \dots$$

$$+ f[x_0,x_1,\dots,x_{n-1},x_n]sh^n(s-1)(s-2)\dots(s-n+1) \quad \text{—(ii)}$$

Now the Newton's forward difference formula is constructed by making use of the forward difference notation Δ as:

$$f[x_0,x_1] = \frac{f[x_1]-f[x_0]}{x_1-x_0} = \frac{1}{1!\,h^1}\Delta^1 f(x_0) = \frac{1}{h}\Delta f(x_0)$$

$$f[x_0,x_1,x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_1} = \frac{1}{2!\,h^2}\Delta^2 f(x_0) = \frac{1}{2h^2}\Delta^2 f(x_0).$$

In General

$$f[x_0,x_1,\dots,x_{n-1},x_n] = \frac{1}{n!\,h^n}\Delta^n f(x_0).$$

Scanned with CamScanner

Now, eqn (ii) can be written as;

$$P_n(x)=P_n(x_0+sh)=f[x_0]+\frac{1}{1!h}\Delta f(x_0).sh+\frac{1}{2!h^2}\Delta^2 f(x_0).sh^2(s-1)$$
$$+\dots+\frac{1}{n!h^n}\Delta^n f(x_0).sh^n(s-1)(s-2)\dots(s-n+1).$$

OR

$$\boxed{P_n(x)=P_n(x_0+sh)=f[x_0]+\Delta f(x_0).\frac{s}{1!}+\Delta^2 f(x_0).\frac{s(s-1)}{2!}+\dots+\Delta^n f(x_0).\frac{s(s-1)(s-2)\dots(s-n+1)}{n!} \quad \text{—(iii)}}$$

This eqn (iii) is called Newton-Gregory forward difference formula.

We can construct Newton's Forward differences by construction of Newton's Forward difference table as below:

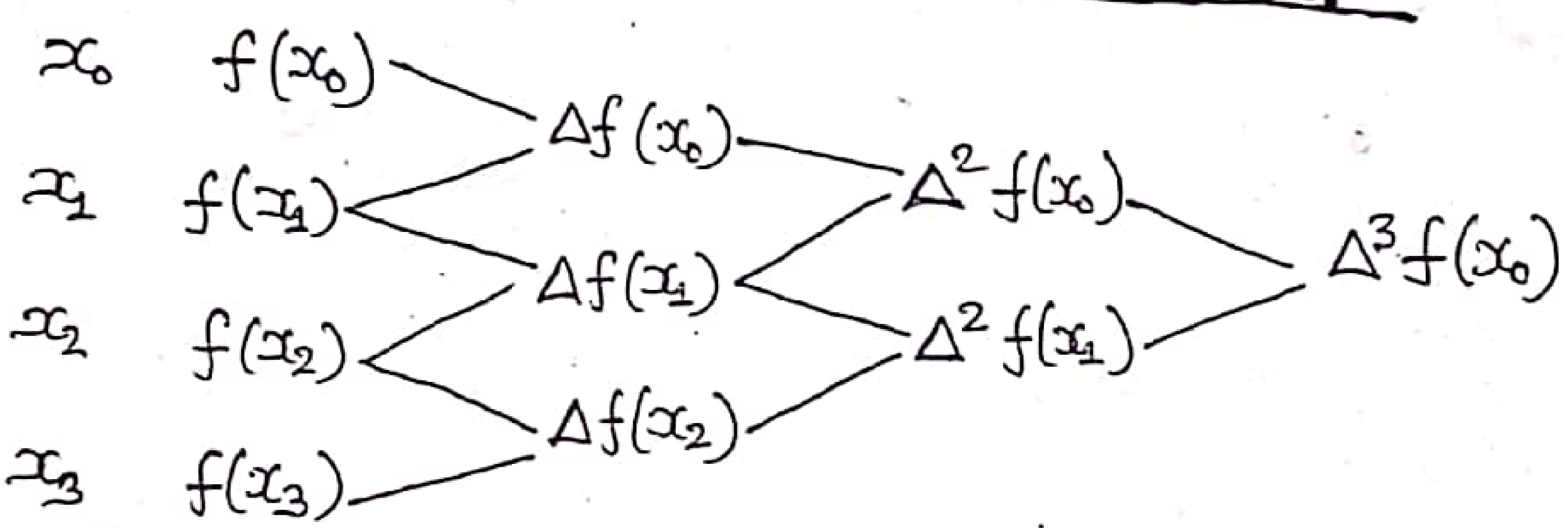

figure: Table of Forward differences for cubic polynomial

Algorithm:

