(10) **Function Spaces**

What is interpolation?

Finding a function through given x/y points.

given

x	y
x_1	y_1
x_2	y_2
x_m	y_m
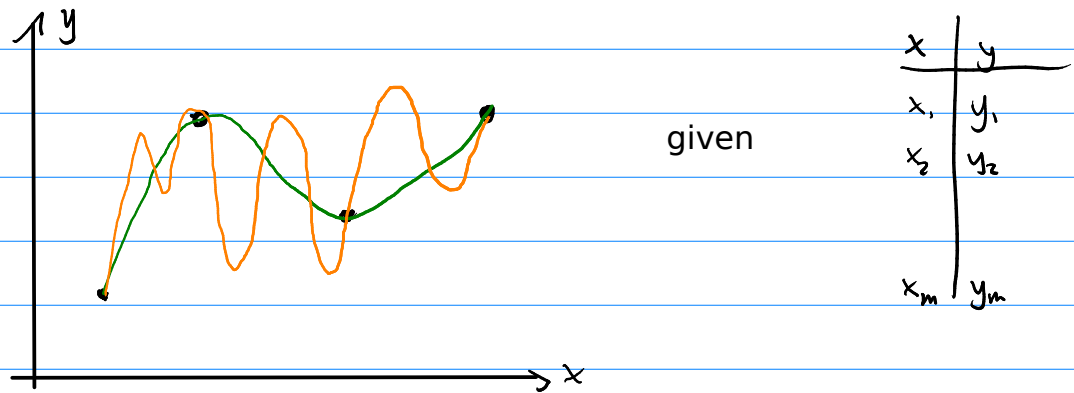
But there are lots of possible functions through those points. Be more precise.

To make interpolation a well-specified problem, we require that the resulting function (the "interpolant") is a linear combination of a given basis of functions.

Given: $\varphi_1(x), \varphi_2(x) \cdots, \varphi_m(x)$

#functions matches #points

Want: $f(x) = \alpha_1 \varphi_1(x) + \cdots + \alpha_m \varphi_m(x)$

so that $f(x_i) = y_i$

By solving a linear system:

$$\alpha_1 \varphi_1(x_1) + \cdots + \alpha_m \varphi_m(x_1) = y_1$$

$$\vdots$$

$$\alpha_1 \varphi_1(x_m) + \cdots + \alpha_m \varphi_m(x_m) = y_n$$

$$\rightsquigarrow \quad \underbrace{\begin{pmatrix} \varphi_1(x_1) & \cdots & \varphi_m(x_1) \\ & \vdots & \\ \varphi_1(x_m) & \cdots & \varphi_m(x_m) \end{pmatrix}}_{V} \vec{\alpha} = \vec{y}$$

V is called the generalized Vandermonde matrix.

The general idea is

$$V \underbrace{\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}}_{\substack{\text{basis} \\ \text{coefficients}}} = \underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}}_{\substack{\text{values} \\ \text{at interp.} \\ \text{points}}} \quad \rightsquigarrow \quad V\vec{\alpha} = \vec{y}$$

Suppose we've fixed m points (also called 'interpolation nodes'):

$$X_1 \quad \cdots \quad X_m$$

We then fix a basis. In this case, we will use the monomials:

$$x^0, x^1, x^2, x^3, x^4, x^5, \ldots, \boxed{x^{m-1}}$$

number of basis functions
must match number of nodes

So we are looking for the coefficients α in

$$\alpha_0 1 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_{m-1} x^{m-1}$$

Set up the Vandermonde matrix:

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^4 \\ 1 & x_2 & & & \\ 1 & x_3 & & & \\ 1 & x_4 & & & \\ 1 & x_5 & \cdots & & x_5^4 \end{bmatrix}$$

The term "Vandermonde matrix" first appeared for monomials, so this is actually *the* Vandermonde matrix. We've earlier 'generalized' it by allowing different sets of functions.

