Clase 7:
Ajustes de datos

Ajustes de datos

- Motivación en el marco de la materia
- Modelado
- Ajustes de datos
    - Regresión
    - Interpolación
- Vamos a hacer en el Colab
- Bibliografía

Motivación en el marco de la materia

- Supongo que la realidad se representa con variables continuas

realidad

$$\vec{x}(t) \xrightarrow{\text{medición}} \boxed{\text{datos}}$$

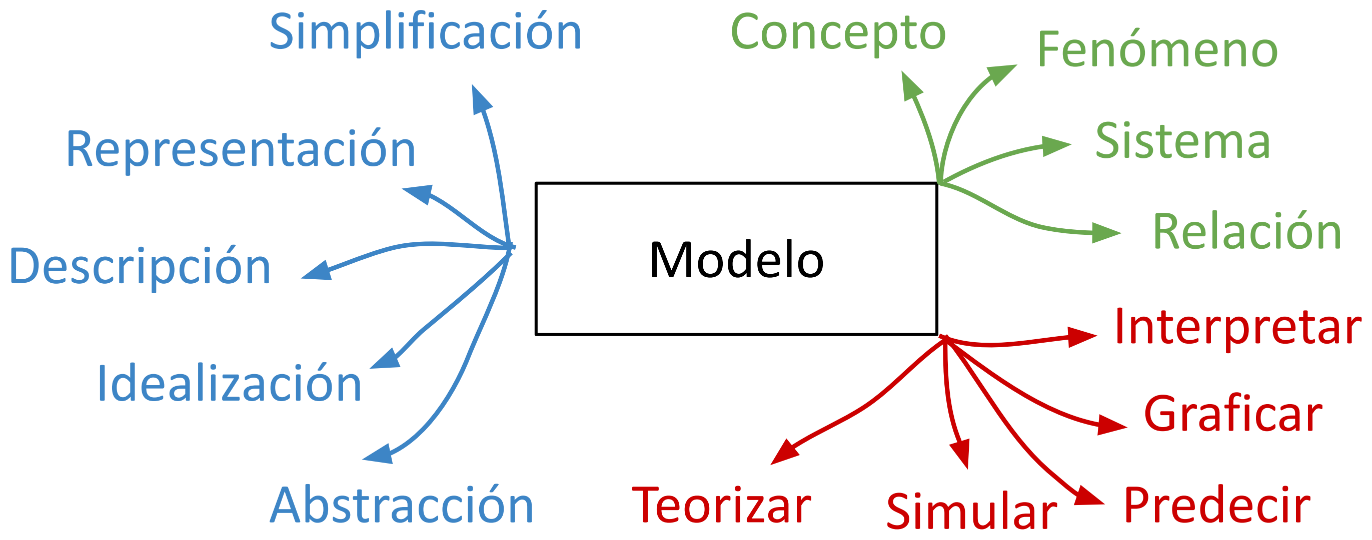
variables

Me pueden ayudar a conocer esa realidad?