1. Start
2. Read number of data points, say n.
3. Read the value at which interpolated value is needed, say xp.
4. Read n data points, say $x[i]$ and $fx[i]$.
5. Set $h=x[1]-x[0]$ and $s=(xp-x[0])/h$
6. Calculate first forward differences as below

```
    For i=0 to n-1
        fd[i]=fx[i]
    End for
```

7. Calculate second to nth forward differences as below:

```
For i=0 to n-1
    For j=n-1 to i+1
        fd[j] = fd[j] - fd[j-1]
    End for
End for.
```

8. Set $v = fd[0]$ and Set $p=1$.
9. Calculate interpolated value as below:

   For $i=1$ to $n-1$.

      For $k=1$ to i

         $p = p * (s-k+1)$

      End For

         $v = v + \dfrac{fd[i] * p}{i!}$

      Reset $p=1$

   End For.
10. Print the interpolated value v at xp.
11. Terminate.

Example: Construct Newton's forward-difference table for data points given in table below and then approximate the value of $f(1.1)$ by using Newton's forward difference formula and fourth divided difference.

x	1.0	1.3	1.6	1.9	2.2
$f(x)$	0.7651977	0.6200860	0.4554022	0.2818186	0.1103623

Solution:

Here, $h = x_{i+1} - x_i = 0.3$ (i.e, $1.3 - 1.0 = 0.3$)

Since, $x = x_0 + sh$

$$\Rightarrow s = \frac{x - x_0}{h} = \frac{1.1 - 1.0}{0.3} = \frac{1}{3}$$

$-0.1646838 - (-0.1451117)$
$= -0.0195721$
Similarly for $\Delta^3 f(x_0)$ and $\Delta^4 f(x_0)$

x_i	$f(x_i)$	$\Delta f(x_0)$	$\Delta^2 f(x_0)$	$\Delta^3 f(x_0)$	$\Delta^4 f(x_0)$
1.0	0.7651977				
		−0.1451117			
			−0.0195721		
1.3	0.6200860			0.0106723	
		−0.1646838			
1.6	0.4554022		−0.008898		0.0003548
		−0.1735836		0.0110271	
1.9	0.2818186		0.0021273		
		−0.1714563			
2.2	0.1103623				

Rough

$\Delta f(x_0) = f(x_{i+1}) - f(x_i)$

e.g. $= 0.6200860 - 0.7651977$
$= -0.1451117$

Similarly $\Delta^2 f(x_0) = \Delta f(x_{i+1}) - \Delta f$

This is the required Newton's forward table for given data points.

Scanned with CamScanner

Now, We know that,

$$P_n(x) = P_n(x_0+sh) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0)$$

where, $\binom{s}{k} = \dfrac{s(s-1)\ldots(s-k+1)}{k!}$

Thus,

$$P_4(x) = f[x_0] + \binom{s}{1}\Delta f(x_0) + \binom{s}{2}\Delta^2 f(x_0) + \binom{s}{3}\Delta^3 f(x_0) + \binom{s}{4}\Delta^4 f(x_0)$$

$$= f[x_0] + \frac{s}{1!}\Delta f(x_0) + \frac{s(s-1)}{2!}\Delta^2 f(x_0) + \frac{s(s-1)(s-2)}{3!}\Delta^3 f(x_0) + \frac{s(s-1)(s-2)(s-3)}{4!}\Delta^4 f(x_0)$$

Hence,

$$P_4(1.1) = 0.7651977 + \frac{1}{3}(-0.1451117) + \frac{\frac{1}{3}\left(\frac{1}{3}-1\right)}{2}(-0.0195721)$$

$$+ \frac{\frac{1}{3}\left(-\frac{2}{3}\right)\left(-\frac{5}{3}\right)}{6}(0.0106721) + \frac{\frac{1}{3}\left(-\frac{2}{3}\right)\left(-\frac{5}{3}\right)\left(-\frac{8}{3}\right)}{24}(0.0003548)$$

$$= 0.712$$

Scanned with CamScanner

⊛. Newtons - Backward difference Interpolation:

We know that Newtons-divided difference polynomial is given by;

