Polynomial Interpolation & Approximation

Goal: Approximate a complicated function $f(x)$ with a simpler function $g(x)$

3 measures of approximation

- interpolation: for a set $\{x_i\}$
  $f(x_i) = g(x_i)$
- least squares: $\int_a^b | f(t) - g(t) |^2 \, dt$ is minimal
- Chebyshev approximation

$$\min \max_{t \in [a,b]} | f(t) - g(t) |$$

Standard basis

$n+1$ points $\Leftrightarrow$ degree n polynomial

$\Rightarrow$ bigger $\Rightarrow$ no solution
smaller $\Rightarrow$ many solutions

Find $p(x) = \sum_{j=0}^{n} a_j x^j$ s.t. $p(x_i) = \underbrace{f(x_i)}_{y_i}$

Linear system solve for a_j

$$\underbrace{\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix}}_{\text{Vandermonde matrix}} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix}$$

<u>Fact</u>: $\det(A) = \prod_{0 \le j < i \le n} (x_i - x_j)$

<u>Consequence</u>: $\exists$ unique solution

<u>Downsides</u>: Numerically sensitive

Relatively expensive $O(n^3)$

Other bases

Lagrange :
$$\frac{(x-x_1)\cdots(x-x_n)}{(x_0-x_1)\cdots(x_0-x_n)}$$

① $$\cdots \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}$$

$$\cdots \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})}$$

② **Newton** : $1,\ (x-x_0),\ (x-x_0)(x-x_1)$

$\ldots,\ (x-x_0)(x-x_1)\cdots(x-x_{n-1})$

Both stable, Newton is cheaper for adding new points

Lagrange

x	1.4	1.25
y	3.7	3.9

$$p_1(x) = \left(\frac{x-1.25}{1.4-1.25}\right) 3.7 + \left(\frac{x-1.4}{1.25-1.4}\right) 3.9$$

$$= 3.7 - \frac{4}{3}(x-1.4)$$

$$p(x) = \underbrace{\left(\frac{x-x_1}{x_0-x_1}\right)}_{l_0(x)} y_0 + \underbrace{\left(\frac{x-x_0}{x_1-x_0}\right)}_{l_1(x)} y_1$$

$l_0(x_0)=1 \quad l_0(x_1)=0 \qquad l_1(x_0)=0 \quad l_1(x_1)=0$

$\Rightarrow$ general sum of n degree polynomials

$$l_i(x_j) = \begin{cases} 1 & j=i \\ 0 & \text{else} \end{cases}$$

$$l_i(x) = c_i \prod_{j \neq i} (x - x_j) \qquad i = 0, \dots, n$$

$$= \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}$$

see slides for proofs

$$p(x) = \sum_{i}^{n} l_i(x)\, y_i$$

Newton bases

$$p_n(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1)$$
$$\dots + c_n(x-x_0)\dots(x-x_{n-1})$$

Basis: $1, x-x_0, \dots \prod_{i=0}^{n-1}(x-x_i)$

Advantage of Newton

When adding points $x_{n+1} \dots x_{n+m}$
$c_0 \dots c_n$ do not change

For splines (where n is fixed)
Lagrange is more appropriate

Ex

$$p_2(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1)$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & x_1-x_0 & 0 \\ 1 & x_2-x_0 & (x_2-x_0)(x_2-x_1) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

* need only $O(n^2)$ operations

⇒ working out some details

$$c_0 = y_0 = f(x_0)$$

$$c_1 = \frac{y_1 - c_0}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$c_2 = \frac{y_2 - c_0 - (x_2 - x_0)\,c_1}{(x_2 - x_1)(x_2 - x_0)}$$

$$= \frac{f(x_2) - f(x_0) - (x_2 - x_0)\,\frac{f(x_1) - f(x_0)}{x_1 - x_0}}{(x_2 - x_1)(x_2 - x_0)}$$

$$= \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

$$f(x_2) - f(x_1) + f(x_1) - f(x_0) - (x_2 - x_0)\,\frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$f(x_2) - f(x_1) + \left(f(x_1) - f(x_0)\right)\underbrace{\left(1 - \frac{x_2 - x_0}{x_1 - x_0}\right)}$$

$$x_1 - \cancel{x_0} - x_2 + \cancel{x_0}$$

$$+x_1 - x_2$$

$$\frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{\left(f(x_1) - f(x_0)\right)\left(\frac{x_2 - x_1}{x_1 - x_0}\right)}{x_2 - x_1}}{x_2 - x_0}$$

Pattern $\quad \dfrac{f(x_j) - f(x_i)}{x_j - x_i} = f[x_i, x_j]$

$$c_1 = f[x_0, x_1] \qquad c_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

* Recursive calculation
* Method of divided difference

<u>Thm</u> $\quad c_i = f[x_0, \ldots x_i] \qquad i = 0, \ldots, n$

$$f[x_0, \ldots x_n] = \frac{f[x_1, \ldots x_n] - f[x_0 \ldots x_{n-1}]}{x_n - x_0}$$

$\Rightarrow$ usually computed via tables

see slides for discussion on tables & examples

185 - 189

Least squares approximation

$(x_0, y_0) \dots (x_n, y_n)$

$$E_{LSQ} = \sqrt{\sum_j (p_k(x_j) - y_j)^2}$$

$$= \sqrt{\sum_j (a_0 + a_1 x \dots + a_k x^k - y_i)^2}$$

Take partial derivatives = set to 0

$\Rightarrow$ get normal system

$$\begin{bmatrix} n & s_1 & \cdots & s_k \\ s_1 & s_2 & & \vdots \\ \vdots & & & \vdots \\ s_k & \cdots & \cdots & s_{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \vdots \\ \sum y_i x^k \end{bmatrix}$$

$s_1 = x_0 + \dots x_n$

$s_2 = x_0^2 + \dots + x_n^2$

$\vdots$

$s_{2k} = x_0^{2k} + \dots x_n^{2k}$

numerically unstable for large #'s of points

(could also go though $A^T A x = A^T b$)

$\hookrightarrow$ still bad

Change of basis

$$\{1, x, \dots x^k\} \rightarrow \{g_0(x), \dots, g_n(x)\}$$

$$a_0 \sum_i g_0^2(x_i) + a_1 \sum g_0(x_i) g_1(x_i)$$
$$\dots + a_n \sum g_0(x_i) g_n(x_i) = \sum f(x_i) g_0(x_i)$$

$$a_0 \sum_i g_1(x_i) g_0(x_i) + a_1 \sum g_1^2(x_i) + \dots$$

$$\vdots$$

* Choose basis so that $\sum g_j(x_i) g_k(x_i) = 0$ if $j \neq k$

$$A = \begin{pmatrix} \langle g_0, g_0 \rangle & \langle g_0, g_1 \rangle & \cdots & \langle g_0, g_n \rangle \\ \vdots & & & \vdots \\ \langle g_n, g_1 \rangle & \cdots & & \langle g_n, g_n \rangle \end{pmatrix}$$

$$x = \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} \quad y = \begin{pmatrix} \langle f, g_0 \rangle \\ \vdots \\ \langle f, g_n \rangle \end{pmatrix}$$

We call $g_0 \dots g_n$ w/ $\deg g_j = j$
first coeff of $g_j = 1$ & $\langle g_j, g_k \rangle = 0$
for $j \neq k$

$\Rightarrow$ orthogonal polynomials

$$p(x) = \sum a_i g_i(x) \Rightarrow a_i = \frac{\langle f, g_i \rangle}{\langle g_i, g_i \rangle}$$

Examples of bases

Legendre
Chebyshev
Hermite