APROXIMACION DE FUNCIONES

Gran parte de las aproximaciones hechas en Análisis Numérico consisten en aproximar una función $g(x)$ por una combinación de funciones, partiendo de alguna clase de funciones conocidas.

Supongamos dada $g(x)$ y $\{f_n(x),\ n = 0,1,\dots\}$ $\therefore$

$$g(x) \approx a_1 f_0(x) + a_1 f_1(x) + \dots a_n f_n(x)$$

Existen distintos criterios para elegir las a_i dando lugar a distintos tipos de aproximación:

- Aproximaciones exactas o por interpolación
- Aproximaciones por mínimos cuadrados
- Aproximaciones de error mínimo-máximo

Familia de Funciones Bases

- Potencias : $1, x, x^2, x^3, \dots$
  son funciones base para polinomios
- Serie de Fourier: 1, sin(ωt), cos(ωt), sin$(2\omega t)$,...
  se usan para funciones periodicas.
- Fns. Spline : actualmente son muy utilizadas para funciones no periodicas

Interpolación

Supongamos tener un conjunto de puntos ya sean datos medidos o valores de una tabla, buscamos una función que pase por esos puntos y además acotar el error cometido. Trabajaremos con polinomios por ser estables.

Interpolación Polinomial

Supongamos que $x_0, x_1, \dots, x_n$ son n+1 puntos $\in [a,b]$ y sea $f(x)$: $[a,b] \longrightarrow \Re$

Queremos construir un polinomio p(x), de grado ≤ n que interpole a $f(x)$ en esos puntos

$$p(x_i) = f(x_i)$$

si

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

tenemos un sistema de n+1 ecuaciones con n+1 incógnitas

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 + \dots + a_n x_0^n = f(x_0) \\ a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n = f(x_1) \\ \dots \\ a_0 + a_1 x_n + a_2 x_n^2 + \dots + a_n x_n^n = f(x_n) \end{cases}$$

En forma matricial:

$$\vec{X}\,\vec{a} = f(\vec{X}) \qquad \vec{X}: \text{matriz de Vandermonde}$$

Se llama Método de los Coeficientes Indeterminados

Teorema de aproximación de Weierstrass

Suponga que *f* está definida y es continua en [*a*, *b*]. Para cada $\varepsilon > 0$ existe un polinomio *p(x)* definido en [*a*, *b*], con la propiedad de que

$$|f(x) - P(x)| < \varepsilon, \text{ para toda } x \text{ en } [a, b]$$

Teorema de Existencia y Unicidad

Dados $n+1$ puntos distintos x_i $i = 0,\dots,n$ y n+1 ordenadas y_i existe y es único un polinomio interpolante $p(x)$ de grado ≤ n que interpola a los y_i en los puntos x_i

Polinomios de LAGRANGE

$$l_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)}$$

FORMA DE LAGRANGE

$$p(x) = \sum_{k=0}^{n} f(x_k) l_k(x)$$

EJEMPLO : Interpolar en x=2

x	$f(x)$
1	0
4	1.386294
6	1.791760

i) Usando P_0 y P_1 :

$$l_0(x) = \frac{x-4}{1-4} \qquad l_1(x) = \frac{x-1}{4-1}$$

$$p(x) = l_0 f(x_0) + l_1 f(x_1)$$

$$p(2) = \frac{2-4}{1-4}\,0 + \frac{2-1}{4-1}\,1.386294 = 0.4620981$$

ii) Usando P_0 , P_1 y P_2 :

$$p(x) = l_0 f(x_0) + l_1 f(x_1) + l_2 f(x_2)$$

$$p(2) = \frac{(2-4)(2-6)}{(1-4)(1-6)}\,0 + \frac{(2-1)(2-6)}{(4-1)(4-6)}\,1.386294 + \frac{(2-1)(2-4)}{(6-1)(6-4)}\,1.79176 = 0.565844$$

ERROR DE INTERPOLACION

Teorema

Sea $f(x):[a,b]\rightarrow \Re,\quad n+1$ veces diferenciable en (a,b), y sean $\{x_i\}\ i=0,1,...,n$,n+1 puntos distintos en [a,b].