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}).$$

Now if the interpolating points are recorded back as: $x_n, x_{n-1}, \dots, x_1, x_0$.
Then, Newton's-divided difference polynomial can be written as;

$$P_n(x) = a_n + a_{n-1}(x-x_n) + a_{n-2}(x-x_n)(x-x_{n-1}) + \dots + a_0(x-x_n)(x-x_{n-1})\dots(x-x_1).$$

i.e, $P_n(x) = f[x_n] + f[x_n, x_{n-1}](x-x_n) + f[x_n, x_{n-1}, x_{n-2}](x-x_n)(x-x_{n-1})$
$+ \dots + f[x_n, x_{n-1}, \dots, x_1, x_0](x-x_n)(x-x_{n-1})\dots(x-x_1)$ — (i)

Let us introduce notation $h = x_{i+1} - x_i$ for each $i = n-1, \dots, 1, 0$.
Suppose $x = x_n + sh$ be a point where we are trying to find value of y & let previous point to x be x_k as $x_k = x_n + (k-n)h$.
Now, subtracting x_k from x we get,

$$\begin{aligned} x - x_k &= \cancel{x_n} + sh - \cancel{x_n} - (k-n)h \\ &= sh - (k-n)h \\ &= (s-k+n)h. \end{aligned}$$

This gives,

$$\begin{aligned} x - x_n &= (s-n+n)h = sh \\ x - x_{n-1} &= (s-(n-1)+n)h = (s+1)h \\ &\vdots \\ x - x_2 &= (s-2+n)h = (s+n-2)h \\ x - x_1 &= (s-1+n)h = (s+n-1)h \end{aligned}$$

Now the Newtons-divided difference formula (i) becomes;

$$P_n(x) = f[x_n] + f[x_n, x_{n-1}]sh + f[x_n, x_{n-1}, x_{n-2}]sh^2(s+1) + \dots + f[x_n, x_{n-1}, \dots, x_1, x_0]sh^n(s+1)\dots(s+n-2)(s+n-1). \quad \text{(ii)}$$

Now, Newtons backward difference formula can be constructed by making use of backward difference notation ∇ as;

$$f[x_n, x_{n-1}] = \frac{1}{h}\nabla f(x_n)$$

$$f[x_n, x_{n-1}, x_{n-2}] = \frac{1}{2h^2}\nabla^2 f(x_n).$$

In General

$$f[x_n, x_{n-1}, \dots, x_{n-k}, x_k] = \frac{1}{k!\,h^k}\nabla^k f(x_n)$$

Now eqn (ii) can be written as;

$$\boxed{P_n(x) = P_n(x_n+sh) = f[x_n] + \nabla f(x_n)\frac{s}{1!} + \nabla^2 f(x_n)\frac{s(s+1)}{2!} + \dots + \nabla^k f(x_n)\cdot\frac{s(s+1)\dots(s+n-2)(s+n-1)}{k!}}$$

This eqn (iii) is called Newton-Gregory backward difference formula.

Scanned with CamScanner

Algorithm

1. Start
2. Read number of points, say n.
3. Inter the value at which interpolated value is required, say xp.
4. Read n data points.
5. Set $h = x[1] - x[0]$ and $s = (xp - x[n-1])/h$
6. Calculate first backward difference as below:-

    For $i=0$ to $n-1$

        $bd[i] = fx[i]$

    End For
7. Calculate 2nd and nth backward differences as below:

    For $i = n-1$ to 1.

        For $j=0$ to $i-1$

            $bd[j] = bd[j+1] - bd[j]$

        End For
8. Set $v = bd[n-1]$
9. Calculate interpolated value as below;

    For $i=1$ to $n-1$

        For $k=1$ to i

            $p = p*(s+k-1)$

        End For.

        $$V = V + \frac{bd[n-i-1]*p}{i!}$$

        Reset $p=1$

    End for
10. Print interpolated value ~~at~~ v at xp.
11. Terminate.

Example: The sale for the last five years is given in the table below. Find 4th order polynomial that passes through the given data then use the polynomial to estimate the sales for the year 1979.