Ajustes de datos

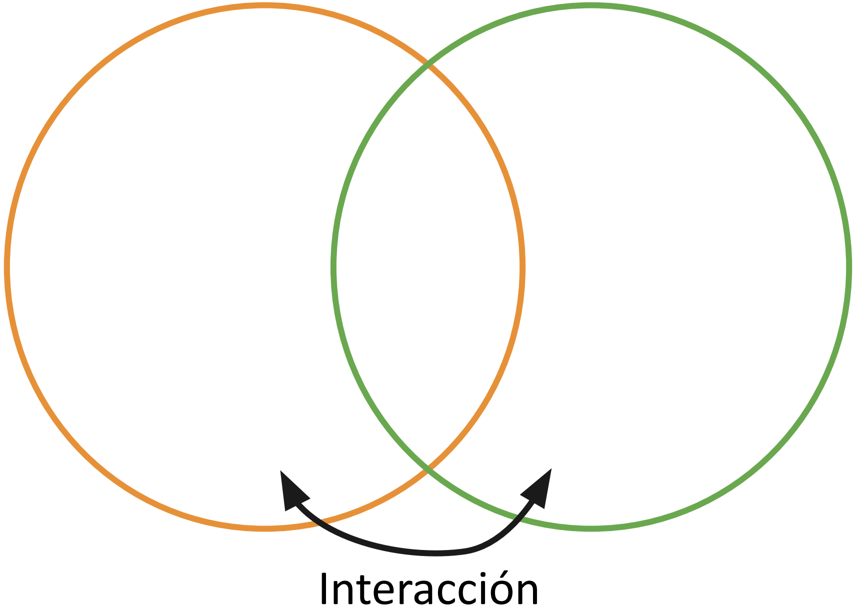
Modelado

Cómo definirían qué es un modelo?



Modelado

Data driven vs hypothesis driven



- Enfoques distintos
  - Capacidad predictiva
  - Interpretabilidad
- Tienen parte del otro enfoque
  - Hipótesis desde una observación
  - Modelos data-driven específicos según la tarea
- Pueden interactuar

Modelado

Qué le pido a un modelo?

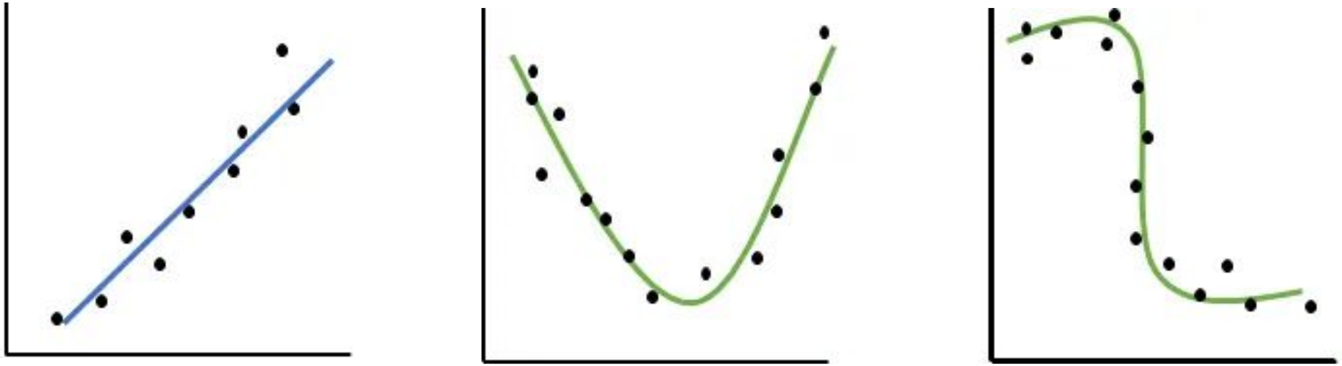
1. Que ajuste bien a los datos
2. Que permita hacer predicciones
3. Que sea interpretable

Modelado estadístico clásico; bondad de ajuste (1)

Modelos predictivos; separación en entrenamiento (1) y testeo (2) para evaluar sobreajuste (generalización)

Principio de parsimonia; usar los modelos más simples dentro de lo útil, porque no sobreajustan (2) y son interpretables (3)

Regresión



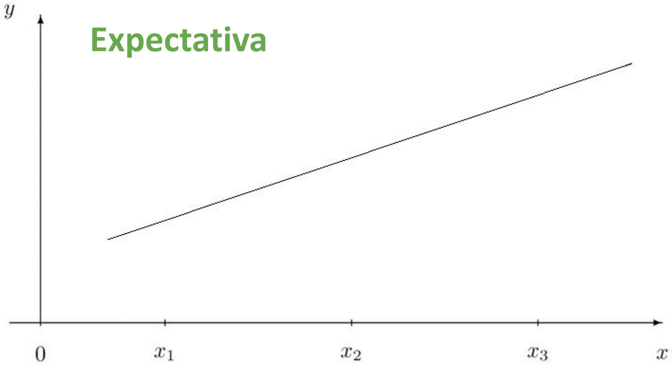
Regresión lineal Regresión no lineal (polinomio, etc)

Propongo un **modelo** a partir del cual hay una **relación entre variables (variables independientes y variables dependientes)**

