

$$\begin{bmatrix} 1 & x_0 & \dots & x_0^N \\ 1 & x_1 & \dots & x_1^N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \dots & x_N^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_N \end{bmatrix}$$

Vandermonde
matrix V

Vandermonde determinant: $\text{Det}(V) = \prod_{i=0}^N \prod_{j=i+1}^N (x_i - x_j)$

Thus, for distinct interpolation points $x_i \neq x_j, i \neq j$, $\text{Det}(V) \neq 0$ and the Vandermonde matrix is nonsingular \rightarrow unique solutions exist

Theorem

There exists a unique interpolating polynomial $P_N(x)$ iff the data points x_0, x_1, \dots, x_N are distinct.

However, the Vandermonde matrix is extremely ~~the~~ ill-conditioned, to the point that it is very difficult to use in practice.

Reason - the basis functions are monomials $x^k, k=0, \dots, N$ which for large k look alike; in analogy with vectors in \mathbb{R}^N , they are almost colinear (linearly dependent).

* Lagrange Interpolating Polynomials

Choose the interpolating polynomial in a special form:

$$P_n(x) = \sum_{k=0}^n \phi_k(x) y_k, \text{ where}$$

$$\textcircled{*} \phi_k(x) = \frac{\prod_{\substack{i=0, \dots, n \\ i \neq k}} (x - x_i)}{\prod_{\substack{i=0, \dots, n \\ i \neq k}} (x_k - x_i)} \quad \text{Lagrange / cardinal polynomial}$$

(The terms $i=k$ are omitted both in the numerator and denominator)

~~For~~ For example (with $n=3$)

$$P_3(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} y_1 \\ + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} y_3$$

Remarks

- The interpolating polynomial is a sum of $n+1$ n -degree polynomials (Lagrange functions)

- The Lagrange functions have the property

$$\phi_k(x_i) = \delta_{ik}, \quad i, k = 0, \dots, n,$$

i.e., have unit value at the corresponding node and zero at all other nodes

Thanks to this property, the interpolation conditions

$$P_n(x_i) = y_i, \quad i = 0, \dots, n$$

are satisfied automatically (by construction).

— the interpolating polynomial is easy to construct (no need to solve an algebraic system)

— computational cost:

$$\left\{ \begin{array}{l} O(n) \text{ operations to evaluate} \\ \phi_i(x) \text{ for each } x \end{array} \right\} \times n = O(n^2)$$

Can we do better than that?

Barycentric Interpolation Formula

(Chapter 5 in Trefethen)

For all k , the numerators in $(*)$ are the same, except that they miss different factors $(x - x_k)$. One should take advantage of that

Define the node polynomial for the given grid

$$\Omega(x) = \prod_{i=0}^n (x - x_i)$$

Then, the Lagrange polynomials become

$$\phi_k(x) = \frac{\Omega(x)}{\Omega'(x_k)(x - x_k)}$$

Examples:

- Lagrange poly. n
- Lagrange interpolated n .

We can also define

$$\lambda_k = \frac{1}{\prod_{\substack{i=0 \\ i \neq k}}^n (x_k - x_i)} = \frac{1}{L'(x_k)}$$

so that we have $\phi_k = L(x) \frac{\lambda_k}{x - x_k}$, $k=0, \dots, n$

So, the Lagrange interpolation formula becomes

$$p_n(x) = L(x) \sum_{k=0}^n \frac{\lambda_k}{x - x_k} y_k$$

The First Form of the Barycentric Interpolation Formula

Remarks

- single dependence on x in side the sum
- if the weights λ_k are known, the formula produces a value for $p(x)$ in just $O(n)$ operations
- evaluation of the weights λ_k requires $O(n^2)$ computation, but ~~this~~ this is independent of x and hence can be performed just once at the beginning (for special grids the weights are known analytically)

- The Lagrangian formula is numerically stable (w.r.t round-off errors)
- Further modifications exist, e.g., for Chebyshev interpolation.

Analysis of Error of Polynomial Interpolation

Assume the data is obtained using the function $f(x)$ as $y_i = f(x_i)$, $i = 0, \dots, n$

Consider the error function: $E(x) = f(x) - p_n(x)$

It vanishes at x_0, x_1, \dots, x_n , thus

$$E(x) = f(x) - p_n(x) = (x-x_0)(x-x_1)\dots(x-x_n)g(x)$$

$g(x)$ accounts for the behavior between the nodes
Then

$$f(x) - p_n(x) - E(x) = f(x) - p_n(x) - (x-x_0)(x-x_1)\dots(x-x_n)g(x) = 0$$

Need to determine $g(x)$. For this purpose we will define a new function $w(t)$ (depending on a new variable t)

$$w(t) = f(t) - p_n(t) - (t-x_0)(t-x_1)\dots(t-x_n)g(t)$$

$w(t)$ has $n+2$ roots: $t = x_0, x_1, \dots, x_n$, and $t = x$.

Assume $w(t)$ is continuous & differentiable

Mean Value Theorem - There is a root of the derivative $w'(t)$ between every two roots of $w(t)$; thus, altogether, there are $n+1$ roots of the derivative

$W''(t) - n$ roots

$W'''(t) - (n-1)$ roots

⋮

$W^{(n+1)}(t) - 1$ root (denoted ξ) in the interval
bounded by $\{x_0, x_n, x\}$

Thus,

$$W^{(n+1)}(\xi) = 0 = \frac{d^{(n+1)}}{dt^{(n+1)}} \left[f(t) - p_n(t) - (t-x_0)(t-x_1)\dots(t-x_n) \right] \Big|_{t=\xi}$$

$$= f^{(n+1)}(\xi) - 0 - (n+1)! g(x) = 0 \Rightarrow$$

$$\Rightarrow g(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}, \text{ where } \xi \text{ varies between } \{x_0, x_1, x\}$$

Check your

$$E(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^n (x-x_k)$$

Remarks

* Since $f(x)$ is usually unknown, the expression for $E(x)$ has only qualitative meaning

* if $f(x) = p_m(x)$, $m \leq n$, then $E(x) \equiv 0$

* Smoothly functions $f(x) \Rightarrow$ smaller error

* For large n the error $E(x)$ increases at the endpoints of the interval (Runge phenomenon);
do not take n larger than 5, 6, ...

Another Look at the Errors of Polynomial Interpolation

$$E_n(x) = \frac{g^{(n+1)}(\xi)}{(n+1)!} \underbrace{\prod_{k=0}^n (x-x_k)}_{\omega_{n+1}(x)}$$

Assume $x \in [a, b]$

Den $x_k = a + kh$, $k = 0, \dots, n$, $h = \frac{b-a}{n}$

Let $M_n = \max_{a \leq x \leq b} |g^{(n)}(x)|$ - upper bound on the n -th derivative

$$C_n = \max_{0 \leq z \leq n} \prod_{k=0}^n |z-k|$$

$$\text{Thus: } E_{\max} = \max_{a \leq x \leq b} |E_n(x)| \leq \frac{1}{(n+1)!} C_n M_{n+1} h^{n+1}$$

For a fixed degree of interpolating polynomial

$$E_{\max} \leq C h^{n+1}, \quad h \ll 1$$

How do reduce the error: $h \rightarrow 0$

- shrink the interval $b-a \rightarrow 0$, while keeping n unchanged

- keep the interval, but increase n

then we need to control the derivatives $\lim_{n \rightarrow \infty} M_n$

The rate of decay (increase) of M_n depends on the analytical (regularity) properties of the interpolated function

* Runge's function (1901)

(42)

$$g(x) = \frac{1}{1+25x^2}, \quad x \in [-1, 1]$$