

CLASE

5

Métodos Numéricos

Interpolación de Funciones

Educación de calidad

INTRO DUCCIÓN

DE LA CLASE

GRADO / POSGRADO / TECNOLOGÍAS

Estudia a tu tiempo
son interrupciones



El estudio de las ecuaciones diferenciales constituye un pilar fundamental del análisis matemático aplicado, ya que permite describir el comportamiento dinámico de sistemas en áreas como la física, la biología, la economía y la ingeniería, partiendo de su clasificación según orden, grado y estructura para establecer un enfoque claro tanto en el análisis como en la solución de problemas, considerando elementos clave como las condiciones iniciales y de frontera, la identificación de variables y los teoremas de existencia y unicidad que garantizan la validez de los modelos, lo que facilita una comprensión sólida de su papel en la formulación de problemas reales, para luego centrarse en ecuaciones de primer orden, abordando distintos tipos y métodos de resolución, incluyendo soluciones explícitas e implícitas y herramientas gráficas como campos direccionales, junto con técnicas para ecuaciones separables, homogéneas, exactas y lineales, además de casos especiales como Bernoulli y Riccati, cerrando con aplicaciones en modelos de crecimiento, circuitos eléctricos y procesos físicos que permiten aplicar estos conceptos en contextos reales.

Además, se estudia el proceso de aproximación y ajuste de curvas como herramienta clave en el análisis de datos y la modelación de fenómenos, diferenciando entre interpolación, que busca pasar exactamente por los datos, y aproximación, que intenta representar la tendencia general permitiendo cierto error, lo cual es útil cuando existen datos con ruido, para luego desarrollar el método de mínimos cuadrados como técnica que minimiza el error entre valores observados y estimados, analizando su formulación y su relación con sistemas lineales, y abordando tanto el ajuste lineal como el no lineal con funciones más complejas, incluyendo el análisis de la calidad del ajuste y la interpretación de los residuos, con el objetivo de que el estudiante pueda modelar datos de manera adecuada y tomar decisiones basadas en resultados numéricos.

Interpolación de Funciones

En esta asignatura hemos cubierto métodos para resolver ecuaciones y sistemas lineales.

Ahora abordamos la interpolación de funciones, una técnica fundamental para aproximar funciones desconocidas o costosas de evaluar a partir de un conjunto finito de puntos de datos (x_i, y_i) , donde $y_i = f(x_i)$.

La interpolación de funciones es un proceso matemático que se utiliza para estimar valores desconocidos dentro del dominio de un conjunto de datos discretos, es decir, cuando solo se tiene información en ciertos puntos específicos. Este proceso es fundamental en diversas áreas de las ciencias aplicadas y la ingeniería, ya que permite construir funciones continuas que pasan exactamente por un conjunto de puntos dados

. Dentro de los métodos de interpolación polinómica, el polinomio de Lagrange es uno de los más utilizados. Este polinomio se construye de manera que para cada uno de los puntos conocidos, se crea un término que es cero en todos los puntos, excepto en el punto correspondiente.

Concepto de interpolación

Dado un conjunto de $n+1$ puntos distintos $\{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}$ con $x_i \neq x_j$ para $i \neq j$, la interpolación consiste en encontrar una función $P(x)$ tal que:

$$P(x_i) = y_i \text{ para todo } i = 0, 1, \dots, n$$

y que pertenezca a una familia parametrizada de funciones (polinomios, splines, funciones racionales, etc.).

Teorema de existencia y unicidad (interpolación polinómica): Para $n+1$ puntos distintos existen un único polinomio $P(x)$ de grado $\leq n$ tal que $P(x_i) = y_i$ para todo i . Este polinomio se denomina polinomio interpolador de Lagrange o de Newton (ambos representan el mismo polinomio, pero expresado de forma diferente).

Error de interpolación (teorema fundamental): Sea $f \in C^{n+1}[a, b]$ y $x_0, \dots, x_n \in [a, b]$ distintos. Entonces para cualquier $x \in [a, b]$ existe $\xi \in (\min x_0, \dots, x_n, \max x_0, \dots, x_n)$ tal que

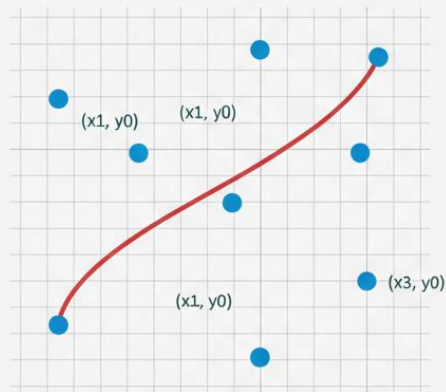
$$f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x)$$

donde $\omega(x) = (x - x_0)(x - x_1) \dots (x - x_n)$ es el polinomio nodal.

