

Clase 11: Regresión no lineal y algoritmos de aproximación

FM849 - Programación Científica para Proyectos de Inteligencia Artificial (IA)

12 de enero de 2026

Resumen de regresión lineal

Ya aprendimos sobre los modelos de regresión lineal para realizar predicciones sobre una variable continua (y), utilizando variables predictivas o regresores ($X = \{x_1, \dots, x_n\}$).

El modelo de regresión lineal nace de la formulación de las funciones lineales ($y = mx + b$) y su extensión a múltiples variables:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon = \beta_0 + \sum_{i=1}^n \beta_i x_i + \varepsilon$$

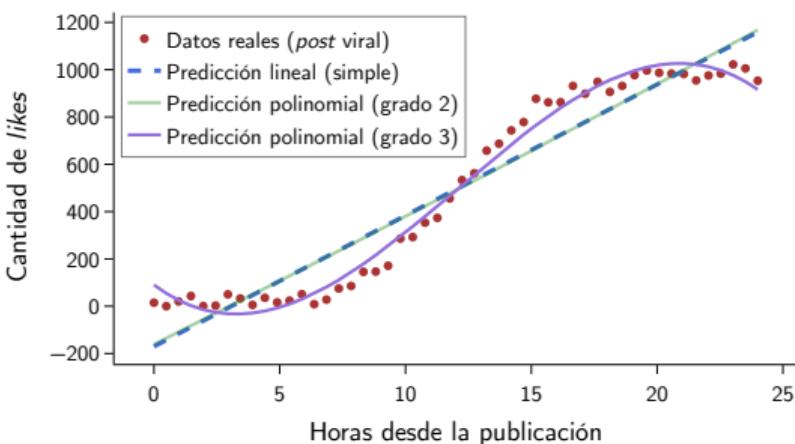
donde ε es el ruido o error aleatorio.

El objetivo de la regresión lineal es encontrar los coeficientes β_0, \dots, β_n que minimicen el error predictivo del modelo.

Limitaciones de la regresión lineal

Hay fenómenos que no pueden ser modelados adecuadamente con una regresión lineal debido a su naturaleza no lineal. En palabras simples, hay problemas donde no podemos trazar una “línea recta” para representar la relación entre la variable objetivo y los regresores.

Transición de modelos: predicción de *likes* en redes sociales



- ▶ ¿Realmente podemos usar una línea recta para modelar estos datos?
- ▶ ¿Por qué la regresión de grado 2 se ve como una recta, si debiese verse como una parábola?
- ▶ ¿Es conveniente probar con polinomios de grado mayor a 3?

Costos de un modelado deficiente

Pensemos que manejamos una empresa, y queremos invertir en anuncios para Instagram. Con datos históricos, observamos que al pasar de un pago en promoción de \$ 10.000/semana a \$ 50.000/semana, las ventas aumentan de manera lineal.

Por este motivo, ajustamos un modelo de regresión lineal que predice las ventas en función de variables asociadas a la inversión, para saber cuánto ganaríamos si invertimos 10 veces más (\$ 500.000/semana).

Resulta que el modelo predice un aumento de las ventas del 1000 %, pero en la realidad, las ventas sólo aumentan un 300 %, debido a distintos factores (p. ej., ya alcanzamos a todo el público objetivo, lo que se conoce como **saturación de audiencia**).

Como consecuencia, terminamos invirtiendo mucho dinero en publicidad sin obtener las ganancias esperadas. Imaginen esto a gran escala.

¿Cómo se identifica una regresión no lineal?

En una regresión lineal, la función de modelamiento $\hat{f}(X) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$ es lineal en los parámetros $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_n$. Este criterio no se cumple en los modelos de regresión **no lineal**, y veremos una aplicación.

Ejemplo: crecimiento bacteriano

El crecimiento de una población de bacterias puede modelarse con la función exponencial $f(t) = a \cdot e^{bt}$. Aquí, t representa el tiempo, a es la población inicial y b es la tasa de crecimiento.

Tenemos que entender que t y $f(t)$ son parámetros controlados del experimento, y queremos estimar los parámetros a y b . Podemos reescribir entonces $f(t) = \beta_0 \cdot e^{\beta_1 t}$. Esta función claramente no es lineal en los parámetros β_0 y β_1 , porque β_1 aparece “atrapado” dentro de la función exponencial.

¿Cómo se identifica una regresión no lineal?

