Interpolation

\[ f \text{ interpolant or interpolating function} \]

Examples we have seen:
- Secant method for nonlinear equations
- Successive parabolic interpolation for minimization

Problem: Given data

\[(c_i, y_i) \quad i = 1, ..., m\]

\[ t_1 < t_2 < \ldots < t_m , \text{ find } f : \mathbb{R} \to \mathbb{R} \]

such that

\[ f(c_i) = y_i , \quad i = 1, ..., m \]

Note: could also impose other conditions on derivatives, smoothness, convexity, etc.
Purposes:
- plot smooth curve through data points
- read between lines of a table
- differentiate or integrate tabular data
- quick function evaluation
- replace complicated function by simple one

Historical: compute approximate values for functions from tables of data.

tool in approximating infinite-dimensional problems by finite-dimensional problems.

Example

\[(t_1, y_1), \quad (t_2, y_2)\] \[\implies f(t) = mt + b\]

\[t_1, t_2\]

equivalent: f more useful as we can immediately evaluate at other points \(t\), see slope, y-intercept.
Note: interpolation not always appropriate.

or

May want to smooth out noise

Many different functions may interpolate same data.

Examples
- polynomials
- piecewise polynomials
- trigonometric functions
- exponential functions
- rational functions
7.2. Existence, Uniqueness & Conditioning

match # of parameters + number of data pts.
too few params → no interpolant
too many params → not unique

\((t_i, y_i) \quad i = 1, \ldots, m\)

interpolant is chosen from set of
basic functions \(\phi_1(t), \ldots, \phi_n(t)\)

\[ f(t) = \sum_{j=1}^{n} x_j \phi_j(t) \]

parameters \(x_j\) are \(TBD\)

\[ f(t_1) = \sum_{j=1}^{n} x_j \phi_j(t_1) \]
\[ f(t_2) = \sum_{j=1}^{n} x_j \phi_j(t_2) \]
\[ \vdots \]
\[ f(t_m) = \sum_{j=1}^{n} x_j \phi_j(t_m) \]

\[
\begin{pmatrix}
\phi_1(t_1) & \phi_2(t_1) & \cdots & \phi_n(t_1) \\
\phi_1(t_2) & \phi_2(t_2) & \cdots & \phi_n(t_2) \\
\vdots & \vdots & \ddots & \vdots \\
\phi_1(t_m) & \phi_2(t_m) & \cdots & \phi_n(t_m)
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
= 
\begin{pmatrix}
y_1 \\
y_2 \\
\vdots \\
y_m
\end{pmatrix}
\]

\(A x = b\)
\(a_{ij} = \phi_j(t_i)\)
Cases:

- $m = n$  
  => $A$ square and $A$ nonsingular  
  unique solution for $x_1, \ldots, x_m$

- $m > n$  
  overdetermined  
  may not be exact solution  
  e.g. Least Squares solution instead

- $m < n$  
  underdetermined  
  impose additional conditions  
  - monotonicity, convexity, smoothness, (examples)

Sensitivity of $x$ to data depends on conditioning of $A$.

For different choice of basis functions even for same family of functions.
§7.3 Polynomial Interpolation

$P_k$ vector space of polynomials of degree at most $k$ (

$k+1$ dimensional)

Choice of basis functions affects
- cost of computing
- cost of evaluating
- sensitivity of parameters

7.3.1 Monomial Basis

$\Phi_j(t) = t^{j-1}$, $j = 1, \ldots, n$

i.e. $1, t, t^2, t^3, \ldots, t^{n-1}$

$p_{n-1}(t) = x_0 + x_1 t + x_2 t^2 + \ldots + x_{n-1} t^{n-1}$

$Ax = b$

\[
\begin{pmatrix}
1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\
1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\
1 & t_3 & t_3^2 & \cdots & t_3^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & t_n & t_n^2 & \cdots & t_n^{n-1}
\end{pmatrix}
\begin{pmatrix}
x_0 \\
x_1 \\
x_2 \\
\vdots \\
x_{n-1}
\end{pmatrix}
=
\begin{pmatrix}
y_0 \\
y_1 \\
y_2 \\
\vdots \\
y_{n-1}
\end{pmatrix}
\]

Vandermonde matrix

Necessarily non-singular if $t_i$ are distinct.

