Lecture 22: Newton's difference polynomials, Inverse Interpolation

(12-Spt-2012)

POLYNOMIAL APPROX... (Contd..)

LECTURE - 22
12 - SPT - 2012

Yesterday, we discussed on :

* Divided differences and corresponding tables
* Divided difference polynomial
* Forward, Backward, and centered differences
* Newton's forward-difference polynomial

Newton's forward-difference polynomial

For a data set of $(n+1)$ points, we can fit an unique n^{th} degree polynomial $P_n(x)$

$$P_n(x) = f_0 + s\,\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_0 + \cdots$$

$$\cdots + \frac{s(s-1)(s-2)\cdots(s-(n-1))}{n!}\Delta^n f_0$$

where $s = \dfrac{x - x_0}{\Delta x}$ or $x = x_0 + s\,\Delta x$

Example

For the same data set of time versus distance, consider the first six data points

time t_i	Distance $f_i(t)$	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
0.0	0.0			
		50.0		
10.0	50.0		50.0	
		100.0		0.0
20.0	150.0		~~100.0~~ 50.0	
		150.0		0.0
30.0	300.0		50.0	
		200.0		0.0
40.0	500.0		50.0	
		250.0		
50.0	750.0			

The unique polynomial that can fit the given data is a $P_5(x)$ and using Newton's forward difference mechanism:

$$P_5(t) = f_0 + s\,\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_0 + \frac{s(s-1)(s-2)(s-3)}{4!}\Delta^4 f_0 + \frac{s(s-1)(s-2)(s-3)(s-4)}{5!}\Delta^5 f_0$$

where $s = \dfrac{t - t_0}{\Delta t}$ or $t = t_0 + s\Delta t$

As obvious from difference table from $\Delta^3 f_i$, the differences are zero.

$$\therefore\ P_5(t) = 0.0 + 50s + 25\,s(s-1)$$

At any time $t = 14$ seconds,

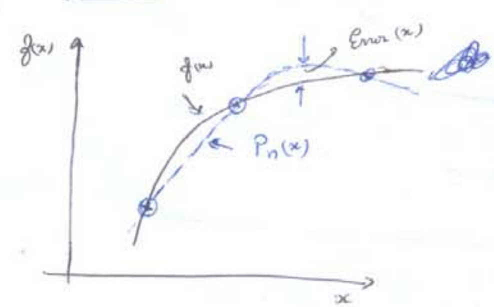
$$P(14) = 50\,s + 25\,s(s-1)\ ,\quad s = \frac{t - t_0}{\Delta t} = \frac{14 - 00}{10} = 1.4$$

$$\therefore\ P(14) = 50 \times 1.4 + 25 \times 1.4 \times 0.4$$

$$= 84$$

The distance moved $=$ $\underline{84\ m}$
(as evaluated by interpolation)

Error Analysis for Newton Forward-Difference Polynomial



If there are $(n+1)$ data points, then $P_n(x)$ is the unique polynomial that can pass through all these $(n+1)$ data points.

In the given data

x_i	f_i
x_0	f_0
x_1	f_1
x_2	f_2
$\vdots$	$\vdots$
x_n	f_n

The error term Error(x) is ~~defined~~ described as below.

* As the polynomial $P_n(x)$ passes through all the data points, there are no errors at those locations.
* The error will be only between any two data points.

$\therefore$ ~~Error (x) =~~ From Taylor's series, we have seen that

$$Error = P_n(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x - x_0)^{n+1} ; \quad x_0 \le \xi \le x$$

(Refer Lecture 20, Page 3)

* As stated earlier, the error in function evaluation is there only between any two data points.

$$\therefore \quad Error(x) = \frac{1}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n) f^{(n+1)}(\xi) ; \quad x_0 \le \xi \le x_n$$

So now in Newton's forward difference polynomial : $\rightarrow$

$x - x_0 = s\Delta x$ or $x = x_0 + s\Delta x$

$$\text{Ily} \quad x - x_1 = (x_0 + s\Delta x) - x_1 = s\Delta x - (x_1 - x_0) = \Delta x(s-1)$$

$$x - x_2 = (x_0 + s\Delta x) - x_2 = \Delta x(s-2)$$

$$(x - x_n) = (s-n)\,\Delta x$$

$$\therefore \text{Error}(x) = \frac{1}{(n+1)!}\, s(s-1)(s-2)\cdots(s-n)\,\Delta x^{n+1} f^{(n+1)}(\xi)$$

Note that $P_n(x)$, the possible term after the n^{th} order should be

$$\Rightarrow \frac{s(s-1)(s-2)\cdots(s-n)}{(n+1)!}\,\Delta^{n+1} f_0$$

$\Rightarrow$ So this quantity should be the error.

