Polynomial Interpolation IV: Hermite Interpolation

Monday, October 14, 2024 9:56 AM

Class 21: October 14, 2024

Recall: Last class we discussed two implementations of polynomial interpolation given n+1 data points (xi,yi): **Newton interpolation** and **Barycentric Lagrange Interpolation**. Remember that given the uniqueness of the polynomial interpolant, these are two different algorithms to compute a representation of and evaluate the ***same polynomial***. We usually assume yi = f(xi), for f a smooth function.

(1) Newton interpolation: This method uses the *Newton basis*, defined incrementally as nu_0(x) = 1, nu_1(x) = (x-x0), nu_2(x) = (x-x0)(x-x1), and so on. Given data (xi,yi), let's say we want to find the coefficients for the interpolant and then evaluate it in an array x_eval of m points. This involves two tasks:

1. We build the (n+1) x (n+1) Newton tableau by taking differences of the previous columns up until the diagonal. This involves O(n^2) operations (2 for each element of the table).
2. We then use Horner's method to evaluate p(x_eval) = c0 + (x-x0)[c1 + (x-x1)[…]] in around 2mn operations.

A nice feature of Newton is that we can add a new point, update one diagonal of the Newton tableau and then re-evaluate p(x_eval) efficiently (without having to re-do it all again).

(2) Barycentric Lagrange: This method uses the *Lagrange basis*, but it re-writes the interpolant

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

To make evaluating it way more stable and fast (computationally). It uses the "second barycentric formula", which takes the form:

p(x) = (Sum yj wj / (x - xj)) / (Sum wj / (x-xj))

Where wj = 1/ Prod_{i not equal to j} (xi - xj)

1. To evaluate this expression, we need to first pre-compute w0,w1,…,wn (or find a formula for them, e.g. for equispace, Chebyshev). If we do not have a formula, this is O(n^2) work.
2. Then, all we need to evaluate is wj / (x_eval - xj) and compute the expression above, which is O(mn) operations.

Note that, once again, we can add a new point, update the weights and then update the numerator and denominator of the barycentric formula cheaply (without having to do it all again).

After a demo in class, we will wrap up this subject and start a new one: How to use derivative information to do polynomial interpolation (Hermite interpolation).

HERMITE INTERPOLATION

So far data $(x_i, y_i) \rightarrow y_i = f(x_i)$ EXACT

- f is smooth.

Data $(x_i, y_i, z_i)_{i=0}^{n} \rightarrow y_i = f(x_i)$
$z_i = f'(x_i)$
(f smooth)

n+1 data pts → n+1 d.o.fs → poly deg ≤ n.

Now: 2(n+1) data → 2(n+1) dofs → poly deg ≤ 2n+1

Hermite interpolant → unique poly h(x) of deg ≤ 2n+1 such that:

of deg $\le 2n+1$ such that.

2n+2 eqs. in 2n+2 vars

$$\left\{\begin{array}{l} h(x_i) = f(x_i) = y_i \\ h'(x_i) = f'(x_i) = z_i \end{array}\right\}$$

First thing: $h(x) = d_0 + d_1 x + d_2 x^2 + \ldots + d_{2n+1} x^{2n+1}$

$$\rightarrow \underbrace{\mathbb{V}_c \cdot}_{\text{"Confluent Vandermonde"}} \vec{d} = \begin{bmatrix} \vec{y} \\ \vec{z} \end{bmatrix}$$

$$\det(\mathbb{V}_c) = \prod_{i \neq j} (x_i - x_j)^2 \neq 0$$

$$K(\mathbb{V}_c) \rightarrow \text{horrible } (K(\mathbb{V})^2)$$

Basis $\{H_j\}$, $\{K_j\}$ $\rightarrow$ poly deg $\le 2n+1$

Interpolant:

$$h(x) = \sum_{j=0}^{n} y_j H_j(x) + z_j K_j(x)$$

$$h(x_i) = \sum y_j \underbrace{H_j(x_i)}_{\delta_{ij}} + z_j \underbrace{K_j(x_i)}_{0}$$

$$h'(x_i) = \sum y_j \underbrace{H_j'(x_i)}_{0} + z_j \underbrace{K_j'(x_i)}_{\delta_{ij}}$$

$$H_j(x) = \left(1 - 2n_j(x - x_j)\right)\underbrace{\left[L_j(x)\right]^2}_{M_j(x)}$$

where $n_j = L_j'(x_j)$;

$$K_j(x) = (x - x_j)\left[L_j(x)\right]^2$$

$\underline{M_j(x_i) = \delta_{ij}}$ $\quad$ $\underline{M_j'(x_i) = 2n_j\,\delta_{ij}}$

<u>Hermite-Newton</u>:

$\mu_0(x) = 1$, $\mu_1(x) = (x - x_0)$, $\mu_2(x) = (x - x_0)^2$

$\mu_3(x) = (x - x_0)^2(x - x_1)$, $\mu_4(x) = (x - x_0)^2(x - x_1)^2, \dots$

---

x_0	$f(x_0)$	$f'(x_0)$	$f[x_0,x_0,x_1]$	$\dots$
x_0	$f(x_0)$	$f[x_0,x_1]$	$f[x_0,x_1,x_1]$	
x_1	$f(x_1)$	$f'(x_1)$	$\vdots$	
x_1	$f(x_1)$	$f[x_1,x_2]$		
x_2	$f(x_2)$	$f'(x_2)$		
x_2	$f(x_2)$			

<u>HORNER</u>: $\quad c_0 + (x - x_0)\left[c_1 + (x - x_0)\left[c_2 + (x - x_1)\left[\dots\right.\right.\right.$

BARYCENTRIC LAG.-HERMITE.