April 9, 2020

Numerical Analysis

Last time: Constructive Formula for polynomial interpolation:

Lagrange Interpolating Formula.

For data $(x_0,y_0), (x_1,y_1), \dots, (x_n,y_n)$, the polynomial interpolant is a degree n polynomial that passes through the points.

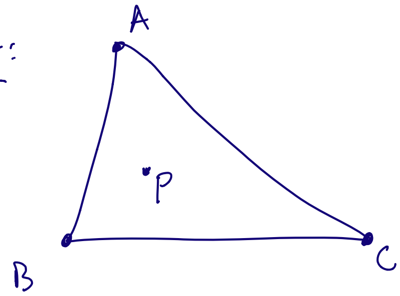
In some cases, the Lagrange form of the polynomial interpolant can be numerically unstable: has a large condition number, large round-off error, etc.

Barycentric Form(s) of Interpolation

The numerical stability of evaluating an interpolating polynomial can be fixed by rearranging its terms – this does not change what the actual interpolant is.

As motivation: Examine the barycentric coordinates on a triangle.

Ex:



The barycentric coordinates of a point P inside a triangle with vertices A, B, C are given by:

$$P = \alpha A + \beta B + \gamma C \qquad (\alpha, \beta, \gamma) \text{ coordinates}$$

with $\alpha + \beta + \gamma = 1$, $\alpha \geq 0$, $\beta \geq 0$, $\gamma \geq 0$.

The center of mass of the triangle is the given by

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}.$$

Idea: Replace A, B, C with functions that sum to 1.

Start with the Lagrange Form: (and rewrite)

$$p_n(x) = \sum_{k=0}^{n} \underbrace{\left(\prod_{j \neq k} \frac{(x - x_j)}{(x_k - x_j)} \right)}_{L_k(x).} y_k$$

$$= \sum_{k=0}^{n} \underbrace{\left(\prod_{j=0}^{n} (x - x_j) \right)}_{\text{does not depend on } k} \frac{1}{x - x_k} \left(\prod_{j \neq k} \frac{1}{(x_k - x_j)} \right) y_k$$

$$= \underbrace{\left(\prod_{j=0}^{n} (x - x_j) \right)}_{\varphi(x)} \sum_{k=0} \frac{1}{x - x_k} \underbrace{\left(\prod_{j \neq k} \frac{1}{x_k - x_j} \right)}_{w_k} y_k$$

$$= \varphi(x) \sum_{k=0}^{n} \frac{w_k}{x - x_k} y_k$$

(Modified Lagrange Form)

First Barycentric Formula

We can even further simplify this form by "dividing by 1".
The polynomial interpolant of the function 1 at the same nodes x_j is simply:

$$\underline{1} = \varphi(x) \sum_{k=0}^{n} \frac{w_k}{x - x_k} \qquad (\text{since } y_k = 1).$$

Then $p_n(x) = \dfrac{\cancel{\varphi(x)} \sum_{k=0} \frac{w_k}{x - x_k} y_k}{\cancel{\varphi(x)} \sum \frac{w_k}{x - x_k}} = \dfrac{\sum_{k=0}^{n} w_k/_{x - x_k} \cdot y_k}{\sum_{k=0}^{n} w_k/_{x - x_k}}$

Second Barycentric Formula.

This form is "stable for any reasonable choice of x_j" (2004, Higham).

One should always use this form to do polynomial interpolation.

Convergence of Polynomial Interpolation

Let's examine the question of what happens as $n \to \infty$, i.e.

$$\lim_{n\to\infty} \underbrace{\max_x |f(x) - p_n(x)|}_{\text{this is the } \infty\text{-norm.}} = ?$$

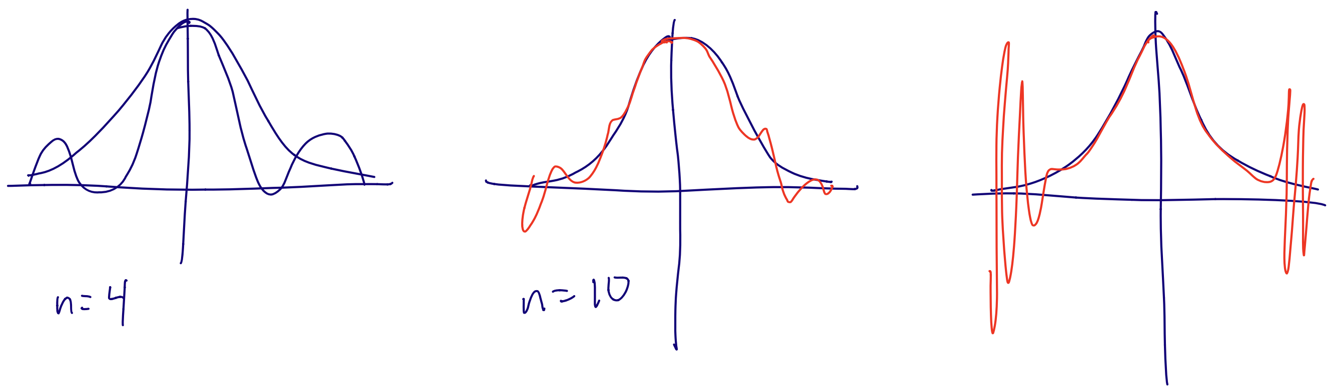
The pointwise error is approximately:

