Interpolacion en ingenieria estadistica e informatica

Ronal Wilder

1. Interpolación Lineal

1.1. Marco Teórico

La interpolación lineal constituye el método más elemental y frecuentemente utilizado para estimar valores intermedios de una función a partir de dos puntos conocidos. Dadas las coordenadas (x_0, y_0) y (x_1, y_1), se asume una relación lineal entre ambos puntos, lo cual permite expresar la función interpoladora como:

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \tag{1}$$

Esta expresión puede reescribirse en forma equivalente como:

$$y = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0} \tag{2}$$

La interpolación lineal es ampliamente empleada en ingeniería informática para tareas como el suavizado de gráficos, la estimación de datos faltantes en conjuntos de datos discretos, y la predicción de comportamientos en sistemas de tiempo real.

1.2. Casos de Aplicación

Ejercicio 1: Estimación del tiempo de procesamiento en algoritmos paralelos

Un equipo de investigación analiza el desempeño de un algoritmo de procesamiento distribuido, registrando los siguientes tiempos promedio de ejecución:

$$(4\,\text{nodos}, 22{,}5\,\text{s}), \quad (10\,\text{nodos}, 11{,}2\,\text{s})$$

Se requiere estimar el tiempo cuando se emplean 7 nodos. Aplicando la fórmula de interpolación lineal:

$$y = 22{,}5 + \frac{11{,}2 - 22{,}5}{10 - 4}(7 - 4) = 22{,}5 - \frac{11{,}3}{6} \cdot 3 \approx 16{,}85\,\text{s}$$

Conclusión: Con 7 nodos, el tiempo estimado de ejecución es aproximadamente 16.85 segundos, lo cual refleja una reducción proporcional del tiempo conforme aumenta el paralelismo.

Ejercicio 2: Estimación de probabilidad acumulada

En el análisis de un proceso estocástico, se dispone de la siguiente información sobre su función de distribución acumulada:

$$(x_0, F(x_0)) = (1{,}5, 0{,}45), \quad (x_1, F(x_1)) = (2{,}0, 0{,}63)$$

Se desea estimar $F(1{,}8)$ mediante interpolación lineal:

$$F(1{,}8) = 0{,}45 + \frac{0{,}63 - 0{,}45}{2{,}0 - 1{,}5}(1{,}8 - 1{,}5) = 0{,}45 + \frac{0{,}18}{0{,}5} \cdot 0{,}3 = 0{,}558$$

Conclusión: La probabilidad acumulada estimada es $F(1{,}8) = 0{,}558$, valor coherente con una distribución continua creciente.

Ejercicio 3: Calibración de sensores digitales

Durante el proceso de calibración de un sensor de presión, se obtienen las siguientes lecturas:

$$(0{,}8\,\text{V}, 25\,\text{kPa}), \quad (2{,}4\,\text{V}, 80\,\text{kPa})$$

Se necesita determinar la presión correspondiente a una lectura de 1.5 V:

$$y = 25 + \frac{80 - 25}{2{,}4 - 0{,}8}(1{,}5 - 0{,}8) = 25 + \frac{55}{1{,}6} \cdot 0{,}7 \approx 49{,}06\,\text{kPa}$$

Conclusión: Para una señal de 1.5 V, se estima una presión de aproximadamente 49.1 kPa.

2. Interpolación de Lagrange

2.1. Fundamentos del Método

El método de interpolación de Lagrange permite construir un polinomio de grado n que pase exactamente por $n+1$ puntos dados. Este polinomio se construye como una combinación lineal de funciones base especiales, denominadas *polinomios de Lagrange*.

Dado un conjunto de puntos $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, el polinomio interpolador se define como:

$$P(x) = \sum_{i=0}^{n} y_i L_i(x) \tag{3}$$

donde cada función base $L_i(x)$ está dada por:

$$L_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{4}$$

Una propiedad fundamental de estos polinomios es que $L_i(x_k) = \delta_{ik}$ (delta de Kronecker), lo cual garantiza que $P(x_i) = y_i$ para todo i.