POLYNOMIAL INTERPOLATION: Unique Polynomial Through N+1 Points

Visualizing Concept of a Unique Interpolating Polynomial

DATA POINTS & INTERPOLATION



✓ Unique polynomial exists for $n+1 \leq 4$ points

KEY CONCEPTS

- Given $n+1$ data points (x_i, y_i) ...
- There exists a unique polynomial $P(x)$...
- The degree of $P(x)$ is $\leq n$.
- Lagrange & Newton forms are common methods.



KEY CONCLUSION: For any set of distinct data points, there is one and only one polynomial of degree $\leq n$ that passes through all of them. This uniqueness is fundamental to polynomial interpolation.

Figura 1: Conjunto de puntos de datos y polinomio interpolador único de grado $\leq n$ para $n+1$ puntos.

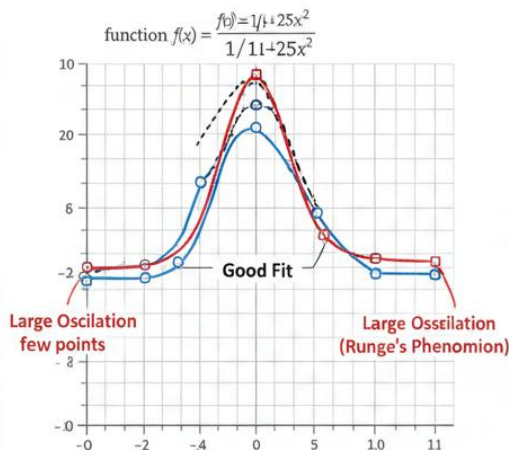
Creación propia.

En ciberseguridad: la interpolación se usa en side-channel attacks (reconstrucción de funciones de leakage mediante polinomios a partir de trazas de potencia/timing), en criptoanálisis polinómico (aproximación de funciones modulares o curvas elípticas), en machine learning para seguridad (interpolación de datos de entrenamiento para detección de anomalías) y en procesamiento forense digital (suavizado/interpolación de señales en imágenes o audio extraídos de dispositivos comprometidos).

POLYNOMIAL INTERPOLATION: Error & Runge's Phenomenon

Visualizing Interpolation Error and Oscillation Over Large Intervals

CONVERGENCE & ERROR VISUALIZATION



KEY CONCEPTS

- **Interpolation Error:** $R(x) = f(x) - P_n(x)$
- **Runge's Phenomenon:** High-degree polynomial with uniformly spaced points can diverge near interval boundaries.
- **Oscillation:** Error grows as interval size increases, especially with uniform nodes.
- **Solution:** Use Chebyshev nodes or spline interpolation



KEY CONCLUSION: While more points generally improve accuracy, uniform can lead to instability and oscillations in polynomial interpolation, necessitating alternative methods for large intervals.

Figura 2: Representación del error de interpolación: término de Runge y fenómeno de oscilación en intervalos grandes.

Creación propia.

El término de Runge es un ejemplo clásico de un error que ocurre cuando se utiliza un polinomio de grado alto para interpolar una función, particularmente en intervalos grandes. En este caso, el error de interpolación crece significativamente en los extremos del intervalo. Este fenómeno es más notable cuando la función a interpolar tiene un comportamiento oscilante o no es suficientemente suave, como sucede con la famosa función de Runge:

$$f(x) = \frac{1}{1 + 25x^2}$$

Cuando se interpola esta función usando polinomios de alto grado, el error no solo se mantiene pequeño en el centro del intervalo, sino que aumenta drásticamente cerca de los bordes, produciendo oscilaciones que se conocen como oscilaciones de Runge.

El fenómeno de oscilación ocurre porque los polinomios de grado alto tienden a "sobrefit" los datos, es decir, ajustan los puntos de datos de manera demasiado precisa, lo que genera

oscilaciones no deseadas, especialmente en los extremos del intervalo. A medida que el grado del polinomio aumenta, las oscilaciones cerca de los bordes del intervalo también se intensifican, lo que resulta en una mala aproximación, a pesar de que el polinomio pase por todos los puntos de datos. Esta oscilación se puede ver en el gráfico como picos y valles en la curva interpolada en las cercanías de los extremos del intervalo.

Interpolación polinómica

La interpolación polinómica busca el polinomio $P_n(x)$ de grado $\leq n$ que pasa exactamente por los $n+1$ puntos dados.

Ventajas matemáticas:

- Unicidad garantizada.
- Error controlado si se conoce $f^{(n+1)}$.
- Fácil evaluación en puntos nuevos.

Desventajas:

- Fenómeno de Runge: para nodos equidistantes y n grande, oscilaciones grandes cerca de los extremos (polinomios de alto grado son inestables).
- Sensibilidad a errores en los datos (mal condicionado para nodos equidistantes).

Nodos de Chebyshev (para minimizar error máximo): $x_k = \cos \left(\frac{(2k+1)\pi}{2(n+1)} \right)$, $k = 0, \dots, n$ reducen significativamente el fenómeno de Runge.