Si $p_n(x)$ es el polinomio de grado $\leq n$ que interpola a $f(x)$ en, $x_0,\ldots,x_n$ $\therefore$ para cada $x\in[a,b]$ $\exists\ \xi\in(a,b)$ tal que:

$$f(x)-p_n(x)=\frac{(x-x_0)\ldots(x-x_n)}{(n+1)!}f^{n+1}(\xi),\quad \xi\in(a,b)$$

Método de LAGRANGE

- Evita resolver el sistema de ecuaciones (O(n^3))
- Tiene un costo del orden de O(n^2)
- Para estimar una cota del error se necesita conocer la derivada de orden n+1
- Si se agrega un punto, hay que recalcular todos los coeficientes

FORMA DE NEWTON

La idea en este método es, dado p_{n-1} poder calcular p_n reusando lo anterior

$$p_n(x)=p_{n-1}(x)+C(x)$$

$$C(x)=p_n(x)-p_{n-1}(x)$$

$C(x)$ debe ser de orden n, además

$$p_n(x)=a_0+a_1(x-x_0)+\ldots+a_{n-1}(x-x_0)\ldots(x-x_{n-2})+a_n(x-x_0)\ldots(x-x_{n-1})$$

$$-\quad p_{n-1}(x)=a_0+a_1(x-x_0)+\ldots+a_{n-1}(x-x_0)\ldots(x-x_{n-2})$$

$$C(x)=a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

O sea que tiene los n ceros $x_0,\ldots,x_{n-1}$ y como $p_n(x_n)=f(x_n)$

$$a_n=\frac{f(x_n)-p_{n-1}(x_n)}{(x_n-x_0)\ldots.(x_n-x_{n-1})}\implies f[x_0,x_1,x_2,\ldots,x_n]$$

n-ésima diferencia dividida de f en $x_0,\ldots,x_n$

Forma de NEWTON

Con 1 punto $(x_0, f(x_0)) \rightarrow p_0(x)=a_0=f(x_0)$

Con 2 puntos $(x_0, f(x_0))$ y $(x_1, f(x_1))$

$$p_1(x)=a_0+a_1(x-x_0)$$

$$p_1(x_1)=f(x_0)+a_1(x_1-x_0)=f(x_1)$$

$$a_1=\frac{f(x_1)-f(x_0)}{(x_1-x_0)}$$

$$p_1(x)=f(x_0)+\frac{f(x_1)-f(x_0)}{(x_1-x_0)}(x-x_0)$$

$$p_1(x)=f(x_0)+f[x_0,x_1](x-x_0)$$

Con n puntos

$$p_n(x)=p_{n-1}(x)+f[x_0,x_1,\ldots,x_n](x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

Forma de Newton de Diferencias Divididas

Propiedades de las Diferencias Divididas

i) Las diferencias divididas se pueden escribir recursivamente en función de diferencias divididas anteriores

$$f[x_i]=f(x_i)\qquad i=0,1,..,n-1$$

$$\ldots$$

$$f[x_i,x_{i+1}]=\frac{f[x_{i+1}]-f[x_i]}{x_{i+1}-x_i}$$

$$f[x_i,x_{i+1},x_{i+2},\ldots,x_{i+k}]=\frac{f[x_{i+1},x_{i+2},\ldots,x_{i+k}]-f[x_i,x_{i+1},\ldots,x_{i+k-1}]}{x_{i+k}-x_i}$$

ii) Si los z_i son un reordenamiento de los x_i

$$f[x_0,x_1,x_2,\ldots,x_n]=f[z_1,z_0,z_2,\ldots,z_n]$$

Tabla de Diferencias Divididas

i	Xi	f(Xi)	Primera	Segunda	Tercera
0	X0	f(X0)			
1	X1	f(X1)	f[X0, X1]		
2	X2	f(X2)	f[X1, X2]	f[X0, X1, X2]	
3	X3	f(X3)	f[X2,X3]	f[X1, X2, X3]	f[X0, X1, X2, X3]

Error de interpolación al usar Newton

Sea $f(x):[a,b]\to\Re$ y sean $\{x_i\}\ i=0,1,\dots,n$, n+1 puntos distintos en [a,b]. Si $p_n(x)$ es el polinomio de grado < n que interpola a $f(x)$ en $x_0,\dots,x_n$ entonces, el error que se comete en la interpolación viene dado por: $e_n = f(x)-p_n(x)$

La forma de $p_{n+1}(x)$ usando Newton :

$$p_{n+1}(x) = f(x_0)+f[x_0,x_1](x-x_0)+\dots+f[x_0,x_1,\dots,x_{n+1}](x-x_0)\dots(x-x_n)$$

Por ser interpolante : $p_{n+1}(x_{n+1}) = f(x_{n+1})$

O sea que:

$$f(x_{n+1}) = \underbrace{f(x_0)+\dots+f[x_0,x_1,\dots,x_n](x_{n+1}-x_0)\dots(x_{n+1}-x_{n-1})}_{p_n(x_{n+1})}+f[x_0,x_1,\dots,x_{n+1}](x_{n+1}-x_0)\dots(x_{n+1}-x_n)$$

Por lo tanto:

$$e_n = f(x_{n+1})-p_n(x_{n+1}) = f[x_0,x_1,\dots,x_{n+1}](x_{n+1}-x_0)\dots(x_{n+1}-x_n)$$