2.2. Aplicaciones Prácticas

Ejercicio 1: Rendimiento energético de procesadores

Se midió la eficiencia energética (MIPS/Watt) de un procesador bajo distintas frecuencias de operación:

$$(2{,}0\,\text{GHz}, 65), \quad (2{,}5\,\text{GHz}, 70), \quad (3{,}0\,\text{GHz}, 60)$$

Para estimar el rendimiento a 2.3 GHz, calculamos primero los polinomios base:

$$L_0(x) = \frac{(x-2{,}5)(x-3{,}0)}{(2{,}0-2{,}5)(2{,}0-3{,}0)} = 2(x-2{,}5)(x-3{,}0)$$

$$L_1(x) = \frac{(x-2{,}0)(x-3{,}0)}{(2{,}5-2{,}0)(2{,}5-3{,}0)} = -4(x-2{,}0)(x-3{,}0)$$

$$L_2(x) = \frac{(x-2{,}0)(x-2{,}5)}{(3{,}0-2{,}0)(3{,}0-2{,}5)} = 2(x-2{,}0)(x-2{,}5)$$

Evaluando en $x = 2{,}3$:

$$L_0(2{,}3) = 2(-0{,}2)(-0{,}7) = 0{,}28$$
$$L_1(2{,}3) = -4(0{,}3)(-0{,}7) = 0{,}84$$
$$L_2(2{,}3) = 2(0{,}3)(-0{,}2) = -0{,}12$$

Por lo tanto:

$$P(2{,}3) = 65(0{,}28) + 70(0{,}84) + 60(-0{,}12) = 18{,}2 + 58{,}8 - 7{,}2 = 69{,}8$$

Conclusión: A 2.3 GHz, se estima un rendimiento de aproximadamente 69.8 MIPS/Watt.

Ejercicio 2: Densidad de tráfico en sistemas distribuidos

Un sistema de almacenamiento distribuido registra la tasa de procesamiento (miles de paquetes por segundo) en función del número de servidores activos:

$$(3, 12), \quad (5, 25), \quad (7, 40)$$

Para estimar el throughput con 6 servidores, calculamos:

$$L_0(6) = \frac{(6-5)(6-7)}{(3-5)(3-7)} = \frac{-1}{8} = -0{,}125$$
$$L_1(6) = \frac{(6-3)(6-7)}{(5-3)(5-7)} = \frac{-3}{-4} = 0{,}75$$
$$L_2(6) = \frac{(6-3)(6-5)}{(7-3)(7-5)} = \frac{3}{8} = 0{,}375$$

$$P(6) = 12(-0{,}125) + 25(0{,}75) + 40(0{,}375) = -1{,}5 + 18{,}75 + 15 = 32{,}25$$

Conclusión: Se estima una densidad de 32.25 mil paquetes por segundo con 6 servidores.

Ejercicio 3: Función de pérdida en entrenamiento de redes neuronales

Durante el entrenamiento de una red neuronal, se registran los siguientes valores de la función de pérdida:

$$(1, 0{,}35), \quad (3, 0{,}22), \quad (5, 0{,}18)$$

Para estimar el costo en la época 2:

$$L_0(2) = \frac{(2-3)(2-5)}{(1-3)(1-5)} = \frac{3}{8} = 0{,}375$$
$$L_1(2) = \frac{(2-1)(2-5)}{(3-1)(3-5)} = \frac{-3}{-4} = 0{,}75$$
$$L_2(2) = \frac{(2-1)(2-3)}{(5-1)(5-3)} = \frac{-1}{8} = -0{,}125$$

$$P(2) = 0{,}35(0{,}375) + 0{,}22(0{,}75) + 0{,}18(-0{,}125) \approx 0{,}274$$

Conclusión: El valor de pérdida estimado en la época 2 es aproximadamente 0.274.

3. Diferencias Divididas de Newton

3.1. Descripción del Método

El método de diferencias divididas de Newton permite construir el polinomio interpolador de forma incremental, lo cual resulta especialmente útil cuando se agregan nuevos puntos de datos sin necesidad de recalcular todo el sistema.

