

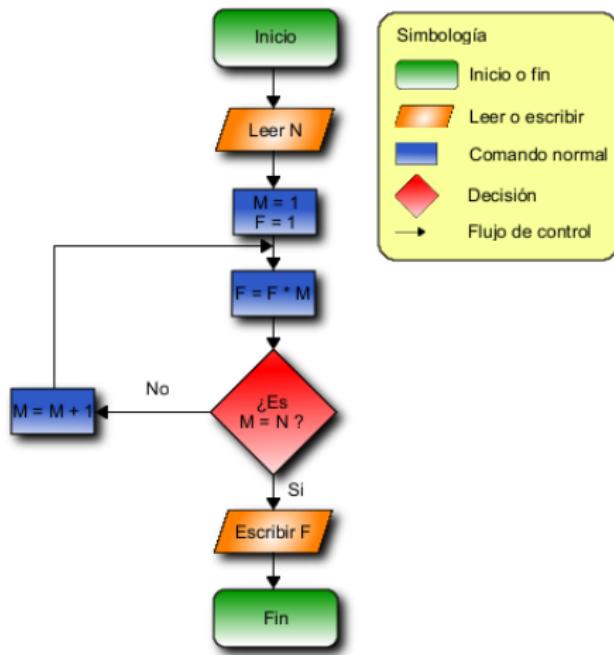
REGRESIÓN LINEAL: UNA INTRODUCCIÓN AL MACHINE LEARNING

Mario I. Caicedo

3 de septiembre de 2021



PROGRAMACIÓN IMPERATIVA



- **Algoritmo.** Secuencia de instrucciones precisas.
- La ejecución transforma la **entrada** al algoritmo en la **salida** (solución) esperada.



EXISTEN TAREAS PARA LAS CUALES NO EXISTE ALGORITMO ALGUNO

DETECCIÓN/CLASIFICACIÓN DE SPAM

- ① **Entrada:** Mensajes de e-mail
 - ② **Salida:** Booleana, un simple si o no nos indicando si un mensaje es o no deseado.
 - Clasificación personalísima (la basura de una persona puede ser el tesoro de otra).
 - **Ejercicio previo:** clasificación (labelling) de una muestra de mensajes por parte del usuario
 - El resultado del labelling es utilizado por un computador para “aprender” cuales mensajes constituyen spam y cuales no.



IA

OBJETIVO: Lograr que los computadores tomen decisiones emulando el comportamiento humano.

OBSERVACIÓN

A diferencia de una decisión cuantitativa, como decidir que número es mayor, 3/4 o 0,333, mirar una fotografía y decidir si en ella aparece una montaña, la respuesta a la segunda pregunta, viene de la experiencia, se reconoce a la montaña no como un resultado de la enseñanza algorítmica escolar, sino como la conjunción de las experiencias del pasado contemplando paisajes. Aprendizaje basado en los datos.



IA EJEMPLOS

Son muchas las veces en que se toman decisiones sobre la base de datos insuficientes, dudosos y hasta inconsistentes, en esos casos, el conocimiento previo (experiencia) es la herramienta de elección.

- Diagnóstico médico
- Reconocimiento de caracteres escritos (OCR) y de rostros



DEFINICIÓN

El Aprendizaje Estadístico es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de los datos, identificar patrones y tomar decisiones con una mínima intervención humana.



PREREQUISITOS

- Álgebra lineal, cálculo en varias variables, probabilidades, estadística.
- Elementos de programación
- Entendimiento de los datos