All this should look relatively familiar: It's exactly what we've been doing in data fitting--except the matrices are square.

$$\begin{array}{c|c} x_i & y_i \\ \hline 0 & 7 \\ 2 & 4 \\ 3 & 4 \end{array}$$

$$\begin{array}{ccc} 1 & x & x^2 \end{array}$$

$$V = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{pmatrix}$$

$$V\vec{\alpha} = \begin{pmatrix} 7 \\ 4 \\ 4 \end{pmatrix}$$

Solve

$$\vec{\alpha} = \begin{pmatrix} 7 \\ -2.5 \\ 0.5 \end{pmatrix}$$

$$f(x) = \alpha_0 1 + \alpha_1 x + \alpha_2 x^2$$

Demo: Interpolation with generalized Vandermonde matrices

Where is interpolation helpful?

There are lots of things you can do if you have a function
that you cannot do if you only have point values.

Examples:

- Evaluate anywhere, not just at the given data points

- Find rate of change depending on x

- More generally, use calculus

Unless you're very lucky, no, that will not typically the case.

But under some assumptions, we can say when the interpolant

will be "close", and in what sense.

Demo: Interpolation Error (Part 1)

Needed assumptions:

–   Function f being interpolated is "smooth" (has many derivatives)

–   Length h of the interpolation interval is "sufficiently small"

In this case:

h: length of the interpolation interval

$$|f(x) - \tilde{f}(x)| \le C \cdot h^{n+1}$$ for all x in the interval

"true"
function

interpolant

C: an unknown constant (depends on f and n, but not h)

n: highest polynomial degree in interpolant

The main purpose of estimates like this is to predict

dependency of the error on the length of the interval h.

So what types of predictions can I make using the interpolation error estimate?

Suppose I have an interpolation error E for some interval length h

for interpolation with a quadratic function.

What would happen if I used h/2 instead of h?

$$\text{Error}(h) \approx C \cdot h^3$$

$$\text{Error}\left(\tfrac{h}{2}\right) \approx C \cdot \left(\tfrac{h}{2}\right)^3 = \tfrac{1}{8} \cdot C \cdot h^3 \approx \tfrac{1}{8} \cdot \text{Error}(h)$$

That's a concrete (testable) prediction!

Demo: Interpolation Error, part 2

Generally, if a method has an error estimate of the form

$$\text{Error} \leq C \cdot h^p$$

it is called p-th order convergent.

So interpolation with polynomials of degree n is (n+1)-th order convergent.

Shorter intervals (smaller h) seem to decrease the error. Why is that?

The function will vary less on a shorter interval.

That means it is easier to "capture" the behavior of the function using

a line/parabola/cubic/...

Have:

- Choice of basis
- Choice of nodes

Let's look at choice of nodes first, then at choice of basis.

What is a "good" set of nodes for polynomial interpolation?

Demo: Choice of interpolation nodes

Observation: Best if nodes cluster towards interval ends

"Best" set of interpolation nodes on [-1,1]: "Chebyshev nodes"

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right) \qquad k = 1 \ldots n$$

(1) Monomial basis $\quad 1, x_1, x_1^2, \ldots, x^{m-1}$

Already discussed above

Demo: Monomial Interpolation

Observation: Works fine up to m=6 or so. Beyond that,
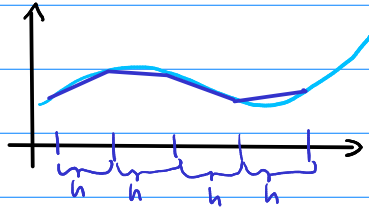the conditioning of the Vandermonde matrix starts to hurt.
Using a better set of nodes helps a little, but not enough
for arbitrarily large n.
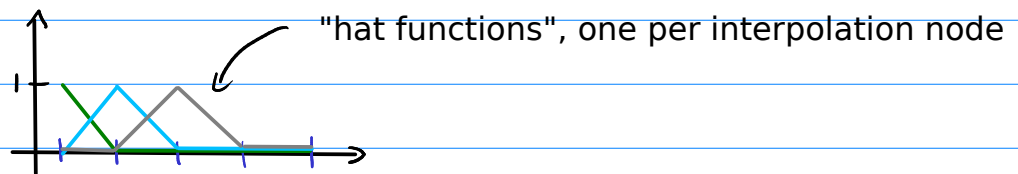
So here's an idea: Use multiple lower-degree pieces.