Year-x	1974	1976	1978	1980	1982
Sales-$f(x)$	40	43	48	52	57

Solution:

Scanned with CamScanner

Here, $h = x_{i+1} - x_i = 2$

Since, $x = x_n + sh \Rightarrow s = \frac{x - x_n}{h} = \frac{1979 - 1982}{2} = \frac{-3}{2}$

x_i	$f(x_i)$	$\nabla f(x_n)$	$\nabla^2 f(x_n)$	$\nabla^3 f(x_n)$	$\nabla^4 f(x_n)$
1974	40				
		3			
1976	43		2		
		5		-3	
1978	48		-1		5
		4		2	
1980	52		1		
		5			
1982	57				

We know that backward divided difference interpolation is given by,

$$P_n(x) = P_n(x_n + sh) = f[x_n] + \nabla f(x_n)s + \frac{1}{2}\nabla^2 f(x_n)s(s+1) + \dots + \frac{1}{n!}\nabla^n f(x_n)s(s+1)\dots(s+n-1)$$

Thus,

$$P_4(x) = P_4(x_n + sh) = f[x_n] + \nabla f(x_n)s + \frac{1}{2!}\nabla^2 f(x_n)s(s+1) + \frac{1}{3!}\nabla^3 f(x_n)s(s+1)(s+2) + \frac{1}{4!}\nabla^4 f(x_n)s(s+1)(s+2)(s+3)$$

Thus, fourth order polynomial is

$$P_4(x) = P_4(x_n + sh) = 57 + 5s + \frac{1}{2}1 \times s(s+1) + \frac{1}{6}2s(s+1)(s+2) + \frac{1}{24}5s(s+1)(s+2)(s+3).$$

Now, put $x = 1979$

$$P_4(1979) = P_4\left(1982 + \left(-\frac{3}{2}\right)\times 2\right) = 57 + 5\left(-\frac{3}{2}\right) + \frac{1}{2}\left(-\frac{3}{2}\right)\left(-\frac{1}{2}\right) + \frac{1}{6}\times 2\times\left(-\frac{3}{2}\right)\left(-\frac{1}{2}\right)\left(\frac{1}{2}\right) + \frac{1}{24}\times 5\times\left(-\frac{3}{2}\right)\left(-\frac{1}{2}\right)\left(\frac{1}{2}\right)\left(\frac{3}{2}\right)$$

$$= 57 - 7.5 + 0.375 + 0.125 + 0.1172$$
$$= 50.1172$$

Hence, Estimated sales of year $1979 = 50.1172$.

Scanned with CamScanner

B> REGRESSION ANALYSIS:

Concept: Regression analysis is a form of predictive modeling technique which investigates relationship between a dependent and independent variables. Dependent variables are also called target variable and independent variables are also called predictors. It is an important tool for modeling and analyzing data.

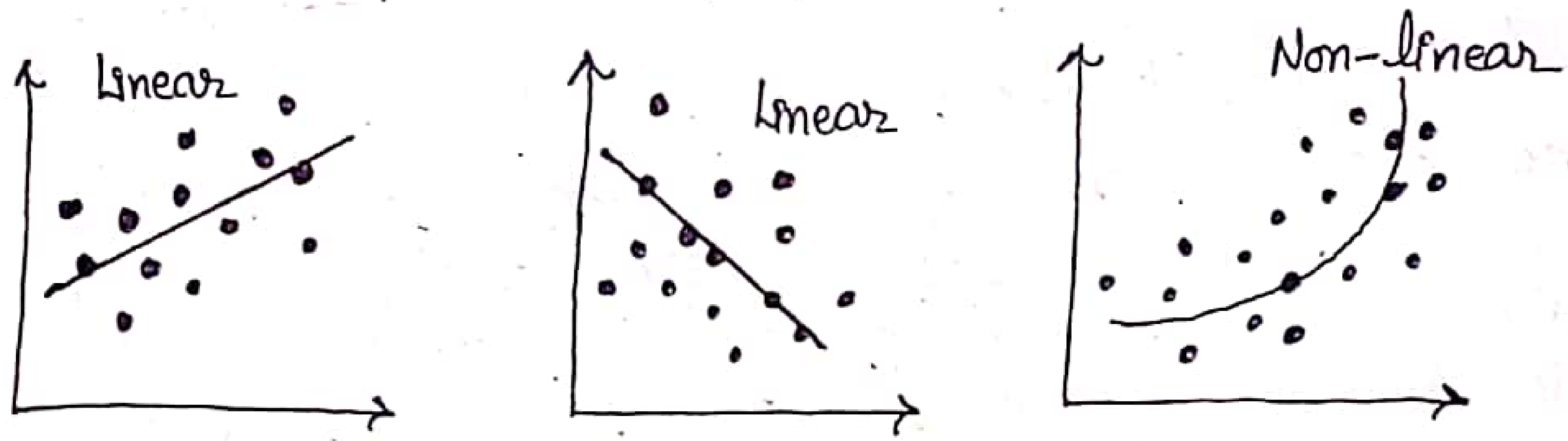

fig: Linear and Non-Linear Regression

Interpolation vs. Regression:

In interpolation we are given with some data points, and we are supposed to find a curve which fits the input/output relationship perfectly. In case of interpolation, we don't have to worry about variance of the fitted curve.