EL PROBLEMA DE LENGUAJE

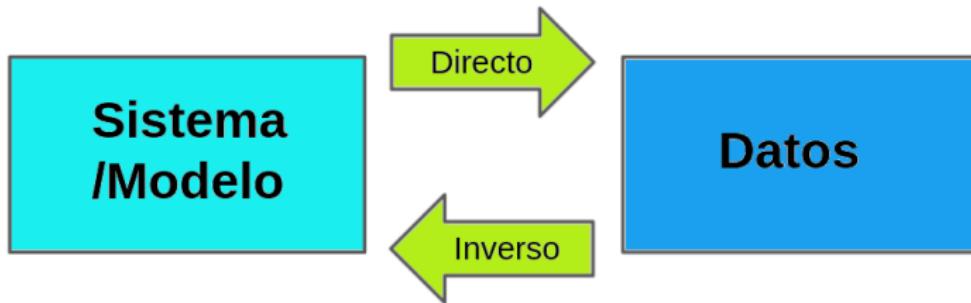
Por razones tanto históricas como de inter y transdisciplinariedad, en ML se consigue una enorme cantidad de vocabulario ad hoc que complica las cosas

- Entrenamiento (Training)
- Conjuntos de prueba y entrenamiento (Training & Testing sets)
- Regularización
- Métodos de Gradiente
- Batch
- HyperParameters



PROBLEMAS DIRECTOS E INVERSOS

$$d = F(m)$$



$$d^{(obs)} = F(m)$$



PROBLEMA DIRECTO

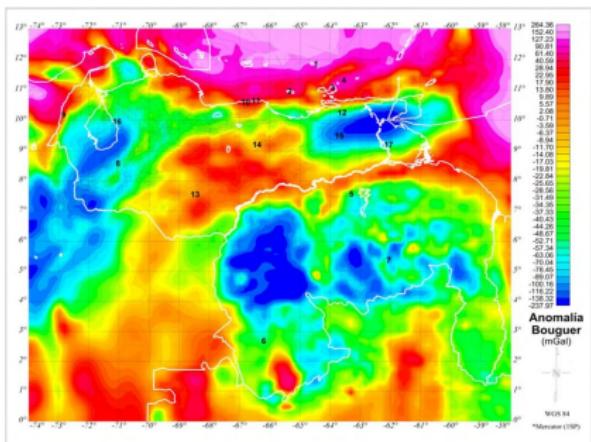
Problema Directo

Conocida la distribución de masa en una región del espacio, encuentre el campo gravitacional en la superficie de la región

$$\mathcal{G}(\mathbf{x}) = G \int \frac{\rho(\mathbf{x}') (\mathbf{x} - \mathbf{x}') d^3x'}{|\mathbf{x} - \mathbf{x}'|}$$



PROBLEMA INVERSO

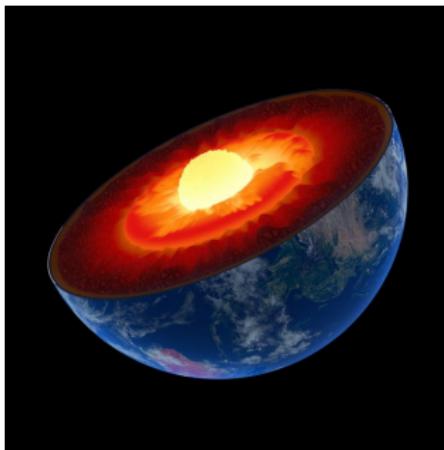
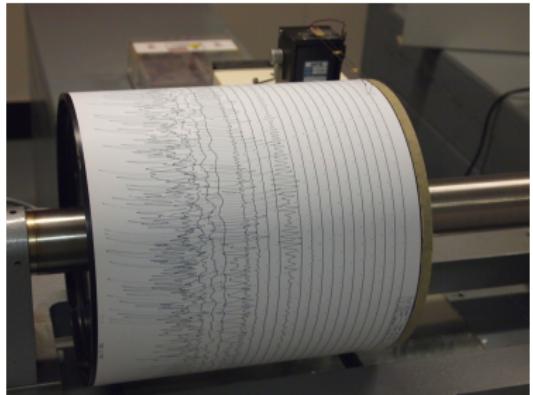


Conociendo la aceleración gravitacional en una superficie, encuentre la densidad de masa en el interior de dicha superficie