(2) Piecewise polynomials

For example: Piecewise linear, continuous

What would a basis for this interpolation scheme look like?

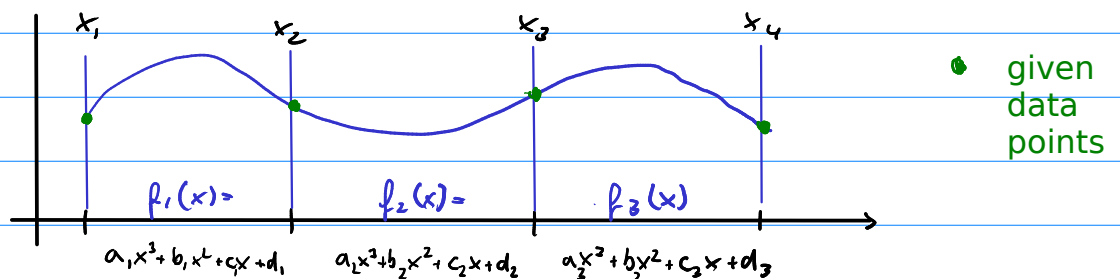"hat functions", one per interpolation node

Problem: Need very small h (lots of intervals) to be accurate.

Idea: Use higher-degree polynomial on each piece.

This results in a so-called "spline".

With polynomials of degree 3, you would get a "cubic spline".

x_1 x_2 x_3 x_4

$f_1(x) =$ $f_2(x) =$ $f_3(x)$

$a_1 x^3 + b_1 x^2 + c_1 x + d_1$ $a_2 x^3 + b_2 x^2 + c_2 x + d_2$ $a_3 x^3 + b_3 x^2 + c_3 x + d_3$

given data points

Need 4 coefficients per piece. -> More coefficients than data points.

Need to impose extra conditions to get a unique answer.   → HW

(3) Sines and cosines ("Fourier basis")

What's the basis? $\sin(0x)$, $\cos(0x)$, $\sin(1x)$, $\cos(1x)$, $\sin(2x)$, $\cos(2x)$, ...

$\to 0$

What to use as points?   equispaced on $[0, 2\pi]$... but:

$2\pi \frac{0}{5}$, $2\pi \frac{1}{5}$, $2\pi \frac{2}{5}$, $2\pi \frac{3}{5}$, $2\pi \frac{4}{5}$, $\boxed{2\pi \frac{5}{5}}$

no — $\cos(n \cdot 2\pi) = \cos(0)$
$\sin(n \cdot 2\pi) = \sin(0)$

So leave out the last point.

Write down a Vandermonde matrix for up to n=2:

$$
\begin{array}{ccccc}
\cos(0 \cdot 0) & \sin(1 \cdot 0) & \cos(1 \cdot 0) & \sin(2 \cdot 0) & \cos(2 \cdot 0) \\
\cos\left(0 \cdot \frac{2\pi}{5}\right) & \sin\left(1 \cdot \frac{2\pi}{5}\right) & \cos\left(1 \cdot \frac{2\pi}{5}\right) & \sin\left(2 \cdot \frac{2\pi}{5}\right) & \cos\left(2 \cdot \frac{2\pi}{5}\right) \\
\cos\left(0 \cdot \frac{2\pi \cdot 2}{5}\right) & \sin\left(1 \cdot \frac{2\pi \cdot 2}{5}\right) & \cos\left(1 \cdot \frac{2\pi \cdot 2}{5}\right) & \sin(2 \cdot \quad) & \cos(2 \cdot \quad) \\
\cos\left(0 \cdot \frac{2\pi \cdot 3}{5}\right) & \sin(1 \cdot \quad) & \cos(1 \cdot \quad) & \sin(2 \cdot \quad) & \cos(2 \cdot \quad) \\
\cos\left(0 \cdot \frac{2\pi \cdot 4}{5}\right) & \sin(1 \cdot \quad) & \cos(1 \cdot \quad) & \sin(2 \cdot \quad) & \cos(2 \cdot \quad)
\end{array}
$$

Demo: Fourier Interpolation

Observations:

- works best for periodic functions
- generalized Vandermonde has orthogonal columns