Las diferencias divididas se definen recursivamente como:

$$f[x_i] = y_i \tag{5}$$

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \tag{6}$$

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} \tag{7}$$

El polinomio resultante adopta la forma:

$$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \tag{8}$$

Este método es particularmente eficiente en aplicaciones donde los datos se van obteniendo gradualmente, como en monitoreo en tiempo real o análisis de rendimiento de sistemas.

3.2. Ejemplos de Aplicación

Ejercicio 1: Tiempo de cómputo en algoritmos de clasificación

Se realizan mediciones del tiempo promedio (en segundos) requerido por un algoritmo de clasificación para procesar diferentes tamaños de muestra:

$$(1000, 0{,}8), \quad (2000, 1{,}6), \quad (5000, 4{,}5)$$

Calculamos las diferencias divididas:

$$f[x_0] = 0{,}8, \quad f[x_1] = 1{,}6, \quad f[x_2] = 4{,}5$$

$$f[x_0, x_1] = \frac{1{,}6 - 0{,}8}{2000 - 1000} = 0{,}0008$$

$$f[x_1, x_2] = \frac{4{,}5 - 1{,}6}{5000 - 2000} = 0{,}000967$$

$$f[x_0, x_1, x_2] = \frac{0{,}000967 - 0{,}0008}{5000 - 1000} = 4{,}17 \times 10^{-8}$$

El polinomio de Newton es:

$$P(x) = 0{,}8 + 0{,}0008(x - 1000) + 4{,}17 \times 10^{-8}(x - 1000)(x - 2000)$$

Para estimar el tiempo con 3000 muestras:

$$P(3000) = 0{,}8 + 0{,}0008(2000) + 4{,}17 \times 10^{-8}(2000)(1000) \approx 2{,}48\,\mathrm{s}$$

Conclusión: Se estima un tiempo de procesamiento de 2.48 segundos para 3000 muestras.

Ejercicio 2: Uso de memoria en sistemas paralelos

Se registra el consumo de memoria (en GB) de un servidor según el número de procesos concurrentes:

$$(5, 2{,}3), \quad (10, 4{,}1), \quad (20, 7{,}8)$$

Diferencias divididas:

$$f[x_0, x_1] = \frac{4{,}1 - 2{,}3}{10 - 5} = 0{,}36$$
$$f[x_1, x_2] = \frac{7{,}8 - 4{,}1}{20 - 10} = 0{,}37$$
$$f[x_0, x_1, x_2] = \frac{0{,}37 - 0{,}36}{20 - 5} = 0{,}00067$$

Polinomio:

$$P(x) = 2{,}3 + 0{,}36(x - 5) + 0{,}00067(x - 5)(x - 10)$$

Para 15 procesos:

$$P(15) = 2{,}3 + 0{,}36(10) + 0{,}00067(10)(5) = 2{,}3 + 3{,}6 + 0{,}0335 \approx 5{,}93\,\mathrm{GB}$$

Conclusión: Con 15 procesos, el uso estimado de memoria es de 5.93 GB.

Ejercicio 3: Latencia en redes de transmisión

Un sistema de monitoreo registra la latencia promedio (en ms) en función de la distancia de transmisión:

$$(50, 12{,}1), \quad (100, 18{,}7), \quad (200, 30{,}4)$$

Diferencias:

$$f[x_0, x_1] = \frac{18{,}7 - 12{,}1}{100 - 50} = 0{,}132$$
$$f[x_1, x_2] = \frac{30{,}4 - 18{,}7}{200 - 100} = 0{,}117$$
$$f[x_0, x_1, x_2] = \frac{0{,}117 - 0{,}132}{200 - 50} = -0{,}0001$$

Polinomio:

$$P(x) = 12{,}1 + 0{,}132(x - 50) - 0{,}0001(x - 50)(x - 100)$$

Para 150 km:

$$P(150) = 12{,}1 + 0{,}132(100) - 0{,}0001(100)(50) = 12{,}1 + 13{,}2 - 0{,}5 = 24{,}8\,\mathrm{ms}$$

Conclusión: La latencia estimada a 150 km es de 24.8 milisegundos.