El caso de las bacterias es sólo un ejemplo. Las regresiones no lineales también incluyen funciones como:

- ▶ Interacciones periódicas (ondas): $f(x) = \beta_0 \cdot \sin(\beta_1 x + \beta_2) + \beta_3$.
 - ▶ Puede modelar fenómenos periódicos, como las horas de sol en un día ($f(x)$) según un mes del año (x).
- ▶ Interacciones racionales: $f(x) = \frac{\beta_0 \cdot x}{\beta_1 + x}$.
 - ▶ Puede modelar fenómenos de saturación, como la energía corporal ($f(x)$) según las dosis de vitaminas (x).

Como siempre, hay muchos más ejemplos. Lo importante es entender la idea que acá los parámetros están “encerrados” en una función no lineal.

Dificultad inherente a la regresión no lineal

Computacionalmente, es fácil encontrar los coeficientes β_0, \dots, β_n en un modelo de regresión lineal, porque se desarrollaron cálculos matemáticos que permiten encontrarlos de manera directa mediante una fórmula (que no veremos en detalle, pero existe).

El problema de la regresión no lineal es que, dada la variedad de funciones posibles en las que pueden venir encerrados los parámetros, no existe ninguna fórmula para deducir sus valores.

¿Cómo se soluciona entonces este problema? Acá entran los algoritmos de aproximación.

Algoritmos de aproximación

Definición

Los algoritmos de aproximación corresponden a una clase de algoritmos que buscan obtener soluciones aproximadas para problemas donde encontrar una solución exacta es muy difícil, dada la estructura matemática que tienen.

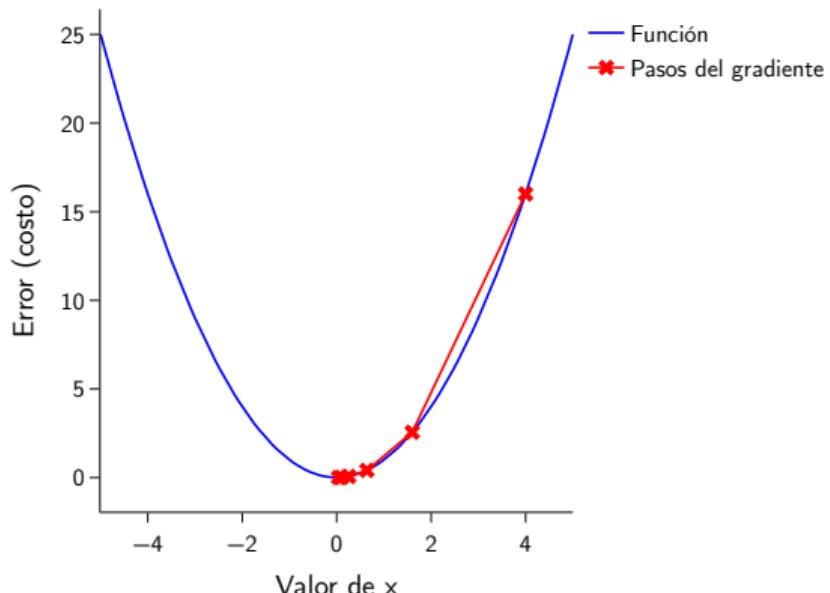
En este curso, veremos uno de los algoritmos más importantes en la era de la inteligencia artificial moderna, conocido como el método del gradiente descendente.

Método del gradiente descendente

Pensemos que estamos en una colina y queremos llegar al punto más bajo. El problema es que estamos vendados, y sólo podemos sentir la inclinación del terreno en el punto donde estamos parados.

El método del gradiente descendente nos dice que, para bajar la colina, debemos movernos en la dirección donde la inclinación es más pronunciada hacia abajo. Repetimos este proceso hasta que lleguemos al punto más bajo.

Simulación de gradiente descendente (5 iteraciones)



Método del gradiente descendente

En términos prácticos, recordemos que lo que queremos hacer al resolver un problema con regresión no lineal (¡también!) es minimizar una función que me indica el error predictivo del modelo (función de costo).

- ▶ Esta función de costo es como la “colina” del ejemplo anterior.
- ▶ El punto más bajo de la colina corresponde al valor mínimo de la función de costo, es decir, el error predictivo más bajo.

Hay casos donde la función puede tener múltiples mínimos, lo que dificulta la búsqueda de la solución óptima. En estos escenarios, se suele usar una versión alternativa, conocida como gradiente descendente estocástico, que introduce aleatoriedad en el proceso para evitar quedar atrapado en mínimos locales (¡el gradiente descendente normal usa todos los datos!).