An important motivation for the Fourier basis is that its coefficients correspond to frequencies.
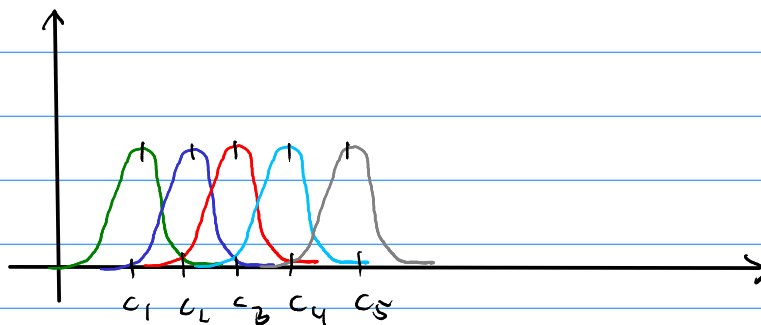
Demo: Audio synthesis with sines

Demo: Audio experiments

(4) Radial basis functions

Idea: Use multiple copies of the same function, centered at a number of locations on the real line.

$$\varphi_1(x) = \varphi\left(x - c_1\right)$$
$$\varphi_2(x) = \varphi\left(x - c_2\right)$$
$$\vdots$$
$$\varphi_s(x) = \varphi\left(x - c_J\right)$$

The "mother function" φ obeys

$$\bullet \;\; \varphi(x) = \varphi\left(|x|\right)$$

φ only depends on "radius" $|x|$
(=distance from origin)

$\longrightarrow$ "radial" basis function

$$\bullet \;\; \varphi(x) \to 0 \;\; \text{as} \;\; |x| \to \infty$$

Demo: Interpolation with Radial Basis Functions

⊕   Easy to construct   ⊖   Can be tricky to stabilize

Quality of result depends on
choice of radius, centers

Easily generalizes
to multiple dimensions

Interpolant decays to zero
away from nodes

So how would I use calculus on an interpolant?

$$\hat{f}(x_i) = \tilde{\hat{f}}(x_i) \quad i=1\ldots n$$

<u>Have</u>: interpolant $\quad \hat{f}(x) = \alpha_1 \varphi_1(x) + \cdots + \alpha_n \varphi_n(x)$

<u>Want</u>: derivative $\quad \tilde{\hat{f}}'(x) = \alpha_1 \varphi_1'(x) + \cdots + \alpha_n \varphi_n'(x)$

Easy because interpolation basis (φ_i) is known!

<u>Have</u>: function values at nodes $\quad (x_i) \qquad f(x_i)$

<u>Want</u>: values of derivative at nodes $\quad f'(x_i) \quad \leftarrow \quad$ hard to get

$$\tilde{f}'(x_i) \quad \leftarrow \quad \text{easy to get!}$$

$\quad$ How?

①   Compute coefficients $\quad \vec{\alpha} = V^{-1} \vec{f} \qquad \vec{f} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$

②   Build generalized Vandermonde with <u>derivatives</u> of basis

$$V' = \begin{pmatrix} \varphi_1'(x_1) & \cdots & \varphi_n'(x_1) \\ & \vdots & \\ \varphi_1'(x_n) & \cdots & \varphi_n'(x_n) \end{pmatrix}$$

③   Compute $V'\vec{\alpha} = \begin{pmatrix} \alpha_1 \varphi_1'(x_1) + \cdots + \alpha_n \varphi_n'(x_1) \\ \vdots \end{pmatrix} = \begin{pmatrix} \tilde{\hat{f}}'(x_1) \\ \vdots \\ \hat{f}'(x_n) \end{pmatrix}$

$$\underbrace{\phantom{\begin{pmatrix} \tilde{\hat{f}}'(x_1) \\ \hat{f}'(x_n) \end{pmatrix}}}_{\vec{\tilde{f}}'}$$

Now all in one step: $\quad \vec{\tilde{\hat{f}}}' = \underbrace{V'V^{-1}}\vec{f}$

matrix to apply a derivative (!)

If you want coefficients of the derivative, use

$$V^{-1} V' V^{-1} \vec{f}.$$

$$\left(\text{to be cont'd} \right)$$

Demo: Taking derivatives with Vandermonde matrices