En ciberseguridad: interpolación polinómica de bajo grado se usa en ataques a implementaciones de curvas elípticas (aproximación de la función de punto en curva), y en side-channel para modelar leakage como polinomio de grado bajo (e.g., Hamming weight model).

RUNGE'S PHENOMENON: Improving High-Degree Polynomial Interpolation Nodes

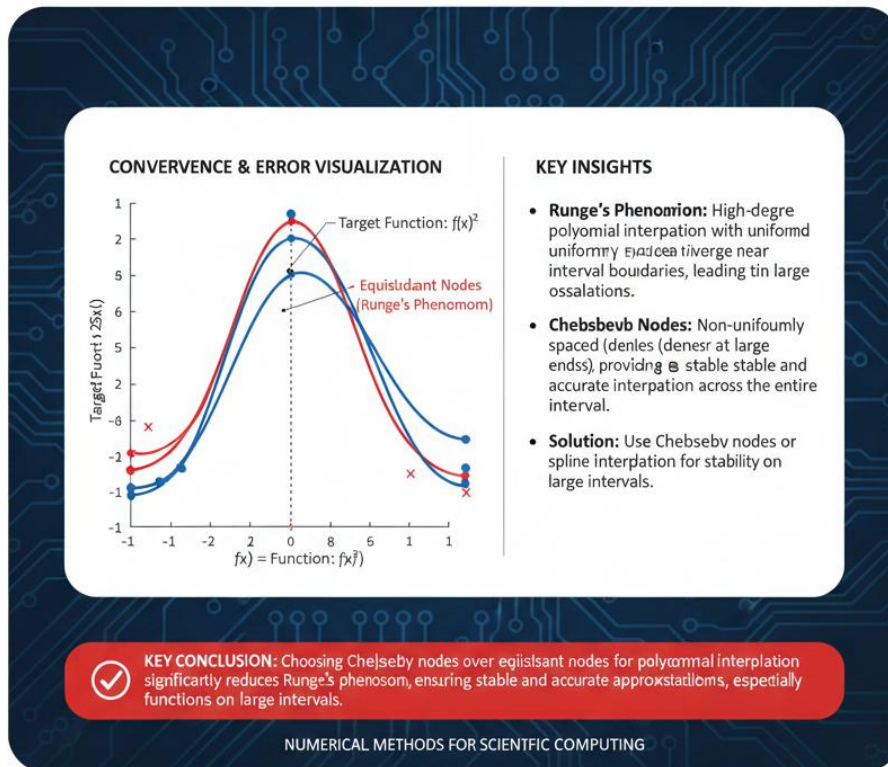


Figura 3: Fenómeno de Runge: interpolación polinómica de grado alto con nodos equidistantes vs nodos de Chebyshev.

Creación propia.

Polinomio de Lagrange

Formulación matemática formal

El polinomio de Lagrange es:

$$P_n(x) = \sum_{i=0}^n y_i l_i(x)$$

donde los **polinomios base de Lagrange** son:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

Propiedades:

- $l_i(x_k) = \delta_{ik}$ (1 si $i=k$, 0 en otro caso) $\rightarrow$ interpolación exacta.
- Grado de cada l_i es $n \rightarrow P_n$ tiene grado $\leq n$.
- Forma explícita, no requiere resolución de sistemas lineales.

Complejidad: $O(n^2)$ para evaluar en un punto nuevo (suma de productos).

Estabilidad numérica: mal condicionado para n grande (número de condición crece exponencialmente con n en nodos equidistantes).

En ciberseguridad: forma de Lagrange útil para evaluar rápidamente funciones interpoladas en ataques de timing o power (e.g., aproximar tiempo de ejecución como función de clave parcial).