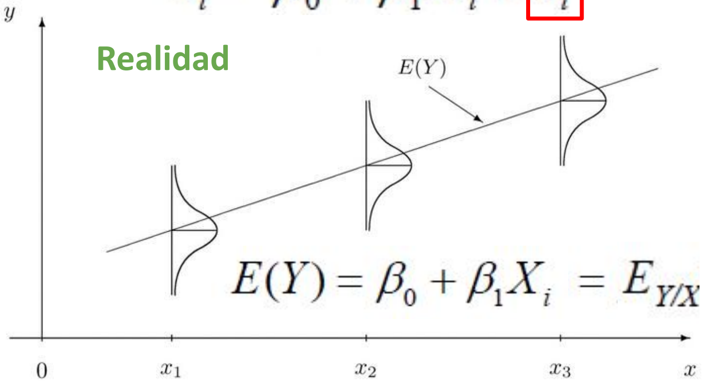
Estimo los **parámetros** que hacen que el **modelo** se **ajuste** a mis **datos**

Regresión lineal

$$Y_i = \beta_0 + \beta_1 X_i$$



$$Y_i = \beta_0 + \beta_1 X_i + \boxed{\varepsilon_i}$$

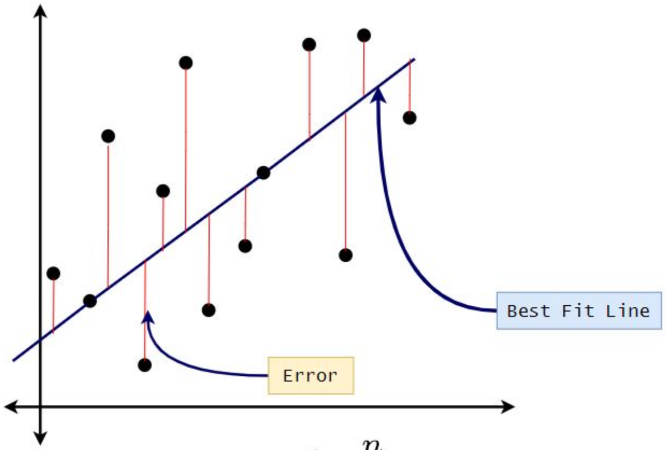


En datos reales tengo una componente aleatoria de ruido (puede ser natural)

Necesito resolver el sistema sobredeterminado

Voy a ajustar con una recta que no pasa exactamente por todos los puntos

Regresión lineal



Variable dependiente

Parámetros

Variable independiente

$$Y_i = \underbrace{\beta_0 + \beta_1}_{\text{Parámetros}} X_i$$

Ajuste del modelo (estimación)

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$$

$$J(\hat{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \left(\underbrace{y^{(i)} - \hat{y}^{(i)}}_{\text{error}} \right)^2$$

Función costo

Error cuadrático medio

Ajustar un modelo es básicamente resolver un problema de **optimización** que consiste en encontrar los **parámetros que minimizan una función costo**

Cuadrados mínimos (OLS)

$$\boxed{MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2} \longleftrightarrow \quad \text{RSS} = \sum_{i=1}^{n} e^{(i)^2} = \sum_{i=1}^{n}\left(y^{(i)} - \hat{\beta}_0 - \hat{\beta}_1 \cdot x_1^{(i)}\right)^2$$

Es posible **minimizar esta suma** de distancias **analíticamente** de modo de hallar los **mejores estimadores** para los parámetros **β0 y β1**

Derivando e igualando a cero las derivadas parciales respecto de los parámetros, se buscan los mínimos de la función suma de **distancias al cuadrado**

$$\frac{\delta \text{RSS}}{\delta \hat{\beta}_0} = 0 \qquad \frac{\delta \text{RSS}}{\delta \hat{\beta}_1} = 0$$

Estimadores de parámetros

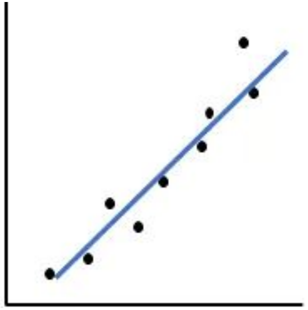
$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