Proof: $A z = 0 \Rightarrow n$ roots of degree $n-1$ polynomial $\Rightarrow$ zero polynomial
Example (parabola)

\((-2, -27), (0, -1), (1, 0)\)

\[ p_2(x) = x_0 + x_1 + x_2 + \cdots \]

\[ f(x) = a_0 + a_1 x + a_2 x^2 \]

\[ f(-2) = a_0 + a_1 (-2) + a_2 4 = -27 \]

\[ f(0) = a_0 + a_1 \cdot 0 + a_2 0 = -1 \]

\[ f(1) = a_0 + a_1 \cdot 1 + a_2 \cdot 1 = 0 \]

\[ \Rightarrow \begin{pmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -27 \\ -1 \\ 0 \end{pmatrix} \]

\[ \Rightarrow \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 5 \\ -4 \end{pmatrix} \]

\[ f(x) = -1 + 5x - 4x^2 \]

Check:

\[ f(-2) = -1 - 10 - 4 \cdot 4 = -27 \checkmark \]

\[ f(0) = -1 \checkmark \]

\[ f(1) = -1 + 5 - 4 = 0 \checkmark \]
Vandermonde Matrix

- Solving $Ax = b$ requires $O(n^3)$ work
- $A$ often ill-conditioned, especially for high-degree polynomials.

\[
\begin{array}{c}
0 \\
1 \\
\vdots \\
1
\end{array}
\quad
\begin{array}{c}
0 \\
1 \\
\vdots \\
1
\end{array}
\]

Monomial basis functions become progressively less distinguishable, leading to columns in $A$ that become progressively less distinguishable for most choices of $t$; $\text{Ker}(A)$ grows at least exponentially in $n$.

A nonsingular in theory, but arbitrarily ill-conditioned.

Polynomial will still fit data points well, as GE of partial pivoting produces small residual, but values of coefficients will be poorly determined.

Both conditioning and amount of work can be improved using a different basis.
Interpolating polynomial is unique.

Proof:

Assume \( p_1(x) \) is degree \( n-1 \).

\( p_2(x) \) is also an interpolating \( (x_i, y_i) \)

\[ p_2(x_i) - p_1(x_i) = y_i, \quad i = 1, \ldots, n \]

Then \( p(x_i) = p_1(x_i) - p_2(x_i) = 0 \) for \( i = 1, \ldots, n \)

degree \( n-1 \)

with \( n \) roots \( \Rightarrow 0 \) polynomial

Improved monomial basis, shifted and scaled:

\[ \phi_j(t) = \left( \frac{t-c}{d} \right)^{j-1} \]

\[ c = \frac{t_1 + t_n}{2}, \quad d = \frac{t_n - t_1}{2} \]

Better, but still not well conditioned.

Evaluation

Efficient evaluation using Horner's Method

\[ p_n(x) = a_n + x(a_{n-1} + x(a_{n-2} + \cdots + a_1 + x a_0) \cdots)) \]

\( n+1 \) additions and \( n+1 \) multiplications

Example:

\[ 4x^4 + 5x^3 - 2x^2 + 3x^2 = -4(8 + 5(x - 2y(x + 3x)) \]

\[ 4x^4 + 4 \]

\[ 1 + x(-4 + x(5 + x(-2 + 3x))) \]

\( 4 + 5 \) and \( 4x^5 \)
Same principle applies in forming Vandermonde matrix
\[
\begin{pmatrix}
1 & x_1 & x_1^2 \\
1 & x_2 & x_2^2 \\
1 & x_3 & x_3^2 \\
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
a_2 \\
\end{pmatrix}
= 
\begin{pmatrix}
y_0 \\
y_1 \\
y_2 \\
\end{pmatrix}
\]

\[a_{ij} = x_i \cdot a_{i,j-1}\]