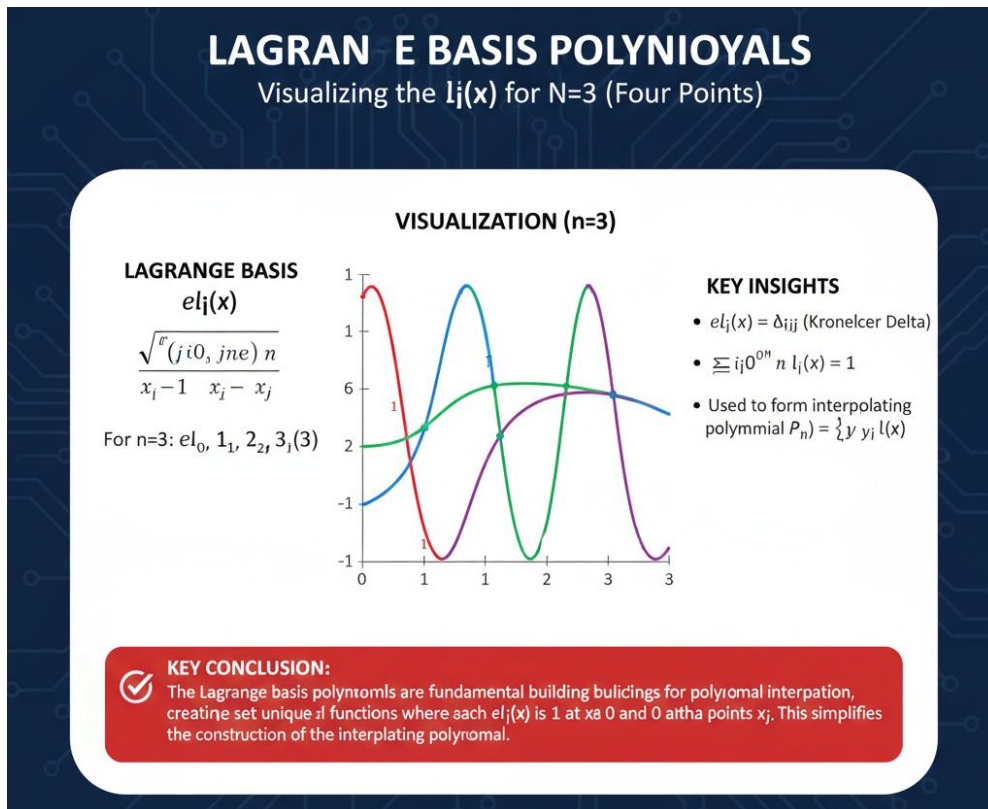


Figura 4: Polinomios base de Lagrange $l_i(x)$ para $n = 3$ (cuatro puntos).

Creación propia.

La figura muestra los polinomios base de Lagrange ($l_i(x)$) para ($n = 3$), es decir, para un conjunto de 4 puntos. Los polinomios base de Lagrange son utilizados en la interpolación polinómica para construir el polinomio interpolador. Para ($n = 3$), hay 4 puntos de datos, y cada uno de estos puntos está asociado con un polinomio base ($l_i(x)$) que tiene la propiedad de que ($l_i(x_j) = 1$) cuando ($i = j$) y ($l_i(x_j) = 0$) para ($i \neq j$). Esto significa que cada ($l_i(x)$) toma el valor 1 en el punto (x_i) y 0 en los demás puntos.

En la figura, probablemente se muestran las curvas de los 4 polinomios base ($l_0(x)$), ($l_1(x)$), ($l_2(x)$) y ($l_3(x)$), cada uno de los cuales es un polinomio de grado 3 (ya que hay 4 puntos en total). Estas curvas se intersectan en los puntos de interpolación, y cada uno de ellos tiene su

máximo en su respectivo nodo (x_i), donde toma el valor 1, mientras que se acercan a cero en los otros nodos.

Cada polinomio base ($l_i(x)$) contribuye a la construcción del polinomio interpolador, de manera que el polinomio total se puede escribir como una suma ponderada de estos polinomios base:

$$P(x) = \sum_{i=0}^3 y_i l_i(x)$$

donde (y_i) son los valores de la función en los puntos (x_i). En la figura, se puede observar cómo cada polinomio base es responsable de "activar" el valor (y_i) correspondiente en su respectivo punto (x_i), mientras que los demás puntos son anulados (por eso los polinomios base toman valor 0 en esos puntos).