REGRESIÓN≈PROBLEMA INVERSO

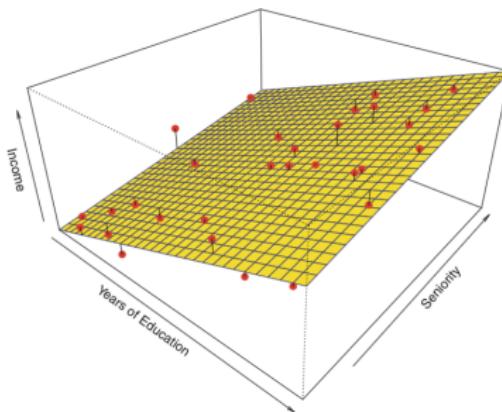
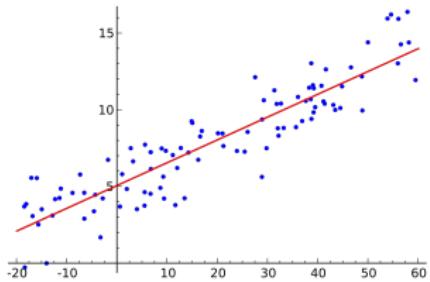
$$\mathbf{d}^{(obs)} = \mathbf{F}(\mathbf{m})$$



REGRESIÓN LINEAL I

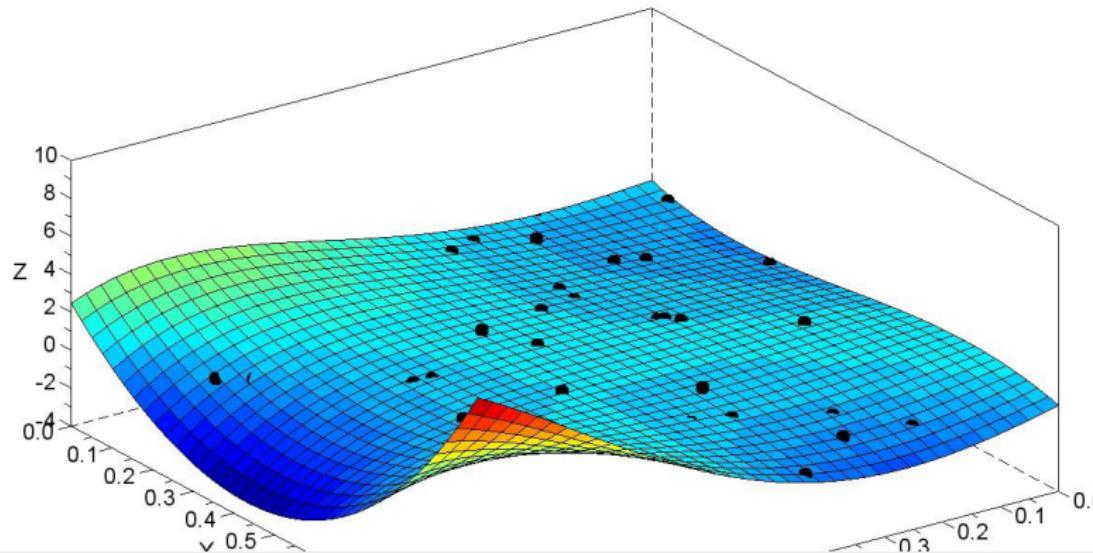
$$y_i = b + m x_i + \epsilon_i$$

$$z_i = z_0 + a_i x_i + b_i y_i + \epsilon_i$$



REGRESIÓN LINEAL II

$$z_i = z_0 + a_i x_i + b_i y_i + c_i x_i^2 + d_i y_i^2 + e_{ij} x_i y_j + [\dots] + \epsilon_i$$



REGRESIÓN LINEAL SIMPLE I

- El problema de interés comienza por un conjunto de datos experimentales que vienen dados de a pares