4. Forma Estándar para Tres Puntos

4.1. Planteamiento del Problema

Dados tres puntos distintos $(x_0, y_0), (x_1, y_1), (x_2, y_2)$, se busca un polinomio cuadrático de la forma:

$$P(x) = ax^2 + bx + c$$

que pase exactamente por los tres puntos. Esto conduce al sistema lineal:

$$\begin{bmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

La resolución de este sistema proporciona los coeficientes a, b, c del polinomio cuadrático interpolador.

4.2. Casos Prácticos

Ejercicio 1: Throughput en función de usuarios concurrentes

Se registran medidas de throughput (peticiones/s) según el número de usuarios:

$$(10, 120), \quad (30, 280), \quad (50, 360)$$

Sistema a resolver:

$$\begin{cases} 100a + 10b + c = 120 \\ 900a + 30b + c = 280 \\ 2500a + 50b + c = 360 \end{cases}$$

Restando la primera ecuación de la segunda y la segunda de la tercera:

$$800a + 20b = 160 \quad \Rightarrow \quad 40a + b = 8$$
$$1600a + 20b = 80 \quad \Rightarrow \quad 80a + b = 4$$

Restando ambas ecuaciones:

$$-40a = 4 \quad \Rightarrow \quad a = -0{,}1$$

Sustituyendo:

$$b = 8 - 40(-0{,}1) = 12, \quad c = 120 - 100(-0{,}1) - 10(12) = 10$$

Polinomio resultante:

$$P(x) = -0{,}1x^2 + 12x + 10$$

Para 40 usuarios:

$$P(40) = -0{,}1(1600) + 12(40) + 10 = -160 + 480 + 10 = 330$$

Conclusión: Se estima un throughput de 330 peticiones/s con 40 usuarios concurrentes.

Ejercicio 2: Función de pérdida durante el entrenamiento

Valores de pérdida registrados en diferentes épocas:

$$(1, 0{,}60), \quad (5, 0{,}20), \quad (10, 0{,}08)$$

Sistema:

$$\begin{cases} a + b + c = 0{,}60 \\ 25a + 5b + c = 0{,}20 \\ 100a + 10b + c = 0{,}08 \end{cases}$$

Procediendo análogamente:

$$\begin{aligned} 24a + 4b = -0{,}40 \quad &\Rightarrow \quad 6a + b = -0{,}10 \\ 75a + 5b = -0{,}12 \quad &\Rightarrow \quad 15a + b = -0{,}024 \end{aligned}$$

Restando:

$$-9a = -0{,}076 \quad \Rightarrow \quad a \approx 0{,}00844$$

$$b \approx -0{,}1507, \quad c \approx 0{,}7422$$

Polinomio:

$$P(x) \approx 0{,}00844x^2 - 0{,}1507x + 0{,}7422$$

Para la época 7:

$$P(7) \approx 0{,}00844(49) - 0{,}1507(7) + 0{,}7422 \approx 0{,}101$$

Conclusión: El valor de pérdida estimado en la época 7 es aproximadamente 0.101.

Ejercicio 3: Consumo energético según frecuencia de CPU

Medidas de consumo (W) frente a frecuencia (GHz):

$$(1{,}0, 30), \quad (1{,}5, 50), \quad (2{,}0, 80)$$

Sistema:

$$\begin{cases} a + b + c = 30 \\ 2{,}25a + 1{,}5b + c = 50 \\ 4a + 2b + c = 80 \end{cases}$$

Resolviendo:

$$\begin{aligned} 1{,}25a + 0{,}5b = 20 \\ 1{,}75a + 0{,}5b = 30 \end{aligned}$$

$$0{,}5a = 10 \quad \Rightarrow \quad a = 20$$

$$b = -10, \quad c = 20$$

Polinomio:

$$P(x) = 20x^2 - 10x + 20$$

Para 1.8 GHz:

$$P(1{,}8) = 20(3{,}24) - 10(1{,}8) + 20 = 64{,}8 - 18 + 20 = 66{,}8\,\mathrm{W}$$

Conclusión: El consumo estimado a 1.8 GHz es de 66.8 W.