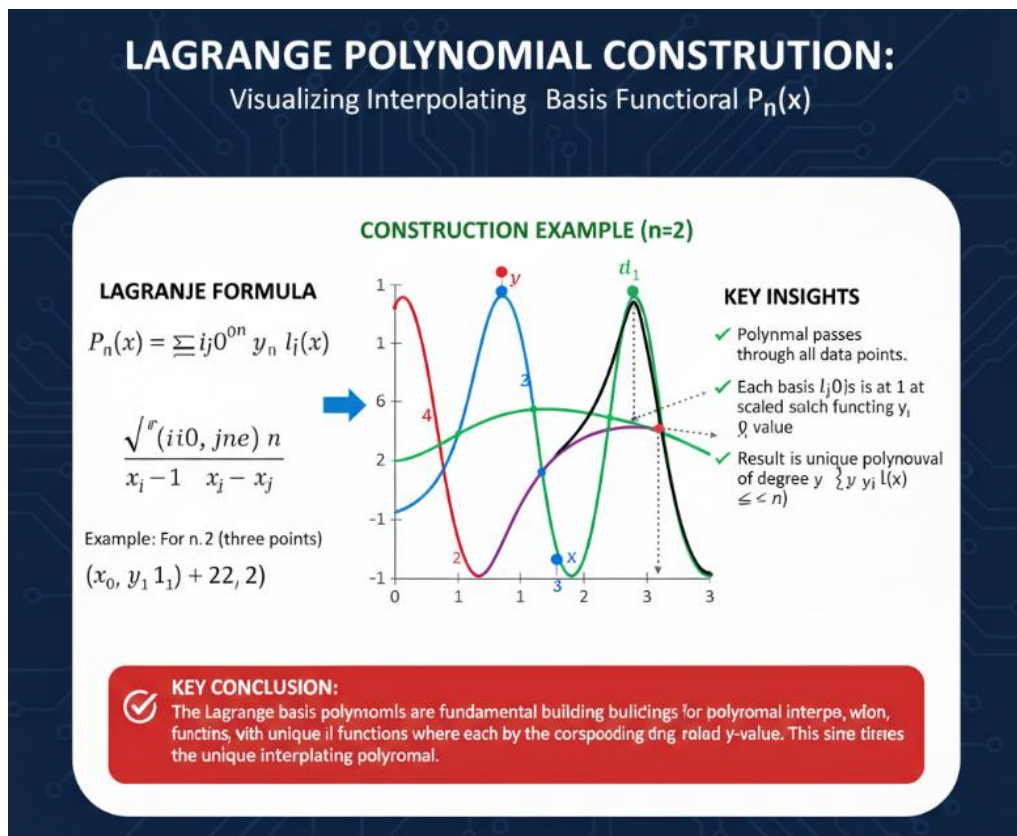


Figura 5: Construcción del polinomio de Lagrange: suma ponderada de bases.

Creación propia.

Polinomio de Newton

Formulación matemática formal

El polinomio de Newton usa diferencias divididas y se escribe en forma anidada (Horner-like):

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n(x - x_0) \dots (x - x_{n-1})$$

donde los coeficientes a_k son las **diferencias divididas**:

- $a_0 = f[x_0] = y_0$
- $a_1 = f[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$
- $a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$
- En general: $f[x_0, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0}$

Ventajas:

- Forma anidada $\rightarrow$ evaluación $O(n)$.
- Fácil añadir puntos nuevos (diferencias divididas adicionales).
- Estabilidad numérica mejor que Lagrange para evaluación secuencial.

Teorema: El polinomio de Newton coincide con el de Lagrange (mismo polinomio único).

Enlace 1: https://en.wikipedia.org/wiki/Newton_polynomial Ver en Wikipedia: Newton polynomial Recurso con diferencias divididas, forma anidada y comparación con Lagrange.

Aproximación y Ajuste de Curvas

En la clase anterior vimos la interpolación, donde una función pasa exactamente por todos los puntos dados. Ahora exploramos la aproximación (o ajuste de curvas), que busca la mejor representación posible cuando los datos contienen ruido, errores de medición o no se espera un ajuste perfecto.

La aproximación y el ajuste de curvas son técnicas utilizadas para encontrar una función que se acerque a un conjunto de datos, pero a diferencia de la interpolación, que busca una función que pase exactamente por todos los puntos, la aproximación busca una función que minimice el error global, es decir, que se acerque lo más posible a todos los puntos sin necesariamente pasar por cada uno de ellos. Mientras que la interpolación es útil cuando se desea una representación exacta de los datos, la aproximación es más adecuada cuando se trabaja con datos ruidosos o cuando se desea una solución más general que no dependa estrictamente de cada punto individual. Un método comúnmente utilizado para la aproximación de curvas es el método de mínimos cuadrados, que tiene como objetivo encontrar la mejor aproximación en

el sentido de minimizar la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos por la función ajustada. Este método se utiliza ampliamente en regresión y es particularmente eficaz cuando los datos presentan ruido o incertidumbre, ya que proporciona un compromiso entre los datos y una función general. El método de mínimos cuadrados se puede aplicar tanto a ajustes lineales como no lineales, dependiendo de la naturaleza de los datos y de la función que se quiera ajustar.

Diferencia entre interpolación y aproximación

Interpolación: Dada un conjunto finito de puntos $\{(x_i, y_i)\}_{i=0}^n$, se busca una función $P(x)$ tal que $P(x_i) = y_i$ exactamente para todo i (error cero en los nodos). El polinomio interpolador único de grado $\leq n$ existe por el teorema de unicidad. El error en puntos intermedios está dado por el teorema de interpolación:

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n)$$

para algún ξ entre $\min(x, x_i)$ y $\max(x, x_i)$.

Aproximación (curve fitting): Cuando los datos y_i contienen ruido (e.g., errores de medición, variabilidad aleatoria), no se busca pasar exactamente por todos los puntos, sino minimizar una medida global del error. El objetivo es encontrar una función $g(x)$ (de una familia parametrizada, e.g., polinomios de grado bajo, exponenciales) que represente la tendencia general o el comportamiento subyacente.

Diferencia clave:

Interpolación $\rightarrow$ error cero en puntos dados; útil para datos precisos sin ruido.

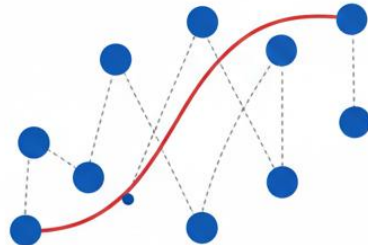
Aproximación $\rightarrow$ minimiza error global (e.g., suma de cuadrados de residuos); útil para datos ruidosos o sobreajuste a evitar.