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

- Típicamente, los valores y_i han sido medidos con algún error y al hacer un gráfico y vs x encontramos que parece haber una relación del tipo

$$y = mx + n$$



REGRESIÓN LINEAL SIMPLE I

- En la notación de **problemas inversos**, se tienen N mediciones $d^{(i)}$, que dependen de cantidades $x^{(i)}$ y dos parámetros m y n , según las fórmulas (modelo),

$$d^{(i)} = m x^{(i)} + n + \varepsilon^{(i)}, \quad i = 1, 2, \dots, N$$

- Los valores $\varepsilon^{(i)}$ son las incertidumbres en las mediciones
 - En la notación estándar de estadística y ML

$$y^{(i)} = h_{\Theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)} + \varepsilon^{(i)}$$

donde h_{Θ} es denominado: **hipótesis** y la cantidad x **predictor**



REGRESIÓN LINEAL SIMPLE II

- Las relaciones entre las mediciones y los parámetros suelen expresarse en cualquiera de las formas matriciales

$$\mathbf{d} = \mathbf{F}\mathbf{m} \text{ ó } \mathbf{y} = \mathbf{F}\Theta$$

- Donde \mathbf{d} (o \mathbf{y}) es el vector (de N entradas) de los datos , \mathbf{F} una matriz $N \times 2$ que se construye con las N mediciones de los predictores y

$$\Theta = \mathbf{m} = \begin{pmatrix} n \\ m \end{pmatrix} = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}$$



REGRESIÓN LINEAL SIMPLE III

- El problema de regresión (ajuste) ó **aprendizaje** consiste en encontrar (calcular) el vector de parámetros Θ
- Una vez que se alcanza el **aprendizaje**, es decir, se encontraron valores para m y n , estos valores (parámetros) se pueden utilizar junto con la hipótesis para llevar a cabo predicciones (extrapolaciones)



REGRESIÓN LINEAL: SOLUCIÓN ANALÍTICA

- Para resolver el problema de **aprendizaje** se introduce un problema de minimización.
- La función de costo (error cuadrático medio) es:

$$J(\mathbf{m}) = (\mathbf{d}^{(obs)} - \mathbf{F}\mathbf{m})^T (\mathbf{d}^{(obs)} - \mathbf{F}\mathbf{m})$$

- La solución al problema de aprendizaje se reduce a minimizar la función de costo con respecto a los parámetros (\mathbf{m}).
- Se puede encontrar una solución directa buscando los puntos críticos de J , esto es, resolviendo

$$\nabla_{\mathbf{m}^T} J = 0,$$

cuya solución analítica es

$$\boxed{\mathbf{m} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{d}^{(obs)}}$$

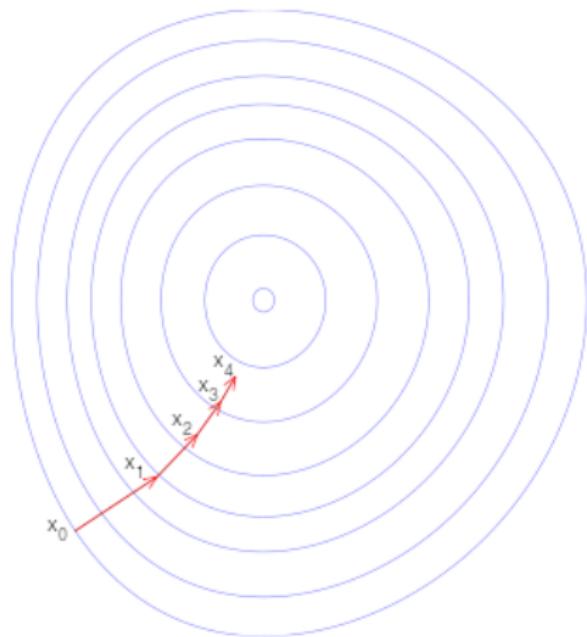


SOLUCIÓN ANALÍTICA: COMENTARIOS

- ① La solución analítica requiere el cálculo de la inversa de la matriz $\mathbf{F}^T \mathbf{F}$ que en **el caso de la regresión lineal simple** es una matriz 2×2 , lo que convierte al problema en casi trivial [la existencia de la inversa no está garantizada]
- ② En el caso de un modelo con p parámetros las cosas se complican entre otras razones por (a) Largos tiempos de cálculo para problemas muy grandes y (b) Mal condicionamiento de la matriz $\mathbf{F}^T \mathbf{F}$
- ③ El problema de aprendizaje también puede resolverse minimizando la función de costo a través de la técnica de **Descenso por Gradiente**.