However one cannot evaluate $\Delta^{n+1} f_0$ using the available data.

$$\therefore \text{We suggest} \quad \Delta^{(n+1)} f_0 \approx \Delta x^{n+1} f^{(n+1)}(\xi)$$

Newton Backward - Difference Polynomial

As you have seen, the forward - difference formula can be applied in the beginning of the data set or middle. At the bottom, you need to go for backward differences.

∇f_n.

$\rightarrow$ The unique n^{th} degree polynomial for $(n+1)$ data points

$$P_n(x) = f_0 + s\nabla f_0 + \frac{s(s+1)}{2!}\nabla^2 f_0 + \cdots + \frac{s(s+1)(s+2)\cdots(s+(n-1))}{n!}\nabla^n f_0$$

where 's' is interpolating variable.

$$s = \frac{x - x_0}{\Delta x}$$

In our example on time vs. distance data, you may do Backward differences ∇f

If we want to evaluate at $t = 35$ s, the distance.
then put $t_0 = 50$ s

$$\cancel{s = \frac{x - x_0}{\Delta x} = \frac{35 - 0}{10} = 3.5}$$

$P_3(x)$ Co $\quad s = \frac{t - t_0}{\Delta t} = \frac{35 - 50}{10} = -1.5$

$$P(t) = f(t=50) + s\,\nabla f_{(50)} + \frac{s(s+1)}{2!}\nabla^2 f + \frac{s(s+1)(s+2)}{3!}\nabla^3 f$$

$$= 750 + (-1.5) \times 250 + \frac{(-1.5)(0.5)}{2} \times 50 + 0$$

$$= \underline{356.25 \text{ m}}$$

Note: If we fit a second degree polynomial $P_2(t)$ with the data point required $t = 35$ s $\rightarrow t = 40$ s as t_0.

Then. $\quad s = \frac{t - t_0}{\Delta t} = \frac{35 - 40}{10} = -0.5$

$$P(t) = f(t=40) + s\,\nabla f + \frac{s(s+1)}{2!}\nabla^2 f$$

$$= 500 + (-0.5) \times 200 + \frac{(-0.5)(0.5)}{2} \times 50$$

$$= \underline{\underline{393.75 \text{ m}}}$$

Inverse Interpolation

In interpolation what we were doing was that for any inter-mediate value of 'x' that lies between the discrete data point, the corresponding functional values are evaluated.

x_i	y_i
x_0	y_0
x_1	y_1
$\vdots$	$\vdots$
x_n	y_n

Q: What happens if some values of function is given and you want to identify the corresponding x value.

S: You require inverse interpolation.

Say for the three data points and you are given $y = 80.0$, find t.

i	t_i	y_i
0	10.0	50.0
1	20.0	150.0
2	30.0	300.0

100 50
150

You can form a unique second degree polynomial which is

$$t \approx P_2(y)$$

$$P_2(y) = \frac{(y-y_1)(y-y_2)}{(y_0-y_1)(y_0-y_2)} t_0 + \frac{(y-y_0)(y-y_2)}{(y_1-y_0)(y_1-y_2)} t_1 + \frac{(y-y_0)(y-y_1)}{(y_2-y_0)(y_2-y_1)} t_2$$

(Lagrange Method)

You can use Newton's forward difference polynomial

$$P_2(t) = f_0 + s\,\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0 \quad ; \quad s = \frac{t-t_0}{\Delta t}$$

$$80 = 50 + s \times 100 + \frac{s(s-1)}{2} \times 50$$

$$\therefore \quad t = 10 + s\Delta t$$
$$= 10 + 10s$$
$$t = 10(1+s)$$

i.e. $$\frac{80}{50} = 1 + 2s + \frac{s^2 - s}{2}$$

Form quadratic eqn. in 's' and evaluate t.

Multi-variate Approximation

If we have $z = f(x,y)$

or $z = f(x_1, x_2)$, etc.

This means that z depends on two variables

It is multi-variate function.

∴ For any given tabular data of multi-variate function we can fit approximate functions and use them for
→ interpolation, differentiation, integration, etc.

Here again, there are
→ Exact fit
→ Approximate fit.

In the exact fit
- ★ Successive univariate polynomial approximation
- ★ Direct multi-variate polynomial approximation.