5. Splines Cúbicos

5.1. Definición General

Un **spline cúbico** es una función continua y suavemente diferenciable que aproxima un conjunto de puntos mediante polinomios de tercer grado definidos por tramos. En cada intervalo $[x_i, x_{i+1}]$ se define:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

donde los coeficientes a_i, b_i, c_i, d_i se determinan de forma que el conjunto total de segmentos sea suave, continuo y ajuste los datos proporcionados.

5.2. Condiciones de Continuidad

Para garantizar que el spline sea suave, se imponen las siguientes condiciones en cada punto interior x_i:

$$S_i(x_i) = y_i \qquad \text{(pasa por cada punto)} \qquad (9)$$
$$S_{i-1}(x_i) = S_i(x_i) \qquad \text{(continuidad)} \qquad (10)$$
$$S'_{i-1}(x_i) = S'_i(x_i) \qquad \text{(primera derivada continua)} \qquad (11)$$
$$S''_{i-1}(x_i) = S''_i(x_i) \qquad \text{(segunda derivada continua)} \qquad (12)$$

5.3. Condiciones de Frontera Natural

Para los **splines cúbicos naturales**, se asume que las segundas derivadas en los extremos son nulas:

$$S''_0(x_0) = 0, \qquad S''_{n-1}(x_n) = 0$$

Estas condiciones evitan oscilaciones no deseadas en los bordes de la curva, haciendo que el spline adopte un comportamiento más suave y natural.

5.4. Ejercicio 1. Suavizado de Datos de Temperatura

Se tienen registros de temperatura promedio (°C) por hora en un laboratorio de cómputo:

$$(0, 18),\ (3, 20),\ (6, 23),\ (9, 22),\ (12, 19)$$

El objetivo es generar una curva suave que modele la evolución térmica sin saltos abruptos. El spline cúbico natural permite construir una función continua $S(x)$ tal que:

$$S(0) = 18, \quad S(3) = 20, \quad S(6) = 23, \quad S(9) = 22, \quad S(12) = 19$$

y cumpla $S''(0) = S''(12) = 0$.

En la práctica, los coeficientes se determinan resolviendo un sistema tridiagonal:

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)$$

donde $h_i = x_{i+1} - x_i$.

El resultado produce una curva que suaviza los puntos medidos, útil en calibración de sensores y control térmico en sistemas computacionales.

5.5. Ejercicio 2. Interpolación de Pérdida en Entrenamiento de Red Neuronal

Durante el entrenamiento de una red neuronal, se registran pérdidas (*loss*) en distintas épocas:

$$(0, 0{,}8),\ (5, 0{,}4),\ (10, 0{,}25),\ (15, 0{,}2)$$

Se desea estimar el valor intermedio en la época $x = 8$ mediante spline cúbico natural.

Procedimiento conceptual:

- Se calculan las diferencias $h_i = x_{i+1} - x_i$: todos valen 5.
- Se construye la matriz tridiagonal de coeficientes c_i con las condiciones naturales $c_0 = c_3 = 0$.
- Se resuelve para c_1 y c_2, obteniendo valores que permiten determinar a_i, b_i, d_i.
- Finalmente, se evalúa $S_1(8)$, el tramo correspondiente a $[5, 10]$.

El resultado aproximado es:

$$S(8) \approx 0{,}304$$

lo cual muestra una disminución progresiva de la pérdida, coherente con un aprendizaje estable.

5.6. Ejercicio 3. Interpolación del Tiempo de Respuesta de un Servidor

Un servidor registra el tiempo medio de respuesta (ms) según la carga de usuarios:

$$(100, 120),\ (200, 135),\ (300, 180),\ (400, 250)$$

Se desea estimar el tiempo de respuesta para 250 usuarios mediante spline cúbico natural.