Si comparamos con la fórmula de error vista antes:

$$f(x_{n+1})-p_n(x_{n+1}) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x_{n+1}-x_0)(x_{n+1}-x_1)\cdots(x_{n+1}-x_n)$$

vemos que:

$$f[x_0,x_1,\cdots,x_n,x_{n+1}] = \frac{f^{(n+1)}(\xi)}{(n+1)!} \qquad \xi\in(a,b)$$

es decir que con un punto más podemos calcular el error

$$\boxed{e_n = f[x_0,x_1,\cdots,x_n,x_{n+1}](x-x_0)(x-x_1)\dots(x-x_n)}$$

Regla del Término Siguiente

Observaciones:

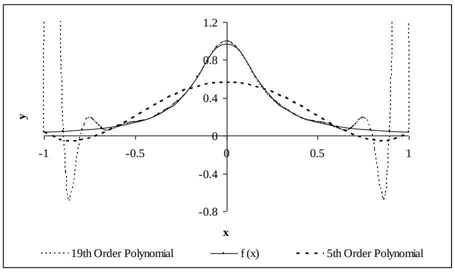
- El polinomio de interpolación suele usarse para estimar valores de una función tabulada, en las abscisas que no aparecen en la tabla.
- El aumento de grado no siempre mejora la aproximación.
- El polinomio es muy sensible a los errores de los datos
- Tipos de error:

  redondeo: datos , coeficientes, aproximación

  truncamiento: depende de la derivada

Ejemplo: Función de Runge

$$f(x)=\frac{1}{1+25x^2} \qquad [-1,1]$$



Interpolación Polinómica Segmentaria

Dados n+1 puntos $(x_0,y_0), (x_1,y_1), \dots, (x_n,y_n)$ con $x_0< x_1 \dots < x_n$, una función spline de orden k (k-Spline) sobre dichos puntos es una función S verificando:

(i) $S(x) = q_k(x)$ polinomio de grado $\le k$, $x\in[x_k,x_{k+1}]$, $k=0,1,\dots,n-1$

(ii) $S(x_k) = y_k$, $k=0,1,\dots,n$

(iii) $S\in C^{k-1}[x_0,x_1]$

Aplicaciones

- Ingeniería y Diseño (CAD/CAM, CNC's)
- Geología
- Aeronáutica y automoción
- Procesamiento de señales e imágenes (Reconocimiento de patrones, recuperación de imágenes)
- Robótica
- Medicina (Aparatos auditivos, mapas cerebrales)
- Meteorología (Mapas climáticos, detección de inundaciones)
- etc.

Cubic Spline

Sea $f(x):[a,b]\rightarrow\Re$ y sean $\{x_i\}\; i = 0,1,\ldots,n$ n+1 puntos distintos en [a,b], a =x0 < x1 < x2 … < xn = b

a) En cada intervalo $[x_i, x_{i+1}]$, S es un polinomio cúbico denotado por $S_i(x)$.

b) $S_i(x_i) = f(x_i)$, i = 0,…, n-2

c) $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$

d) $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1})$

e) $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1})$

f) Se satisface alguna de las siguientes condiciones de frontera: $S''(x_0) = S''(x_n) = 0$ (frontera libre)

$S'(x_0) = f'(x_0)$ y $S'(x_n) = f'(x_n)$ (frontera sujeta)

Se plantea un sistema de ecuaciones para calcular los coeficientes del polinomio $S_i(x)$:

a) $S_i(x) = a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3, \; i = 0,1,\ldots,n-1$

b) $S_i(x_i) = f(x_i) = a_i$

si $h_i = x_{i+1} - x_i$ y utilizando las otras propiedades obtenemos:

$$h_{i-1}c_{i-1} + 2(h_{i+1} - h_i)c_i + h_i c_{i+1} = \frac{3}{h_i}[a_{i+1} - a_i] - \frac{6}{h_{i-1}}[a_i - a_{i-1}]$$

Esta matriz es tridiagonal, diagonalmente dominante, o sea que siempre tiene solución y se resuelve con el método de Thomas

$$\begin{pmatrix} 2(h_1+h_0) & h_1 & \cdots & 0 \\ h_1 & 2(h_2+h_1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_{n-2} \\ 0 & \cdots & h_{n-2} & 2(h_{n-1}+h_{n-2}) \end{pmatrix}$$

Una vez calculados los ci:

$$b_i = \frac{1}{h_i}[a_{i+1} - a_i] - \frac{h_i}{3}[2c_i + c_{i+1}]$$

$$d_i = \frac{(c_{i+1} - c_i)}{3h_i}$$

Si el punto a interpolar está en el intervalo (x_i , x_{i+1}) se debe utilizar la función S_i

Error de Interpolación

Al usar una spline natural para interpolar una función *f(x)*, el error es proporcional a h^4. Lo mismo ocurre cuando utilizamos una spline cúbica sujeta.