When we do regression, we look for a function that minimizes some cost, usually sum of squares of errors. We don't require the function to have the exact values at given points; we just want a good approximation.

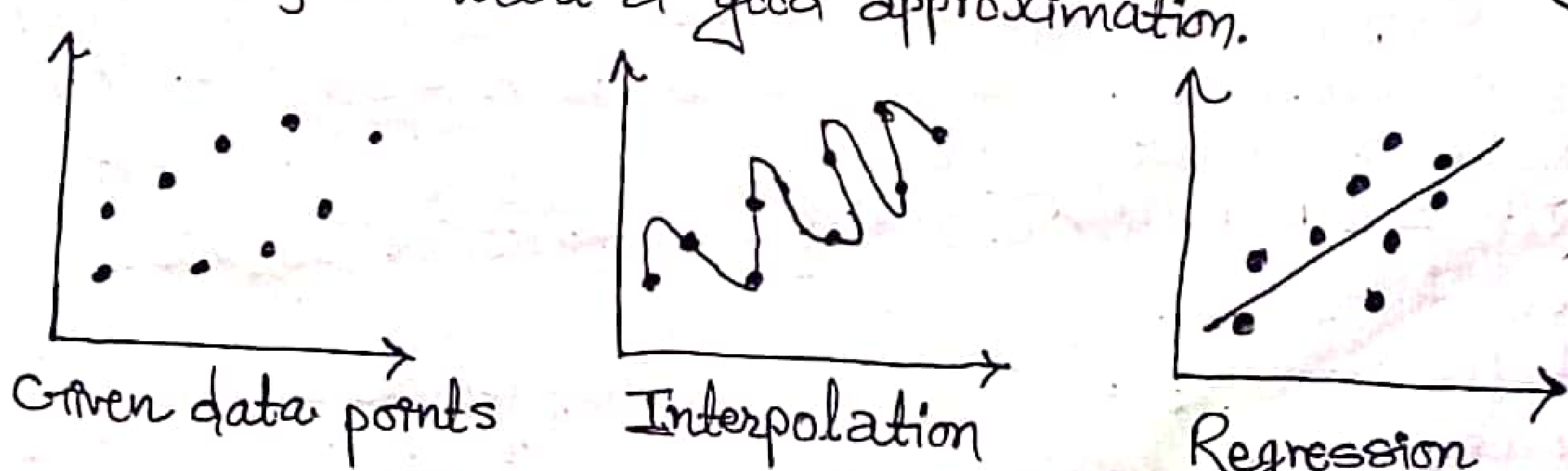

fig. Interpolation vs. Regression

Scanned with CamScanner

⊛. Parameter Estimation Methods

It is the process of estimating values of parameters of the model, based on the observed pairs of values and minimizing certain objective function. The mostly used method of parameter estimation is as follows:

→ Minimize the sum of square of errors (least square regression)
i.e, minimize $\sum e_i^2 = \sum (y_i - f(x_i))^2$, $i = 1, 2, \ldots, n$.

⊛. Least Squares Method:

The least squares method is a form of mathematical regression analysis used to determine the line of best fit for a set of data, providing a visual demonstration of the relationship between the data points.

In this method we need to minimize following objective function or error function.

$$\sum e_i^2 = \sum (y_i - f(x_i))^2, \; i = 1, 2, \ldots, n.$$

Applications:

i) It is a statistical procedure to find the best fit for a set of data points by minimizing the sum of the residuals of the points from the plotted curve.

ii) It is used to predict the behaviour of dependent variables.

1) Linear Regression:

Fitting a straight line is simplest approach of regression analysis, which is called linear regression. A straight line can be represented by using the mathematical equation $y = f(x) = a + bx$, where a and b are regression coefficients to be determined.