Resolviendo el sistema tridiagonal:

$$\begin{aligned}
100c_0 + 400c_1 + 100c_2 &= 3\left(\frac{180-135}{100} - \frac{135-120}{100}\right) = 3(0{,}45 - 0{,}15) = 0{,}9 \\
100c_1 + 400c_2 + 100c_3 &= 3\left(\frac{250-180}{100} - \frac{180-135}{100}\right) = 3(0{,}70 - 0{,}45) = 0{,}75
\end{aligned}$$

Con $c_0 = c_3 = 0$, el sistema se reduce a:

$$400c_1 + 100c_2 = 0{,}9$$
$$100c_1 + 400c_2 = 0{,}75$$

Multiplicando la primera ecuación por 4:

$$1600c_1 + 400c_2 = 3{,}6$$

Restando la segunda ecuación:

$$1500c_1 = 2{,}85 \quad \Rightarrow \quad c_1 = 0{,}0019$$

Sustituyendo:

$$c_2 = \frac{0{,}75 - 100(0{,}0019)}{400} = \frac{0{,}56}{400} = 0{,}0014$$

Para el segmento $[200, 300]$, los coeficientes son:

$$\begin{aligned}
a_1 &= 135 \\
b_1 &= \frac{180 - 135}{100} - \frac{100}{3}(2(0{,}0019) + 0{,}0014) = 0{,}45 - 0{,}1733 = 0{,}2767 \\
d_1 &= \frac{0{,}0014 - 0{,}0019}{3(100)} = -0{,}0000017
\end{aligned}$$

Evaluando en $x = 250$ (donde $x - 200 = 50$):

$$\begin{aligned}
S_1(250) &= 135 + 0{,}2767(50) + 0{,}0019(50)^2 - 0{,}0000017(50)^3 \\
&= 135 + 13{,}835 + 4{,}75 - 0{,}2125 \\
&\approx 153{,}4\,\text{ms}
\end{aligned}$$

Respuesta: El tiempo de respuesta estimado para 250 usuarios es de aproximadamente 153.4 ms.

5.7. Conclusión

Los splines cúbicos son esenciales tanto en estadística como en informática aplicada, ya que permiten construir curvas suaves a partir de datos discretos sin generar errores de extrapolación grandes. Se utilizan ampliamente en:

6. Error de Interpolación

6.1. Definición General

En cualquier método de interpolación, el valor obtenido mediante el polinomio interpolador $P_n(x)$ no coincide exactamente con la función original $f(x)$, generándose un error de aproximación definido como:

$$E(x) = f(x) - P_n(x)$$

Este error permite medir qué tan precisa es la interpolación realizada.

6.2. Cota del Error

El error de interpolación para un polinomio de grado n está acotado por la siguiente expresión:

$$|f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \prod_{i=0}^{n} |x - x_i|$$

donde:

- M_{n+1} es el valor máximo de la derivada de orden $(n+1)$ de la función $f(x)$ en el intervalo considerado.
- $x_0, x_1, \ldots, x_n$ son los puntos de interpolación.

6.3. Interpretación

La cota del error muestra que:

- El error disminuye cuando los puntos x_i están más próximos entre sí.
- Usar polinomios de mayor grado puede reducir el error localmente, pero también puede causar oscilaciones si se usan demasiados puntos (fenómeno de Runge).
- En la práctica, los **splines cúbicos** suelen ofrecer un equilibrio ideal entre precisión y estabilidad numérica.

6.4. Aplicación

El análisis del error permite decidir qué método es más adecuado:

- **Interpolación lineal**: rápida y simple, con error proporcional a $(x - x_i)^2$.
- **Interpolación de Lagrange o cuadrática**: mayor precisión en zonas suaves.
- **Splines cúbicos**: menor error global y continuidad hasta la segunda derivada.

Conclusión: El estudio del error de interpolación es esencial para evaluar la calidad de las aproximaciones numéricas y seleccionar el método más eficiente en problemas de ingeniería estadística e informática.