DESCENSO POR GRADIENTE I



- El método de descenso por gradiente es un **algoritmo** de minimización **iterativo**
- El algoritmo utiliza la propiedad geométrica fundamental del gradiente.



DESCENSO POR GRADIENTE II

- ① El algoritmo comienza dando una semilla (valor inicial) $\Theta^{(0)}$
- ② A partir de la semilla se itera (el superíndice k indica la iteración) para conseguir nuevos valores de los parámetros

$$\Theta^{(k+1)} = \Theta^{(k)} - \mu \nabla_{\Theta^T} J(\Theta^{(k)})$$

- ③ La iteración se detiene con algún criterio. [Un criterio típico consiste en la estabilización de la función de costo].



DESCENSO POR GRADIENTE III

- Función de costo [LOSS]. Obs: J es necesaria para el algoritmo

$$J(\Theta) = (\mathbf{d}^{(obs)} - \mathbf{F}\Theta)^T(\mathbf{d}^{(obs)} - \mathbf{F}\Theta)$$

- Gradiente

$$\nabla_{\Theta^T} J(\Theta) = -\mathbf{F}^T(\mathbf{d} - \mathbf{F}\Theta)$$



PSEUDOCÓDIGO

Descenso por Gradiente

procedure GRADIENT DESCENT

Input $\Theta^{(0)}$: Semilla

$\Theta^{(k)} = \Theta^{(0)}$

While $epsilon \geq criterio$

$J^{(k)} = J(\Theta^{(k)})$

Paso de Gradiente:

$$\Theta^{(k+1)} = \Theta^{(k)} - \mu \nabla_{\Theta^T} J(\Theta^{(k)})$$

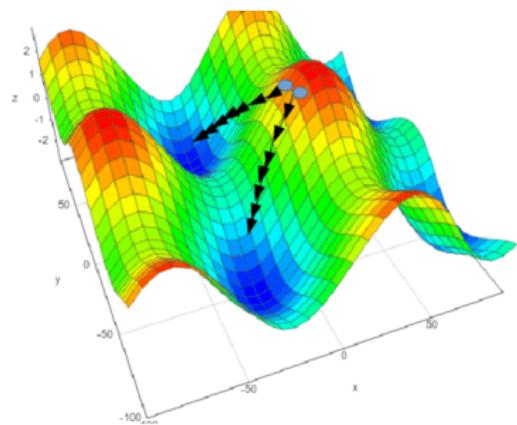
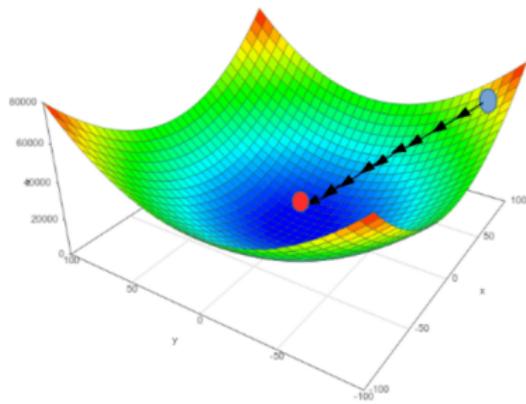
$$J^{(k+1)} = J(\Theta^{(k+1)})$$

$$\epsilon = |J^{(k+1)} - J^{(k)}|$$

Output: return $\Theta^{(k+1)}$



DESCENSO POR GRADIENTE IV



REGRESIÓN LINEAL: OBSERVACIONES I

- No hay garantías acerca de la inversibilidad o buen condicionamiento de $\mathbf{F}^T \mathbf{F}$
- Para $\dim(\boldsymbol{\Theta}) = N$, $\dim(\mathbf{F}^T \mathbf{F}) = N \times N$
- A pesar de las lindas propiedades de $\mathbf{F}^T \mathbf{F}$ e incluso si es inversible, la inversión puede ser un problema bien difícil de álgebra lineal numérica
- Para las regresiones lineales, la función de pérdida de mínimos cuadrados es convexa lo que garantiza que el método de descenso encontrará un mínimo (se alcanza el aprendizaje)



