CS 450: Numerical Anlaysis

Lecture 19
Chapter 7 Interpolation
Basics of Interpolation

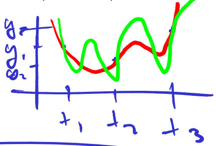
Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

March 28, 2018

Interpolation

- Given $(t_1, y_1), \dots, (t_m, y_m)$ with $t_1 < \cdots < t_m$ an *interpolant* f satisfies:



$f(t_i) = y_i$ for all i

with noisy data, we want usually to minimize residual of approximation that is smooth $\neq$ interpolation

interpolant works well for analytic function modeling

- Interpolant is usually constructed as linear combinations of *basis functions* $\{\phi_j\}_{j=1}^{n} = \{\phi_1, \dots, \phi_n\}$ so $f(t) = \sum_{j=1}^{n} x_j \phi_j(t)$.

$$V(t, \{\phi_j\}_{j=1}^{n}) = \begin{bmatrix} \phi_1(t_1) & \phi_2(t_1) & \cdots \\ \phi_1(t_2) & & \\ \vdots & & \\ & & \phi_n(t_m) \end{bmatrix} = A$$

$$y = Ax$$

interpolation usually works with $n = m$

unique interpolant

Polynomial Interpolation

- The choice of *monomials* as basis functions, $\phi_j(t) = t^{j-1}$ yields a degree $n-1$ polynomial interpolant:

Vandermonde matrix

$$V = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \\ 1 & t_4 & t_4^2 & t_4^3 \end{bmatrix}$$

Solving $Vx = y$ gives polynomial interpolant

$$f(t) = x_1 + x_2 t + x_3 t^2 + \dots$$

- Polynomial interpolants are easy to evaluate and do calculus on:

Horner's evaluation rule $f(t) = x_1 + t(x_2 + t(x_3 \dots))$

n - additions

n - multiplications

easy to $\int$, $\frac{d}{dx}$ polynomials

Conditioning of Interpolation

- Conditioning of interpolation matrix $\boldsymbol{A}$ depends on basis functions and coordinates $t_1, \ldots, t_m$:

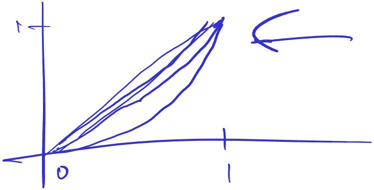
nodes

$A = V(t, \{\varphi\}_{i=1}^{n})$

nodes are far apart (in relative terms)

basis functions that are 'different' so that cols of A are not linearly independent

- The Vandermonde matrix tends to be ill-conditioned:



x^j looks like x^{j+1} for large j

naively also n^3 work to solve lin-sys with A

Lagrange Basis

- n-points fully define the unique $(n-1)$-degree polynomial interpolant in the Lagrange basis:

given n points, n-basis functions yield unique interpolant, with monomials give $n-1$ degree polynomial

$$\phi_j(t) = \underbrace{\prod_{i=1, i\neq j}^{n} (t - t_i)}_{\text{numerator}} \Big/ \underbrace{\prod_{i=1, i\neq j}^{n} (t_j - t_i)}_{\text{denominator}}$$

$\phi_j(t_j) = 1$

$\phi_j(t_i) = 0 \; (i \neq j)$

- Lagrange polynomials yield a convenient Vandermonde system, but the basis functions are hard to evaluate and do calculus on:

$$A = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = I$$

jth coefficient is y_j since only $\phi_j(t_j) \neq 0$ $= 1$ | $\phi_i(t_j) = 0$

solving system is trivial, evaluation is harder worst case $O(n^2)$

calculus is also harder

Newton Basis

- The Newton basis functions $\phi_j(t) = \prod_{k=1}^{j-1}(t - t_k)$ seek the best of monomial and Lagrange bases:

$\varphi_j(t) = 0$ if $t = t_i$ where $i < j$

• fast evaluation (monomial)
• easy Vandermonde system (Lagrange)

lower

- The Newton basis yields a triangular Vandermonde system:

$$\begin{bmatrix} \varphi_1(t_1) & \varphi_2(t_1) = 0 & & 0 \\ \varphi_1(t_2) & \vdots & \ddots & \\ \vdots & & & \end{bmatrix}$$

so solving $Ax = y$ has cost $O(n^2)$ via forward-substitution

Recurrences for Newton Basis

- The Newton basis functions $\phi_j(t) = \prod_{k=1}^{j-1}(t - t_k)$ can be evaluated at t with $O(n)$ work using a simple recurrence:

$$\phi_{j+1}(t) = \phi_j(t)(t - t_j)$$

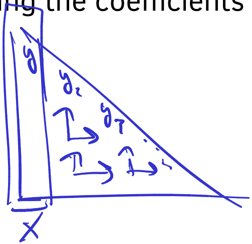
1 product per basis fn

Overall $O(n)$ work to evaluate

- A recurrence known as the divided-differences formula gives a stable way of efficiently computing the coefficients $\boldsymbol{x}$:

$$x = l_{i1}$$

$L =$

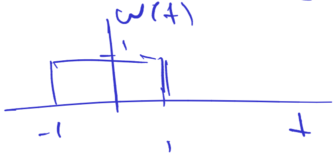


$$l_{ij} = \frac{l_{i,j-1} - l_{i+1,j}}{x_j - x_i}$$

Orthogonal Polynomials

- Recall that good conditioning for interpolation is achieved by constructing a well-conditioned Vandermonde matrix, which is the case when the columns (corresponding to each basis function) are orthonormal. To construct robust basis sets, we introduce a notion of *orthonormal functions*:

$$\langle p, q \rangle_w = \int_{-\infty}^{\infty} w(t)\, p(t)\, q(t)\, dt \quad \Big| \quad \text{e.g. } w(t) = 1 \text{ if } t \in [-1, 1]$$



$\{e_i\}_{i=1}^{n}$ are orthonormal if

$$\langle e_i, e_j \rangle_w = \delta_{ij} = \begin{cases} 1 & : \text{if } i = j \\ 0 & : \text{otherwise} \end{cases}$$

Legendre Polynomials

- The Gram-Schmidt orthogonalization procedure can be used to obtain an orthonormal basis with the same span as any given arbitrary basis:

Given $\{\varphi_i\}_{i=1}^n$ (arbitrary)

comp. orthonormal $\{\hat{\varphi}_i\}_{i=1}^n$

$\|f\| = \sqrt{\langle f, f \rangle_w}$

$\hat{\varphi}_i = \frac{\psi_i}{\|\psi_i\|}$ | $\psi_i = \varphi_i - \sum_{j=1}^{i-1} \hat{\varphi}_j \langle \hat{\varphi}_j, \varphi_i \rangle$

- The Legendre polynomials are obtained by Gram-Schmidt on the monomial basis, with normalization done so $\hat{\phi}_i(1) = 1$ and $w(t) = \begin{cases} 1 : -1 \le t \le 1 \\ 0 : \text{otherwise} \end{cases}$

let $\varphi_i(t) = t^{i-1}$ and Gram-Schmidt gives $\{\hat{\varphi}_i\}$

Legendre is slightly different, normalize so that $\varphi_i(1) = 1$, $\varphi_i(t)$ defined on $[-1, 1]$

Usually normalized function f on [-1, 1] with w = 1 must satisfy

$$\int_{-1}^{1} f(t)^2\,dt = 1$$

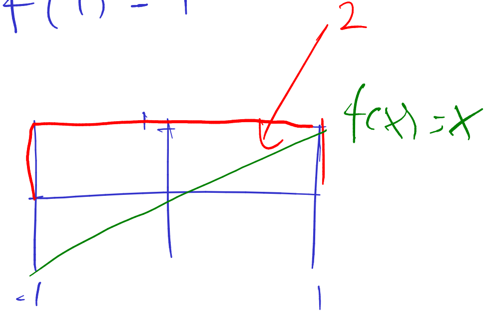
$$\langle f, f \rangle_w = 1$$

w = 1

$$\langle f, g \rangle_w = \int_a^b w(t) f(t) g(t)\,dt$$

Legendre

$$f(1) = 1$$



Legendre for $n=3$ is $\{1, x, (3x^2-1)/2\}$

$e_1(x) = 1$

$\varphi_2(x) = x - \frac{1}{2}\int_{-1}^{1} x \, dx = 0$

$\varphi_3(x) = x^2 - \frac{1}{2}\underbrace{\int_{-1}^{1} x^2 \, dx}_{1/3} - x\underbrace{\int_{-1}^{1} x^3 \, dx}_{0}$

$e_j(1) = 1$

$w(t) = 1$ on $[-1, 1]$