Let the sum of squares of individual errors can be expressed as:

$$E = \sum e_i^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

We should choose regression coefficient a and b such that E is minimum. Necessary condition for E to be minimum is:

$$\frac{\partial E}{\partial a} = 0 \quad \text{and} \quad \frac{\partial E}{\partial b} = 0.$$

Scanned with CamScanner

$$\frac{\partial E}{\partial a} = 2\sum_{i=1}^{n}(y_i - a - bx_i)(-1) = 0$$

(partial derivative on E i.e. $\sum_{i=1}^{n}(y_i - a - bx_i)^2$

∂a गर्दा a भएको term मा मात्रै operation हुन्छ, अरु उस्तै, $\frac{\partial -a}{\partial a} = -1$ $\boxed{\text{power आगाडि लेखिन्छ}}$

$$\& \quad \frac{\partial E}{\partial b} = 2\sum_{i=1}^{n}(y_i - a - bx_i)(-x_i) = 0$$

b को term मा मात्रै operation हुन्छ

$$\frac{\partial \cdot (y_i - a - bx_i)}{\partial b} = \frac{\partial -bx_i}{\partial b} = -x_i$$

Simplifying above equations, we get,

$$-\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} a + \sum_{i=1}^{n} bx_i = 0$$

and $$-\sum_{i=1}^{n} y_i x_i + \sum_{i=1}^{n} ax_i + \sum_{i=1}^{n} bx_i^2 = 0.$$

Since, $$\sum_{i=1}^{n} a = a + a + \dots + a = na$$

Above equations can be written as;

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \quad \text{——— (i)} \qquad \left(\because \sum_{i=1}^{n} a = na\right)$$

and $$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \quad \text{——— (ii)}$$

Solving above equations (i) and (ii) gives

$$b = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$a = \frac{\sum_{i=1}^{n} y_i}{n} - b\frac{\sum_{i=1}^{n} x_i}{n} = \bar{y} - b\bar{x}$$

a र b को value याद गर्नु question solve गर्दा काम लाग्छ।

where, $$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \text{ and } \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Scanned with CamScanner

Algorithm:

1. Start
2. Read number of points, say n.
3. Read given data points, say x[i] and y[i].
4. Find summations of x, y, xy and x^2 as below;

   for i=0 to n-1

   sx=sx+x[i]

   sy=sy+y[i]

   sxy=sxy+x[i] *y[i],

   sx2=sx2+x[i] *x[i]

   End for
5. Calculate values of parameters as below:-

   b=((n*sxy)-(sx*sy))/((n*sx2)-(sx*sx))

   and a=(sy/n)-(b*sx/n).
6. Display the equation ax+b
7. Terminate.

Example: The values of y and their corresponding values of y are shown in the table below;

x	0	1	2	3	4
Y=f(x)	2	3	5	4	6

a) Find the least square regression line y=ax+b.

b) Estimate the value of y when x=10.

Solution:

x_i	y_i	x_i^2	x_iy_i
0	2	0	0
1	3	1	3
2	5	4	10
3	4	9	12
4	6	16	24
$\sum x_i=10$	$\sum y_i=20$	$\sum x_i^2=30$	$\sum x_iy_i=49$

Scanned with CamScanner

Now, $b = \dfrac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} = \dfrac{5\times49 - 10\times20}{5\times30 - 10^2} = \dfrac{45}{50} = 0.9$

$$a = \frac{\sum_{i=1}^{n} y_i}{n} - b\frac{\sum_{i=1}^{n} x_i}{n} = \frac{20}{5} - 0.9\times\frac{10}{5} = 4 - 1.8 = 2.2$$

Thus the least square regression line that best fits through given data points is;

$$y = 0.9x + 2.2 \quad \text{--- (i)}$$

Now putting $x = 10$ in eqn (i) we get

$$y = 0.9\times10 + 2.2$$
$$= 11.2$$

2) Non-Linear Regression:

It is a form of regression analysis in which observational data are modeled by a function. It can be described in two ways;

a) By Fitting Exponential Model:

An exponential model is described by the equation: $y = ae^{bx}$. In this equation the coefficients a and b are the constants and given as;

$$a = \frac{\sum_{i=1}^{n} y_i \cdot e^{bx_i}}{\sum_{i=1}^{n} e^{2bx_i}}$$

& $f(b) = \sum_{i=1}^{n} y_i x_i e^{bx_i} - \dfrac{\sum_{i=1}^{n} y_i e^{bx_i}}{\sum_{i=1}^{n} e^{2bx_i}} \cdot \sum_{i=1}^{n} x_i \cdot e^{2bx_i} = 0.$

Example: Below is the given relative intensity of radiation as a function of time.

x (time)	0	1	3	5	7	9
y (intensity of radiation)	1.0	0.891	0.708	0.562	0.447	0.355

Scanned with CamScanner

If the level of the relative intensity of radiation is related to time via an exponential formula $y = ae^{bx}$, find the value of a and b.