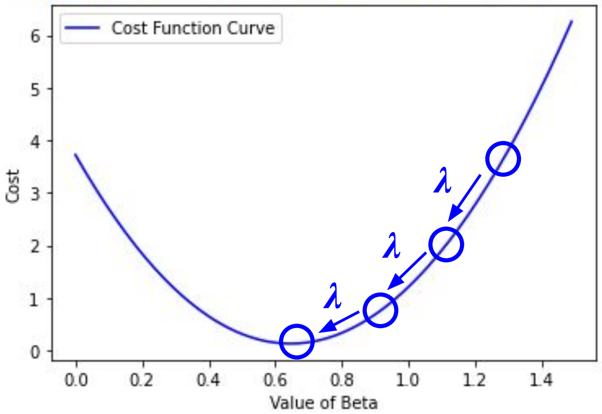
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Tienen una distribución de probabilidades

Descenso por el gradiente

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$$





Error cuadrático medio

$$J(\hat{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \big(\underbrace{y^{(i)} - \hat{y}^{(i)}}_{\text{error}} \big)^2$$

Función costo

Gradiente: $\dfrac{\partial J(\hat{\beta})}{\partial \hat{\beta}}$

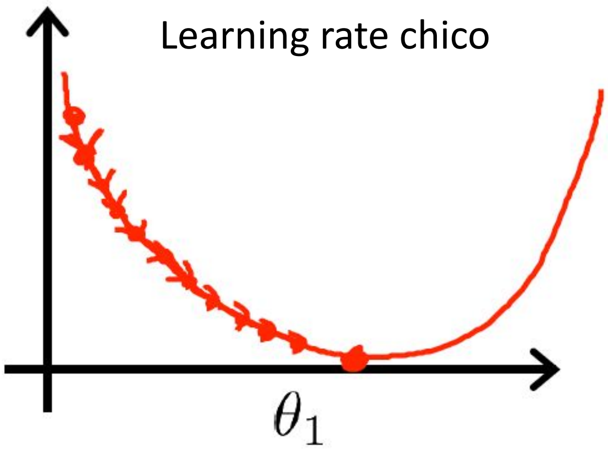
Uno de los métodos de minimización de la función costo, y el más usado, es el de **descenso por el gradiente**, con un paso llamado **learning rate** (lambda)

Descenso por el gradiente

$$\theta_j \leftarrow \theta_j - \boxed{\alpha} \frac{\partial}{\partial \theta_j} J(\theta)$$

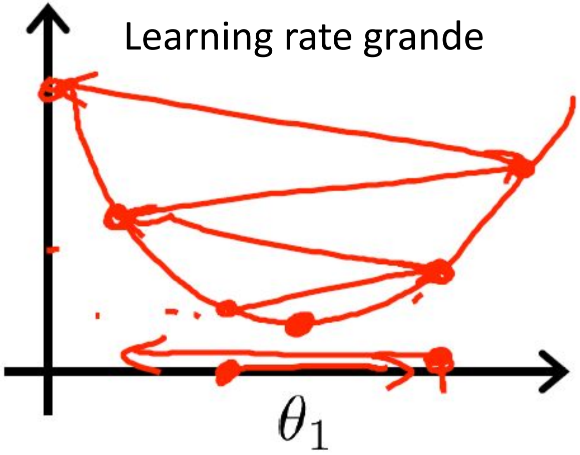
Learning rate (alpha; lambda): Factor que aporta al paso en el que me muevo cada iteración en la dirección que desciende el gradiente

Learning rate chico



Tardo en encontrar el mínimo, a riesgo de cortar la iteración antes de encontrar un valor suficientemente óptimo