Teorema de aproximación (Weierstrass): Cualquier función continua en un intervalo cerrado puede aproximarse uniformemente por polinomios de grado arbitrariamente alto (pero en práctica, grado alto causa inestabilidad numérica).

INTERPOLATION VS BEST FIT APPROXIMATION: Exact vs Global Error Minimization

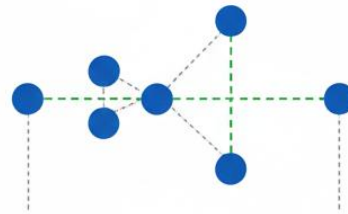
Visualizing Polynomial

EXACT INTERPLTATION



- Passes through ALL data points (Zero Errors)
- High-degree polynomial
- Example: Lagrange, Newton

BEST-FIT APPROMIITATION (Least Squares)



- Minimizes TOTAL error (e.g. sum of sum of of squares)
- Lower-degree polynomial (Doe'sn't pass Smoother)

Complexity ↑, Flexibility ↑, Noise Sensitivity ↑



KEY CONCLUSION: Interpolon's method yeilds curve curve passepes exactly exactly all given points, wauisy.inhiple given points, while best-fit approxizatiion on corte-edsse error across other awly offer noise alvatages is structure.

Figura 6: Diferencia gráfica entre interpolación exacta (pasa por todos los puntos) y aproximación por mejor ajuste (best fit, minimiza error global).

Creación propia.

Método de mínimos cuadrados

Dado un conjunto de m puntos $\{(x_i, y_i)\}_{i=1}^m$ ($m > n + 1$ usualmente) y una familia de funciones lineales en parámetros:

$$y \approx f(x; \beta) = \beta_0 \varphi_0(x) + \beta_1 \varphi_1(x) + \dots + \beta_n \varphi_n(x)$$

donde $\{\varphi_j(x)\}$ es una base (e.g., $1, x, x^2, \dots$ para polinomios; o exponenciales, logaritmos, etc. para no lineal).

El método de mínimos cuadrados encuentra el vector de parámetros β^* que minimiza la suma de cuadrados de residuos (SSR o SSE):

$$S(\beta) = \sum_{i=1}^m r_i^2 = \sum_{i=1}^m [y_i - f(x_i; \beta)]^2$$

Teorema fundamental (condición necesaria):

Si $S(\beta)$ es diferenciable, el mínimo satisface las ecuaciones normales:

$$\partial S / \partial \beta_j = 0 \text{ para } j = 0, \dots, n$$

$$\Rightarrow -2 \sum_{i=1}^m [y_i - f(x_i; \beta)] \partial f / \partial \beta_j(x_i) = 0$$

Para ajuste lineal en parámetros (f lineal en β), esto da un sistema lineal:

$$A^T A \beta = A^T y$$

donde A es la matriz de diseño ($A_{ij} = \varphi_j(x_i)$), y el vector $y = [y_1, \dots, y_m]^T$.

Solución: $\beta = (A^T A)^{-1} A^T y$ (si $A^T A$ invertible).

Error residual: $SSE = \sum (y_i - \hat{y}_i)^2$; $R^2 = 1 - SSE/SST$ (SST = suma total de cuadrados respecto a la media).