$$\max_{\xi} \frac{|f^{(n+1)}(\xi)|}{(n+1)!} \cdot \max_x \prod_{j=0}^{n} |x - x_j|$$

It's not obvious if this increases or decreases as $n \to \infty$...

[see Matlab demo for interpolation of Runge's Function $f(x) = \dfrac{1}{1+(3x)^2}$

The Runge Effect



This behavior is related to the fact that the function $f(x) = \dfrac{1}{1+x^2}$ has a singularity at $x = \pm i$ in the complex plane. $f(i) = \dfrac{1}{1+i\cdot i} = \dfrac{1}{1+-1} = \dfrac{1}{0} = \infty$.

This dictates the radius of convergence of its Taylor series:

$$f(x) = 1 - x^2 + x^4 - x^6 + x^8 - x^{10} + \ldots$$

(can be fixed, we'll see later on)

Function approximation

Polynomial interpolation mainly has applications in function approximation, with respect to some norm:

For functions, some example norms are:

$$\|f\|_\infty = \max_{x\in[a,b]} |f(x)|$$

$$\|f\|_2 = \sqrt{\int_a^b |f(x)|^2\,dx}$$

$$\|f\|_1 = \int_a^b |f(x)|\,dx$$

Just like for n-dimensional vectors.

Norms of functions satisfy the same properties as those in the finite dimensional vector case:

1. $\|f\| \geq 0$, $\|f\| = 0$ iff $f = 0$
2. $\|cf\| = |c|\,\|f\|$
3. $\|f+g\| \leq \|f\| + \|g\|$

Ex: The 2-norm of a function can be generalized by introducing a "weight" function $w>0$:

$$\|f\|_{2,w} = \sqrt{\int_a^b |f(x)|^2\, w(x)\,dx}$$

So: the polynomial p_n of degree n that best approximates a function f in the ∞-norm is

$$\min_{p_n \in P_n} \underbrace{\|p_n - f\|_\infty}_{\text{maximum pointwise error.}}$$

Do not think of p_n as a polynomial interpolant of f.

From analysis class, we know that continuous functions f on some finite interval can be approximated arbitrarily well by a polynomial of "some" degree: this result is known as the Weierstrass Approximation Theorem.

I.e. For any $\epsilon > 0$, there exists a polynomial p such that $\| f - p \|_\infty < \epsilon$, $\| f - p\|_2 < \infty$.

Unfortunately, this is a completely useless theorem for numerical approximation.
It doesn't tell you how to find p!

The question of restricting $p \in P_n$ is much more interesting, and actually useful.

To pose the problem:

For $n > 0$, find $p_n \in P_n$ such that

$$\| f - p_n \|_\infty = \min_{q \in P_n} \| f - q \|_\infty .$$

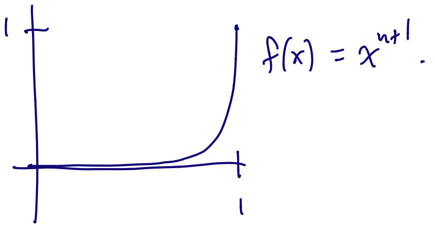
Theorem: Such a p_n exists, and is unique.

(The proof does not tell us how to find p_n.)

In general, one cannot write down the minimax polynomial, i.e. the polynomial p_n such that

$$\| f - p_n \|_\infty = \min_{q \in P_n} \max_{x \in [a,b]} | f(x) - q(x) |$$

However, we can explicitly write down then minimax polynomial approximation to the monomial $f(x) = x^{n+1}$ on $[0,1]$



Theorem Let $n \geq 0$, then $\|p_n - f\|_\infty$, with $f(x) = x^{n+1}$, is minimized when

$$p_n(x) = \underbrace{x^{n+1} - \frac{1}{2^n}\cos\left((n+1)\,\operatorname{acos} x\right)}_{\text{polynomial of degree } n}.$$

The function $T_n(x) = \cos(n \operatorname{acos} x)$ is known as the Chebyshev polynomial of degree n. These functions play a very important role in numerical analysis.