Overview

Interpolation

revis

Recap: Interpolation

Starting point: Looking for a linear combination of functions φ_i to hit given data points (x_i, y_i).

Interpolation becomes solving the linear system:

$$y_i = f(x_i) = \sum_{j=0}^{N_{\text{func}}} \alpha_j \underbrace{\varphi_j(x_i)}_{V_{ij}} \qquad \leftrightarrow \qquad V\boldsymbol{\alpha} = \boldsymbol{y}.$$

Want unique answer: Pick $N_{\text{func}} = N \to V$ square.

V is called the (generalized) Vandermonde matrix.

Main lesson:

$< \varepsilon$

$$V\,(\text{coefficients}) = (\text{values at nodes})\,.$$

Rethinking Interpolation

We have so far always used monomials $(1, x, x^2, x^3, \ldots)$ and equispaced points for interpolation. It turns out that this has *significant problems*.

Demo: Monomial interpolation

Demo: Choice of Nodes for Polynomial Interpolation

Interpolation: Choosing Basis Function and Nodes

Both function basis and point set are under our control. What do we pick?

Ideas for basis functions:

- Monomials $1, x, x^2, x^3, x^4, \ldots$
- Functions that make $V = I \to$ 'Lagrange basis'
- Functions that make V triangular $\to$ 'Newton basis'
- Splines (piecewise polynomials)
- Orthogonal polynomials
- Sines and cosines
- 'Bumps' ('Radial Basis Functions')

Ideas for nodes:

- Equispaced
- 'Edge-Clustered' (so-called Chebyshev/Gauss/... nodes)

Better Conditioning: Orthogonal Polynomials

What caused monomials to have a terribly conditioned Vandermonde?

What's a way to make sure two vectors are *not* like that?

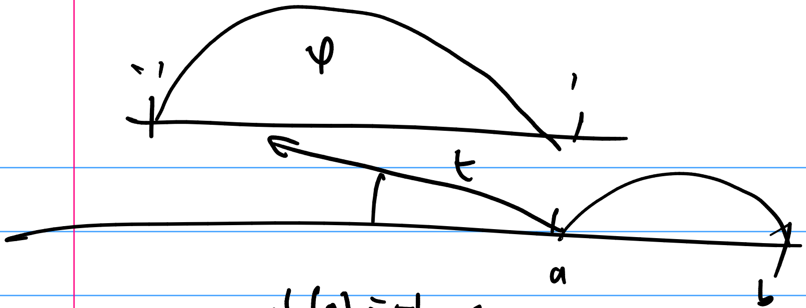
orthogonal

But polynomials are functions!

$$\cancel{(f,g) \text{ " = " } \sum_{i=1} f(x_i)\cdot g(x_i)}$$

$$\longrightarrow \quad (f,g) = \int_{-1}^{1} f(x)\,g(x)\,dx$$

$$(f, g) = \int_{-1}^{1} f(x)\, g(x) \frac{1}{\sqrt{1-x^2}}\, dx$$



$t(a) = -1 \leftarrow$

$t(b) = 1 \leftarrow$

$t(x) = mx + n$

$m = \frac{2}{b-a}$

Better Conditioning: Orthogonal Polynomials (II)

But how can I practically compute the Legendre polynomials?

Another Family of Orthogonal Polynomials: Chebyshev

Three equivalent definitions:

- Result of Gram-Schmidt with weight $1/\sqrt{1-x^2}$

> What is that weight?

T_0, T_1, T_2, T_3

- $T_k(x) = \cos(k \cos^{-1}(x))$
- $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$ recurrence relation

Demo: Chebyshev interpolation part I

> What are good nodes to use with Chebyshev polynomials?

T_0 T_1

$$T_k = \cos\left(k \cdot \cos^{-1}(x_i)\right)$$

$$x_i = \cos(\ldots)$$

Chebyshev Nodes

Might also consider zeros (instead of roots) of T_k:

$$x_i = \cos\left(\frac{2i+1}{2k}\pi\right) \quad (i = 1\ldots,k).$$

The Vandermonde for these (with T_k) can be applied in $O(N \log N)$ time, too.

It turns out that we were still looking for a good set of interpolation nodes.

We came up with the criterion that the nodes should bunch towards the ends. Do these do that?

Demo: Chebyshev interpolation part II

Calculus on Interpolants

Suppose we have an interpolant $\tilde{f}(x)$ with $f(x_i) = \tilde{f}(x_i)$ for $i = 1, \ldots, n$:

$$\tilde{f}(x) = \alpha_1 \varphi_1(x) + \cdots + \alpha_n \varphi_n(x)$$

How do we compute the derivative of $\tilde{f}$?

Suppose we have function values at nodes $(x_i, f(x_i))$ for $i = 1, \ldots, n$ for a function f. If we want $f'(x_i)$, what can we do?

About Differentiation Matrices

How could you find coefficients of the derivative?

Give a matrix that finds the second derivative.