Superior to explicit exponentiation.
§7.3.2 Lagrange Interpolation

\[(x_i, y_i) \quad i = 1, \ldots, n\]

Lagrange basis functions for \(P_{n-1}\) (Fundamental polynomials)

\[
l_j(x) = \prod_{k \neq j} \frac{(x-x_k)}{(x_j-x_k)} \quad j = 1, \ldots, n
\]

Note: \(l_j(x)\) is a polynomial of degree \(n-1\)

\[
\sum_{j=1}^{n} l_j(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}
\]

Recall \(a_{ij} = l_j(x_i) = \delta_{ij} = 1\)

\[\Rightarrow\] coefficients are \(y_i\)

\[p(x) = y_1 l_1(x) + y_2 l_2(x) + \cdots + y_n l_n(x)\]
+ easy to determine interpolating polynomial
+ parameters perfectly conditioned

- Lagrange form is more expensive to evaluate
- more difficult to differentiate, integrate, etc.

Example Lagrange Interpolation

\[ (-2, 27), (0, -1), (1, 0) \]

\[ p(x) = \frac{-27}{(-2-0)(-2-1)} (x-0)(x-1) + \frac{-1}{(0-2)(0-1)} (x-2)(x-1) + 0 \]

\[ = \frac{-27}{6} x(x-1) + \frac{x+20}{2}(x-1) \]
Newton Interpolation

Data points $(X_i, Y_i), i = 1, \ldots, n$

Basis functions $\Pi_j(x) = \prod_{k=1}^{j-1} (x - X_k), j = 1, \ldots, n$

\[
\begin{align*}
\Pi_1(x) &= 1 \\
\Pi_2(x) &= (x - X_1) \\
\Pi_3(x) &= (x - X_1)(x - X_2) \\
&\vdots \\
\Pi_n(x) &= (x - X_1)(x - X_2) \cdots (x - X_{n-1}) \\
\end{align*}
\]

\[p(x) = a_1 \Pi_1(x) + a_2 \Pi_2(x) + \cdots + a_n \Pi_n(x)\]

\[
\begin{pmatrix}
\Pi_1(X_1) & \Pi_1(X_2) & \cdots & \Pi_1(X_n) \\
\Pi_2(X_1) & \Pi_2(X_2) & \cdots & \Pi_2(X_n) \\
\vdots & \vdots & & \vdots \\
\Pi_n(X_1) & \Pi_n(X_2) & \cdots & \Pi_n(X_n)
\end{pmatrix}
\begin{pmatrix}
a_1 \\
a_2 \\
\vdots \\
a_n
\end{pmatrix}
=
\begin{pmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_n
\end{pmatrix}
\]

Note: $\Pi_j(X_i) = 0$ when $i < j$
Example \((-2, -27), (0, 1), (1, 0)\)

\[\Pi_1(x) = x \quad 1\]
\[\Pi_2(x) = x - x_1 = x + 2\]
\[\Pi_3(x) = (x - x_1)(x - x_2) = (x + 2)(x)\]

\[
\begin{pmatrix}
1 & 0 & 0 \\
1 & 2 & 0 \\
1 & 3 & 3
\end{pmatrix}
\begin{pmatrix}
a_1 \\
a_2 \\
a_3
\end{pmatrix}
= \begin{pmatrix}
-27 \\
-1 \\
0
\end{pmatrix}
\Rightarrow \begin{align*}
a_1 &= -27 \\
a_2 &= 13 \\
a_3 &= -4
\end{align*}

\[p(x) = -27 + 13(x + 2) + -4(x + 2)x\]

**Evaluation:** can be efficiently evaluated using Horner's rule.

\[p(x) = a_1 \Pi_1(x) + a_2 \Pi_2(x) + \cdots + a_n \Pi_n(x)\]

\[= a_1 + a_2 (x - x_1) + a_3 (x - x_1)(x - x_2) + \cdots + a_n (x - x_1)(x - x_2) \cdots (x - x_{n-1})\]

\[= a_1 + (x - x_1)(a_2 + (x - x_2)(a_3 + (x - x_3)(\cdots (a_{n-1} + a_n(x - x_{n-1}))\cdots)))\]
Newton Interpolation

