Polynomial Interpolation (III): Newton and Barycentric Lagrange

Friday, October 11, 2024 12:50 PM

Class 20: October 11, 2024

Recall: Last time, we finished our discussion of the Lagrange interpolation method, focusing on how we would implement a function that takes data (xi,yi) and evaluation points x_eval, forms a matrix L = L_j(x_eval(i)), and evaluates p(x_eval).

We then discussed the main theorem we have for polynomial interpolation error. This theorem is very close to that for the Taylor residual theorem. It says that if the data y_i is equal to f(x_i) for f a (n+1) times continuously differentiable function on an interval [a,b] that contains the interpolation nodes x_i, then

$$f(x) - p(x) = [\, f^{(n+1)}(\eta_x) / (n+1)! \,] * [(x-x_0)(x-x_1)\ldots(x-x_n)]$$

We discussed how this error is the product of two terms: the first *only depends on the function*, while the second *only depends on the interpolation nodes*. In lab, you explored how a bad set of interpolation nodes (e.g. equispaced nodes) can sometimes produce issues with error that blows up near the end-points of our interval.

Finally, we introduced a different way to find and evaluate our polynomial interpolant: Newton polynomial interpolation. Today, we will continue our discussion of Newton and then we will "fix" Lagrange so it is much more stable and fast.

0 order diff $\rightarrow f[x_0] = f(x_0)$

1 order diff $\rightarrow \left(f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right)$

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

2 order diff: $f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$

$\vdots$

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

EXAMPLE:

$(16 - 81)/1$

x	0	1	3	4	6
f(x)	81	16	0	1	81
1ST ORDER	-65	-8	1	40	
2ND ORDER	19	3	13		
3RD ORDER	-4	2			
4TH "	1				

Compute coeffs $O(n^2)$

$$p(x) = 81 - 65(x-0) + 19x(x-1) - 4x(x-1)(x-3) + 1 \cdot x(x-1)(x-3)(x-4)$$

$+ 1 \cdot x(x-1)(x-3)(x-4)$

Eval at x_{eval}:

HORNER ALGORITHM:

$$p(x) = 81 + x\left[-65 + (x-1)\left[19 + (x-3)\left[-4 + 1(x-4)\right]\right]\right]$$

Given x_{eval}, c_i:

```
Px = Cn;
for j from n-1 to 0
    Px = Cj + (x - xj) Px;
return Px.
```

$O(mn)$ work

$$p(x) = 3 + 2x - 4x^2 + 5x^3 - 10x^4$$
$$= 3 + x\left[2 + x\left[-4 + x\left[5 + x(-10)\right]\right]\right]$$

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots,$$

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & - \\ 1 & (x-x_0) & 0 & \cdots & - \\ 1 & (x_2-x_0) & (x_2-x_0)(x_2-x_1) & 0 \cdots & \\ \vdots & \vdots & & \ddots & \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \end{bmatrix}$$

Barycentric Lagrange

$(x_i, y_i)_{i=1}^{n+1}$ data, $\dfrac{L_j(x)}{\prod_{i \neq j}(x - x_i)}$

$p(x) = \sum^{n}$ u_i $\dfrac{\quad}{\quad}$

$$p(x) = \sum_{j=0}^{n} y_j \frac{\prod_{i \neq j} (x - x_i)}{\prod_{i \neq j} (x_j - x_i)}$$

$$\omega_j = \frac{1}{\prod_{i \neq j} (x_j - x_i)}, \quad \psi(x) = \prod_{i=0}^{n} (x - x_i)$$

$$\boxed{p(x) = \sum_{j=0}^{n} y_j \frac{\omega_j}{x - x_j} \psi(x)}$$

$$\boxed{p(x) = \psi(x) \sum_{j=0}^{n} y_j \frac{\omega_j}{x - x_j}}$$

1ST BARYCENTRIC FORMULA.

- Precompute $\psi(x)$ and ω_j's and they only depend $x_0, x_1, \ldots, x_n$.
- Given y_i, compute $p(x)$.
- For certain pts (equisp., chebyshev) ↳ Formulas for ω_j

Trick: Interpolate 1:

$$1 = \psi(x) \sum_{j=0}^{n} \frac{\omega_j}{x - x_j}$$

$$\boxed{\frac{p(x)}{1} = \frac{\psi(x) \sum_{j=0}^{n} y_j \frac{\omega_j}{x - x_j}}{\psi(x) \sum_{j=0}^{n} \frac{\omega_j}{x - x_j}}}$$

2nd Bary. Formula.

$$\cancel{\psi(x)} \sum_{j=0}^{n} \frac{\omega_j}{x - x_j}$$

Formula.

② $\omega_j = c\,\tilde{\omega}_j$ → Simplify (relatively)
→ make weights smaller.