Polynomial Interpolation (II): Lagrange and Newton Divided Differences

Wednesday, October 9, 2024 12:05 PM

Class 19: October 9, 2024

Recall: Last class, we defined the so-called "interpolation problem": Given data-points (x0,y0), (x1,y1), …, (xn,yn) , we want to find a function f(x) such that f(x_i) = y_i for i=0,1,…,n+1. That is, we want to find a functional model that "interpolates the data". This is then useful to perform a number of tasks using f(x): evaluate at other values of x, compute derivatives, compute integrals, solve differential equations …

We then mentioned that the problem "interpolate this data with a nice function" is not well defined; it has many solutions. We then explored the problem of interpolating using *a polynomial of degree <= n". We showed that, **given n+1 points (for which the x_i's are distinct), there is a unique polynomial of degree <= n that interpolates the data.**

The way we showed this was by writing p(x) = a0 + a1 x + … + an x^n, and then writing a system of n+1 linear equations p(x_i) = y_i. These can be written as V a = y, where V is the Vandermonde matrix, a is a vector of unknowns, and y is a vector of data. I mentioned that we can derive (using induction) that

Det(V) = Product_{i > j} (x_i - x_j)

Which is not 0 if and only if the x_i's are distinct. However, we then said that the algorithm " a = solve(V,y)" is not stable, since **kappa(V) grows exponentially with n**.

We then defined a special polynomial basis, known as the Lagrange polynomial basis. This basis of Pn, made of {L0(x),L1(x),…,Ln(x)}, is found by asking the conditions necessary so that the interpolant is

p(x) = y0 L0(x) + y1 L1(x) + … + yn Ln(x)

In other words, so that (a0,a1,…,an) = (y0,y1,…,yn). The conditions we need to impose are that **Li(xj) = 1 if i=j**, and 0 otherwise.

Today, we will continue our discussion of Lagrange, and introduce another method known as Newton interpolation.

<ins>3 points</ins>: L_0, L_1, L_2

$$L_0(x_0) = 1, \quad L_0(x_1) = 0, \quad L_0(x_2) = 0$$

$$L_0(x) = c\,(x-x_1)(x-x_2) \longrightarrow \overset{1}{\overset{\|}{L_0(x_0)}} = c\,(x_0-x_1)(x_0-x_2)$$

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$$

n+1 points

$$\boxed{L_i(x) = \frac{\prod_{j\neq i}(x-x_j)}{\prod_{j\neq i}(x_i-x_j)}}$$

Give me (x_i, y_i)

$$\rightarrow \underline{p(x) = y_0 L_0(x) + y_1 L_1(x) + \dots + y_n L_n(x)}$$

def lagrange_interp$(\underbrace{\vec{x}}_{n+1}, \underbrace{\vec{y}}_{n+1}, \underbrace{\vec{x}_{EVAL}}_{m})$:

- [ ] Evaluate $L_0(x_{EVAL}), L_1(x_{EVAL}), \dots, L_n(x_{EVAL})$
- [ ] Compute $p(x_{\dots})$

- [x] Evaluate $L_0(x_{EVAL}), L_1(x_{EVAL}), \dots, L_n(x_{EVAL})$
- [x] Compute $p(x_{EVAL})$.

return $p(x_{EVAL})$

$$\underset{m\times(n+1)}{\mathbb{L}}\vec{y} = \underset{m\times(n+1)}{\begin{bmatrix} | & | & & | \\ L_0(x_{EVAL}) & L_1(x_{EVAL}) & \cdots & L_n(x_{EVAL}) \\ | & | & & | \end{bmatrix}} \underset{(n+1)\times 1}{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}} = \underline{p(x_{EVAL})}$$

Evaluation $\underline{O(mn)}$

Forming $\mathbb{L} \to \underline{O(mn^2)}$

Interpolation Error

- $\{(x_i, y_i)\}_{i=0}^{n}$, $y_i = f(x_i)$ Exact data.
- Assume f is smooth ($n+1$ derivatives cont)