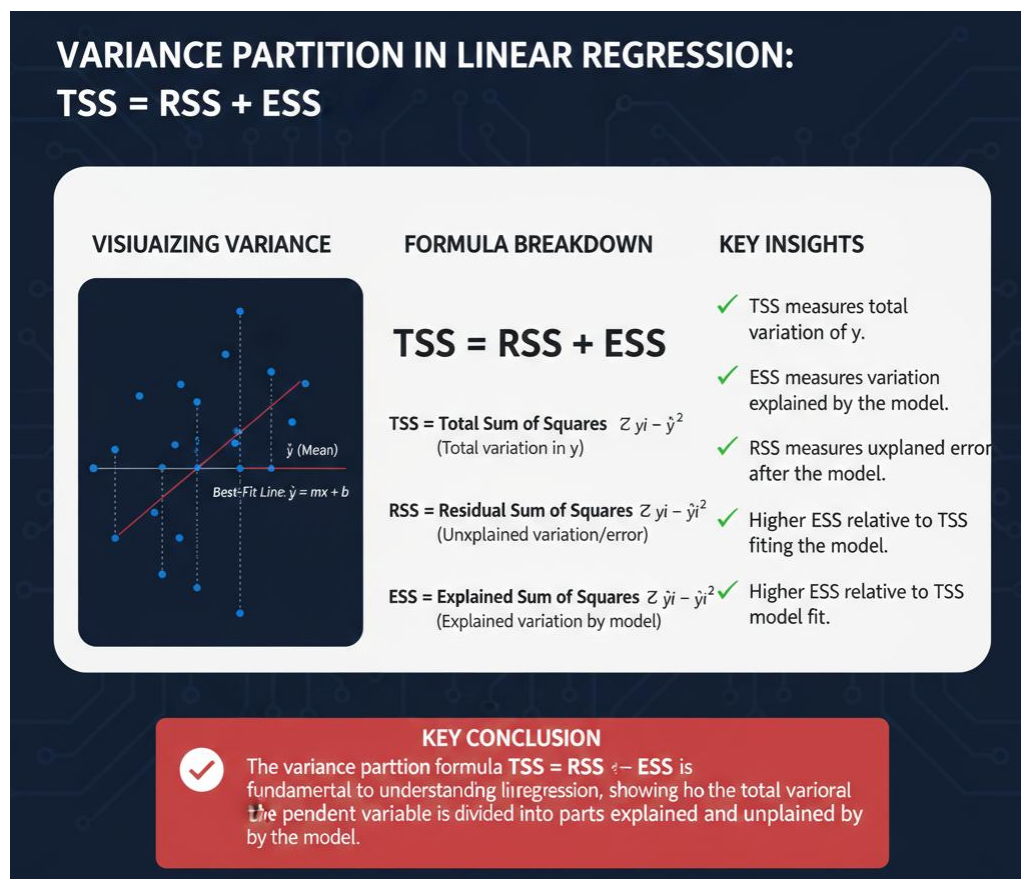


Figura 7: Partición de varianza: TSS = RSS + ESS en regresión lineal.

Creación propia.

La figura ilustra la partición de la varianza en el contexto de la regresión lineal y muestra cómo la varianza total de los datos (denotada como (TSS)) se descompone en dos componentes: la varianza explicada por el modelo (denotada como (ESS)) y la varianza no explicada, o residuos, (denotada como (RSS)).

- TSS (Total Sum of Squares): Es la suma total de los cuadrados de las diferencias entre los valores observados y el valor medio de la variable dependiente (y). Representa la varianza total en los datos, o cuán dispersos están los puntos de datos en relación con la media. Se calcula como:

$$TSS = \sum_{i=1}^n (y_i - \bar{y})^2$$

donde (y_i) son los valores observados y ($\bar{y}$) es la media de (y).

- RSS (Residual Sum of Squares): Es la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos por el modelo de regresión. Representa la varianza que no es explicada por el modelo, es decir, la parte del error residual que no puede ser capturada por la línea de regresión. Se calcula como:

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

donde ($\hat{y}_i$) son los valores predichos por el modelo.

- ESS (Explained Sum of Squares): Es la suma de los cuadrados de las diferencias entre los valores predichos por el modelo y la media de los valores observados. Representa la varianza que el modelo ha explicado o capturado. Se calcula como:

$$ESS = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

En la figura, estos tres componentes se muestran de manera que (TSS) se descompone en (ESS) y (RSS). La relación es:

$$TSS = ESS + RSS$$

Esto muestra cómo la varianza total en los datos se reparte entre la parte explicada por el modelo y la parte no explicada (los errores residuales). El objetivo de la regresión lineal es maximizar (ESS) y minimizar (RSS), logrando así un buen ajuste del modelo a los datos.

Ajuste lineal

Caso más simple: ajuste lineal (grado 1):

$$y \approx \beta_0 + \beta_1 x$$

Matriz de diseño A ($m \times 2$):

$$\begin{matrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{matrix}$$

Ecuaciones normales:

$$A^T A \beta = A^T y$$

$$m \sum x_i \sum x_i \sum x_i^2 \beta_0 \beta_1 = \sum y_i \sum x_i y_i$$

Solución explícita:

$$\beta_1 = \frac{m \sum x_i y_i - (\sum x_i)(\sum y_i)}{m \sum x_i^2 - (\sum x_i)^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

Teorema de Gauss-Markov (para errores independientes, media cero, varianza constante): el estimador de mínimos cuadrados es el mejor lineal insesgado (BLUE).