**incremental**

\[ P_n(x) \] polynomial of degree \( n-1 \)
interpolates \((x_i, y_i), i = 1, \ldots, n\)

\[ P_{n+1}(x) \] polynomial of degree \( n \)
interpolates \((x_i, y_i), i = 1, \ldots, n\)
and \((x_{n+1}, y_{n+1})\)

\[ P_{n+1}(x) = P_n(x) + a_{n+1} \prod_{n+1}(x) \]

(since \( \prod_{n+1}(x_i) = 0 \) \( \Rightarrow P_{n+1}(x_i) = P_n(x_i) = y_i \))

Choose \( a_{n+1} \) to interpolate \((x_{n+1}, y_{n+1})\)

\[ P_{n+1}(x_{n+1}) = P_n(x_{n+1}) + a_{n+1} \prod_{n+1}(x_{n+1}) = y_{n+1} \]

\[ \Rightarrow a_{n+1} = \frac{y_{n+1} - P_n(x_{n+1})}{\prod_{n+1}(x_{n+1})} \]
(x_1, y_1) \ldots (x_n, y_n)

coefficients through divided differences

\[ f[x_1, x_2, \ldots, x_k] = \frac{f[x_2, \ldots, x_k] - f[x_1, \ldots, x_{k-1}]}{x_k - x_1} \]

\[ f[x_{i_0}] = y_{i_0} \]

Example \((-2, -27), (0, -1), (1, 0)\)
7.3.4. Orthogonal Polynomials.

Inner product:

\[(p, q) = \int_a^b p(t)q(t)w(t)dt\]

\(w(t)\) non-negative weight function

\(p, q\) are orthogonal if \((p, q) = 0\)

\(p, p\) or monomial if

\[(p_i, p_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}\]

Given set of polynomials, G-S orthogonalization process can be used to get normalized set spanning the same space.

E.g. take \(w(t) = 1\) on \([-1, 1]\)

apply G-S to monomials \(1, t, t^2, t^3, \ldots\), and scale so that \(p_0(1) = 1\), get Legendre polynomials

\[1, t, \frac{(3t^2-1)}{2}, \frac{(5t^3-3t)}{2}, \frac{(35t^4-30t^2+3)}{8}, \frac{(63t^5-70t^3+15t)}{2}, \ldots\]
\[ 1, x, x^2, x^3 \] on \([-1, 1]\)

First, orthonormal set:

\[ J_1 = |x| \bigg|_{-1}^{1} = 1 - (-1) = 2 \Rightarrow p_0(x) = \frac{1}{2} \]

\[ \Phi (\frac{1}{2}, x) = \int_{-1}^{1} \frac{1}{2} \, dx = \frac{1}{2} \int_{-1}^{1} 1 = \frac{1}{2} \left[ \frac{x}{2} \right]_{-1}^{1} = \frac{1}{2} \left[ -\frac{1}{2} \right] = 0 \]

\[ p_1(x) = \int_{-1}^{1} x \, dx = \frac{1}{3} \left[ x^3 \right]_{-1}^{1} = \frac{1}{3} (1 + 1) = \frac{2}{3} \Rightarrow p_1(x) = \sqrt{\frac{3}{2}} x \]

\[ \int_{-1}^{1} \frac{3}{2} x^2 \, dx = \frac{3}{2} \left[ x^3 \right]_{-1}^{1} = \frac{3}{2} \left[ \frac{1}{3} + 1 \right] = \frac{3}{2} \frac{2}{3} = 1 \sqrt{\frac{3}{2}} \]

Orthogonal polynomials satisfy 3-term recurrence:

\[ p_{k+1}(x) = (c_k x + \beta_k) p_k(x) - \gamma_k p_{k-1}(x) \]