REGRESIÓN LINEAL: OBSERVACIONES II

- ¿De donde sale la función de costo? [Máxima Verosimilitud].
- Por diversas razones conviene “regularizar” la función de costo

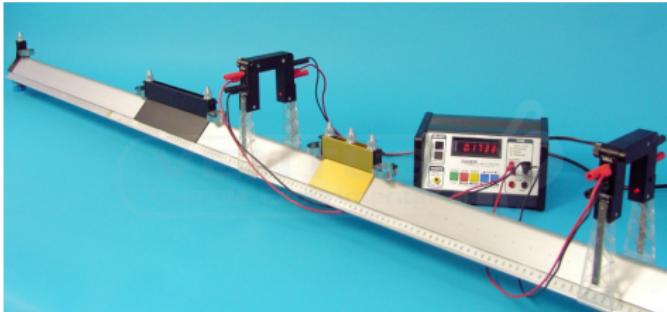
$$J(\Theta) = (\mathbf{d}^{(obs)} - \mathbf{F}\Theta)^T(\mathbf{d}^{(obs)} - \mathbf{F}\Theta) + \lambda^2 \Theta^T \Theta$$

- La regularización implica la solución exacta:

$$\Theta = [\mathbf{F}^T \mathbf{F} + \lambda^2 \mathbf{I}]^{-1} \mathbf{F}^T \mathbf{d}^{(obs)}$$

- ¿Como saber que tan “buena” es la regresión?.





En un experimento de medición de posición en función del tiempo se encuentra que los datos parecen alinearse ($x = x_0 + v_0 t$)

$$\mathbf{X}^{(obs)} = \mathbf{F}\boldsymbol{\Theta} \iff \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_N \end{pmatrix} \begin{pmatrix} x_0 \\ v_0 \end{pmatrix}$$

SOLUCIÓN ANALÍTICA

$$\boldsymbol{\Theta} = [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{X}^{(obs)}, \quad \boldsymbol{\Theta} = \begin{pmatrix} N & \sum t_i \\ \sum t_i & \sum (t_i)^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum x_i \\ \sum t_i x_i \end{pmatrix}$$

$$\theta_0 = n = \text{intercepto} = \frac{N \sum y_i - m \sum x_i}{N}$$

$$\theta_1 = m = \text{pendiente} = \frac{N \sum x_i y_i - \sum x_i \sum y_j}{N \sum x_i^2 - (\sum x_i)^2}$$



DESCENSO POR GRADIENTE I

$$\nabla_{\Theta^T} J(\Theta) = -\mathbf{F}^T (\mathbf{X}^{(obs)} - \mathbf{F}\Theta)$$

$$\nabla_{\Theta^T} J(\Theta) = - \begin{pmatrix} 1 & 1 & 1 \dots & 1 \\ t_1 & t_2 & t_3 \dots & t_N \end{pmatrix} \left[\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix} - \begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ 1 & t_3 \\ \vdots & \vdots \\ 1 & t_N \end{pmatrix} \begin{pmatrix} x_0 \\ v_0 \end{pmatrix} \right]$$



DESCENSO POR GRADIENTE II

$$\nabla_{\Theta^T} J(\Theta) = - \begin{pmatrix} \sum x_i \\ \sum t_i x_i \end{pmatrix} + \begin{pmatrix} N & \sum t_i \\ \sum t_i & \sum (t_i)^2 \end{pmatrix} \begin{pmatrix} x_0 \\ v_0 \end{pmatrix}$$



● Lenguajes

- ① R
- ② Python

● Bibliotecas

- ① caret, R
- ② Sklearn, Python
- ③ TensorFlow, Python Neural Networks
- ④ Pytorch, Python Neural Networks
- ⑤ Keras, Python Framework para TensorFlow

● Recursos Cloud

- ① R Studio Cloud
- ② Colab
- ③ Kaggle