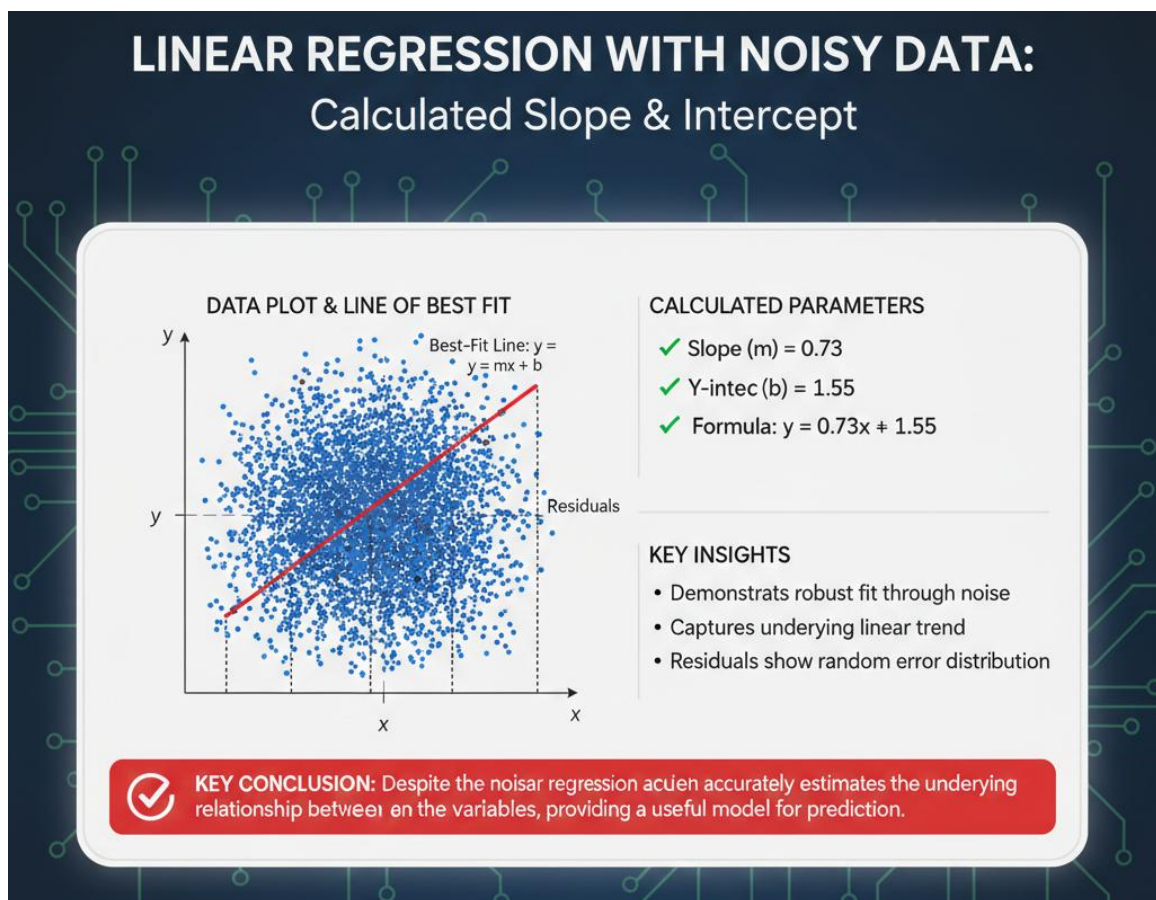


Figura 8: Ejemplo de regresión lineal con datos ruidosos: pendiente y ordenada calculadas.

Creación propia.

Ajuste no lineal

Cuando el modelo es no lineal en parámetros:

$$y \approx f(x; \beta), \text{ con } \beta \text{ no lineal (e.g., } y \approx \beta_0 e^{\beta_1 x}, y \approx \beta_0 + \beta_1 \sin(\beta_2 x + \beta_3), \text{ etc.)}.$$

No hay solución cerrada; se minimiza $S(\beta) = \sum [y_i - f(x_i; \beta)]^2$ mediante métodos numéricos iterativos (gradiente descendente, Levenberg-Marquardt, Gauss-Newton).

Gauss-Newton: aproximación lineal local de f alrededor de $\beta^{(k)}$:

$$f(x; \beta) \approx f(x; \beta^{(k)}) + J(\beta^{(k)})(\beta - \beta^{(k)})$$

donde J es la matriz Jacobiana $(\partial f / \partial \beta)$ evaluada en $\beta^{(k)}$.

Iteración: $\beta^{(k+1)} = \beta^{(k)} + (J^T J)^{-1} J^T r^{(k)}$, donde $r^{(k)} = y - f(\beta^{(k)})$.

Levenberg-Marquardt: modifica Gauss-Newton añadiendo término de amortiguación: $(J^T J + \lambda I)^{-1}$ para estabilidad.

Convergencia: localmente cuadrática si Jacobiana tiene rango completo y buena inicial.

Enlace 2: https://en.wikipedia.org/wiki/Nonlinear_least_squares Ver en Wikipedia: Nonlinear least squares Recurso con Gauss-Newton, Levenberg-Marquardt y convergencia.

RE FE REN CIAS

Glosario:

- Burden, R. L., Faires, J. D., & Burden, A. M. (2015). Numerical analysis (10ª ed.). Cengage Learning.
- Higham, N. J. (2002). Accuracy and stability of numerical algorithms (2ª ed.). SIAM.
- Atkinson, K. E. (1989). An introduction to numerical analysis. Wiley.
- Paar, C., & Pelzl, J. (2010). Understanding cryptography: A textbook for students and practitioners. Springer.
- Boneh, D., & Shoup, V. (2020). A graduate course in applied cryptography.
<https://toc.cryptobook.us>



síguenos en:



www.pucevirtual.puce.edu.ec