Learning rate grande

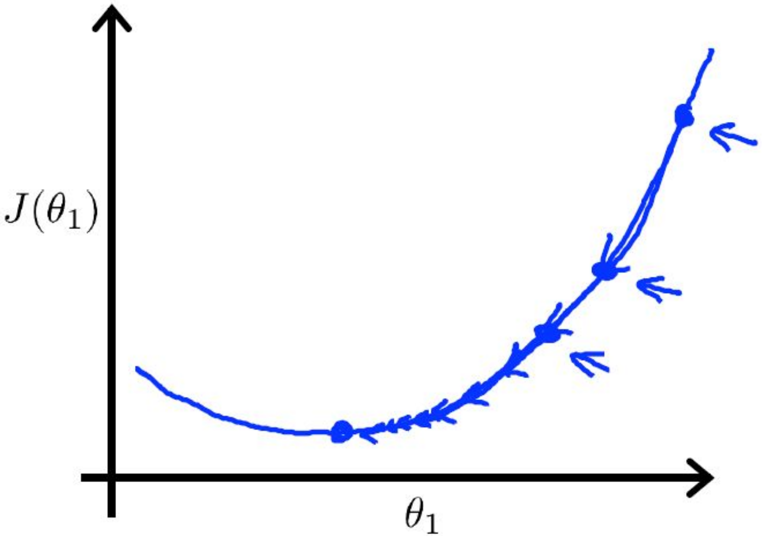


Pasos muy grandes me pueden hacer oscilar alrededor del mínimo a riesgo de nunca encontrarlo

Descenso por el gradiente

$$\theta_j \leftarrow \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta)}$$

Amplitud del gradiente: Me dice cuánto voy a reducir el costo al moverme un paso en cada dirección que indica el gradiente



El tamaño del paso también está determinado por la amplitud del gradiente evaluado en los valores de parámetros antes de moverme.

Cuando me acerco al mínimo, la derivada se hace más chica, y el paso también, aún cuando el learning rate esté fijo.

No es necesario ir reduciendo el alpha siempre, aunque a veces puede servir.

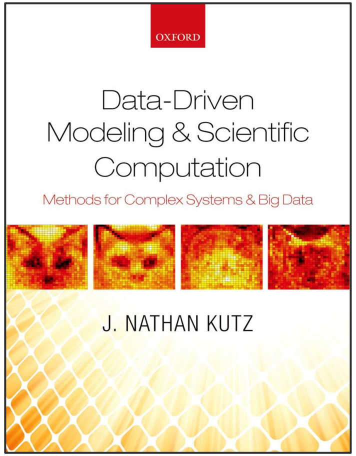
Interpolación

- Ajustamos para que el modelo pase por todos los N puntos (no ruido)
- Lo más estándar es usar polinomios de grado N-1
- Métodos numéricos clásicos: Lagrange, Newton, Vandermonde
- Puede presentar oscilaciones para N grande (fenómeno de Runge)
- Soluciones pueden ser Chebyshev o interpolación por tramos (splines)
- De gran utilidad para visualización, porque encuentro valores intermedios entre puntos, con curvas suaves, imponiendo la restricción de que la curva pase exactamente por esos puntos
- También para mejorar estimaciones de las derivadas, de las raíces, etc

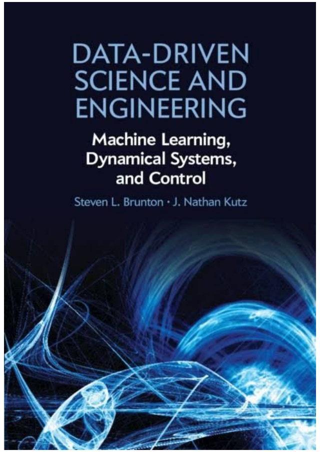
Vamos a hacer en el Colab

- Regresión
  - Población
  - Oscilador amortiguado (video)
- Filtro Savitzky-Golay
  - Derivada numérica suave
- Interpolación
  - Herramienta para encontrar raíces, visualizar

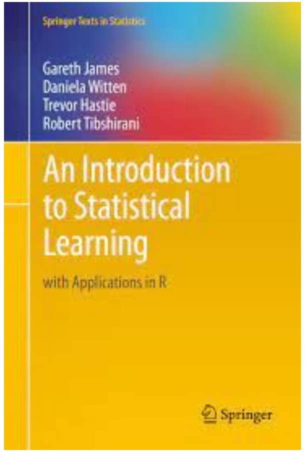
Bibliografía recomendada



Kutz 2013



Brunton & Kutz 2019



James et al 2017

Google

stackoverflow

towards data science

YouTube