Solution:-

The values of a and b are evaluated by solving following nonlinear equations;

$$a = \frac{\sum_{i=1}^{n} y_i e^{bx_i}}{\sum_{i=1}^{n} e^{2bx_i}} \quad \text{---(i)}$$

$$\& \; f(b) = \sum_{i=1}^{n} y_i x_i e^{bx_i} - \frac{\sum_{i=1}^{n} y_i e^{bx_i}}{\sum_{i=1}^{n} e^{2bx_i}} \sum_{i=1}^{n} x_i e^{2bx_i} = 0 \quad \text{---(ii)}$$

Eq^n (ii) can be solved for b using bisection method. To estimate the initial guesses, we assume $b = -0.120$ and $b = -0.110$. We need to check whether these values first bracket the root of $f(b) = 0$.

i	x_i	y_i	$y_i x_i e^{bx_i}$	$y_i e^{bx_i}$	e^{2bx_i}	$x_i e^{2bx_i}$
1	0	1	0.00000	1.00000	1.00000	0.00000
2	1	0.891	0.79205	0.79205	0.78663	0.78663
3	3	0.708	1.4819	0.49395	0.48675	1.4603
4	5	0.562	1.5422	0.30843	0.30119	1.5060
5	7	0.447	1.3508	0.19297	0.18637	1.3046
6	9	0.355	1.0850	0.12056	0.11533	1.0379
$n=6$			$\Sigma = 6.2501$	$\Sigma = 2.9062$	$\Sigma = 2.8763$	$\Sigma = 6.0954$

Now,

$$f(-0.120) = (6.2501) - \frac{2.9062}{2.8763}(6.0954)$$

$$= 0.091357$$

Similarly

$$f(-0.110) = -0.10090$$

Since $f(-0.120) \times f(-0.110) < 0$, the value falls in the bracket of $[-0.120, -0.110]$. The next guess of root then is;

Scanned with CamScanner

$$b = \frac{-0.120 + (-0.110)}{2}$$
$$= -0.115$$

Continuing with the bisection method, the root of $f(\lambda) = 0$ is found as $b = \lambda = -0.11508$. ~~Form Equation(~~ From eqⁿ (i), a can be calculated as:

$$a = \frac{\sum_{i=1}^{6} y_i e^{bx_i}}{\sum_{i=1}^{6} e^{2bx_i}} = \frac{2.9373}{2.9378} = 0.99983$$

Thus the regression formula is given by,

$$y = 0.99983\, e^{-0.11508x}.$$

b) Fitting Exponential Model by Linearization:

We know that, exponential model is given by $y = ae^{bx}$ —— (i)

Taking natural log on both the sides gives.

$$\log y = \log(ae^{bx})$$
$$\Rightarrow \log y = \log a + bx \longrightarrow (ii)$$

This equation is similar to the form of linear equation $y = a + bx$. Thus we can evaluate parameters a and b by using the equation of linear regression model as below:

$$b = \frac{n\sum_{i=1}^{n} x_i \log y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} \log y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \quad \text{—— (iii)}$$

And
$$\log a = \frac{\sum_{i=1}^{n} \log y_i}{n} - b\frac{\sum_{i=1}^{n} x_i}{n} \quad \text{—— (iv)}$$

Let,
$$R = \frac{\sum_{i=1}^{n} \log y_i}{n} - b\frac{\sum_{i=1}^{n} x_i}{n}$$

Now eqⁿ (iv) becomes $\log a = R$

Taking antilog on both sides we get, $a = e^R$ —— (v)

Now, we can calculate values of regression coefficients by using equations (iii) and (v).

Example: Fit the curve $y = ae^{bx}$ through the data given below:

Solution:

x	-4	-2	0	1	2	4
$y=f(x)$	0.57	1.32	4.12	6.65	11	30.3

We know that exponential model is given by. $y = ae^{bx}$

$$\Rightarrow \log y = \log a + bx.$$

This equation is similar in form to the linear equation $y = a + bx$. Thus we can evaluate parameters a and b by using the equation of linear regression model as below:

$$b = \frac{n\sum_{i=1}^{n} x_i \log y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} \log y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

And $$\log a = \frac{\sum_{i=1}^{n} \log y_i}{n} - b\frac{\sum_{i=1}^{n} x_i}{n}$$