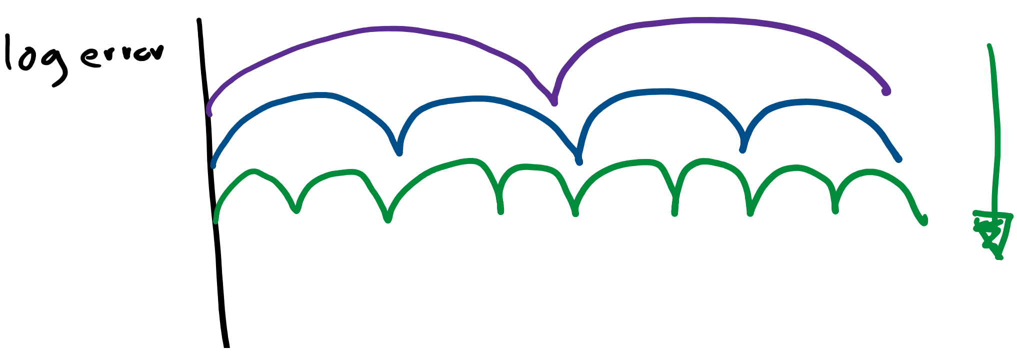
Error:

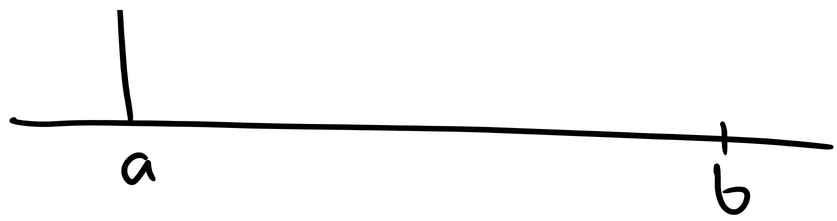
$$f(x) - p(x) = \boxed{\frac{f^{(n+1)}(\xi_x)}{(n+1)!}} (x-x_0)(x-x_1)\cdots(x-x_n)$$

Depends on f

Only dependent on the points

Equispaced pts are the best...





Two other ways — Newton polymials
— Barycentric Lagrange.

Newton

$$\underline{p(x) = c_0 \nu_0(x) + c_1 \nu_1(x) + \dots + \overset{c_n}{\nu_n}(x)}$$

$\nu_0(x) = 1 \longrightarrow x_0$

$\nu_1(x) = (x - x_0) \longrightarrow x_1, x_0$

$$y_0 = p(x_0) = c_0 \cdot 1 + \cancel{c_1}(x_0 - x_0) = c_0$$

$$y_1 = p(x_1) = \underbrace{c_0}\cdot 1 + c_1(x_1 - x_0)$$

$$y_1 = y_0 + c_1(x_1 - x_0)$$

$$c_0 = f(x_0) \;/\; c_1 = \frac{y_1 - y_0}{x_1 - x_0} = \underbrace{\frac{f(x_1) - f(x_0)}{x_1 - x_0}}_{\text{divided diffevence}}$$

$$p(x) = f(x_0)\,1 + \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0}\right)(x - x_0) + \underline{c_2 \nu_2(x)}$$

$\nu_2(x) = (x - x_0)(x - x_1),$

$$p(x_2) = y_2 \;\; (f(x_2)) \longrightarrow c_2 = \frac{\left(\frac{f(x_2) - f(x_1)}{x_2 - x_1}\right) - \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0}\right)}{x_2 - x_0}$$

$$p(x_2) = y_2 \quad (f(x_2)) \longrightarrow C_2 = \frac{\quad\quad}{x_2 - x_0}$$

"second divided diff."

$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

Find $C_0, C_1, C_2, \ldots, C_n$

x_0	$f(x_0)$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$\leftarrow \vec{c}$
x_1	$f(x_1)$	$f[x_1, x_2]$	$\vdots$	
x_2	$f(x_2)$	$\vdots$	$\vdots$	
$\vdots$	$\vdots$	$\vdots$		
x_n	$f(x_n)$			

<u>Notation</u>: $f[x_i] = f(x_i)$

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

$$\vdots$$

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