Now calculate required summations as below:

i	x_i	y_i	$\log y_i$	$x_i \log y_i$	x_i^2
1	-4	0.57	-0.562	2.248	14
2	-2	1.32	0.277	-0.555	4
3	0	4.12	1.415	0	0
4	1	6.65	1.894	1.894	1
5	2	11.0	2.394	4.796	4
6	4	30.3	3.411	13.645	16
$n=6$	$\sum x_i = 1$		$\sum = 8.835$	$\sum = 22.03$	$\sum = 41$

Now,

$$b = \frac{6 \times 22.03 - 1 \times 0.835}{6 \times 41 - 1} = \frac{123.345}{245} = 0.503$$

$$\log a = \frac{8.835}{6} - 0.503 \times \frac{1}{6} = 1.472 - 0.084 = 1.388$$

Since $\log a = 1.388$

$$\Rightarrow a = e^{1.388} = 4.006$$

Thus the regression formula then is $y = 4.006\, e^{0.503x}$

Scanned with CamScanner

Fitting Polynomial Models:

It is a form of regression analysis in which the relationship between the independent x and the dependent variable y is modeled as an nth degree polynomial in x.

Let n data points are $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and we want to fit an mth order polynomial through the given data points.
General form of polynomial of degree m is given below;

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m, \quad m < n \quad \text{--- (i)}$$

Now, the residual at each data point is given by,

$$e_i = y_i - a_0 - a_1 x_i - \ldots - a_m x_i^m$$

The sum of the square of the residuals is given by,

$$E = \sum_{i=1}^{n} e_i^2$$

$$= \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - \ldots - a_m x_i^m\right)^2$$

To find the constants of the polynomial regression model, we equate the derivatives respect to a_p to zero and setting those equations in matrix form gives;

$$\begin{bmatrix} n & \left(\sum_{i=1}^{n} x_i\right) & \cdots & \left(\sum_{i=1}^{n} x_i^m\right) \\ \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) & \cdots & \left(\sum_{i=1}^{n} x_i^{m+1}\right) \\ \cdots & \cdots & \cdots & \cdots \\ \left(\sum_{i=1}^{n} x_i^m\right) & \left(\sum_{i=1}^{n} x_i^{m+1}\right) & \cdots & \left(\sum_{i=1}^{n} x_i^{2m}\right) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \vdots \\ \sum_{i=1}^{n} x_i^m \cdot y_i \end{bmatrix}$$

The above equation can be solved for $a_0, a_1, \ldots, a_m$.

Example: Fit the quadratic curve through the following data and estimate value of y at $x = 12$.

X	1	3	4	5	6	7	8	9	10
f(x)	2	7	8	10	11	11	10	9	8

Scanned with CamScanner

Solution:

General form of quadratic equation is $y = a_0 + a_1 x + a_2 x^2$. The coefficients a_0, a_1, a_2 of the quadratic equation can be found as ~~form~~ from following matrices;

$$\begin{bmatrix} n & \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) \\ \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) \\ \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) & \left(\sum_{i=1}^{n} x_i^4\right) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i^2 y_i \end{bmatrix}$$

Now calculate required summations as below:

i	x	y	x^2	x^3	x^4	$x \times y$	$x^2 \times y$
1	1	2	1	1	1	2	2
2	3	7	9	27	81	21	63
3	4	8	16	64	256	32	128
4	5	10	25	125	625	50	250
5	6	11	36	216	1296	66	396
6	7	11	49	343	2401	77	539
7	8	10	64	512	4096	80	640
8	9	9	81	729	6561	81	729
9	10	8	100	1000	10000	80	800
	$\Sigma = 53$	$\Sigma = 76$	$\Sigma = 381$	$\Sigma = 3017$	$\Sigma = 25317$	$\Sigma = 489$	$\Sigma = 3547$

Thus we have following matrix;

$$\begin{bmatrix} 9 & 53 & 381 \\ 53 & 381 & 3017 \\ 381 & 3017 & 25317 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 76 \\ 489 \\ 3547 \end{bmatrix}$$

Solving the above system of simultaneous linear equations, we get

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -1.46 \\ 3.605 \\ -0.268 \end{bmatrix}$$

Thus the quadratic curve that passes through given data points is $y = -1.46 + 3.605x - 0.268x^2$.

Now put $x = 12$ in above equation, we get

$y = -1.46 + 3.605 \times 12 - 0.268 \times 12^2 = 3.208$.

Scanned with CamScanner