Clase 8:
Reconstrucción de ODEs

Reconstrucción de ODEs

- Motivación en el marco de la materia
- Regresión + Regularización
- Reconstrucción de ODEs
  - Regresión LASSO
  - SINDy
- Vamos a hacer en el Colab
- Bibliografía

Motivación en el marco de la materia

- Sistemas dinámicos = ODEs

$$\frac{d\vec{x}}{dt} = \vec{f}\,(t, \vec{x}) \quad \xRightarrow{\quad\int\quad} \quad \vec{x}(t) \rightarrow \text{variables}$$

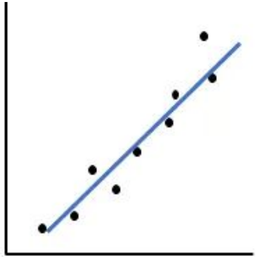
dato

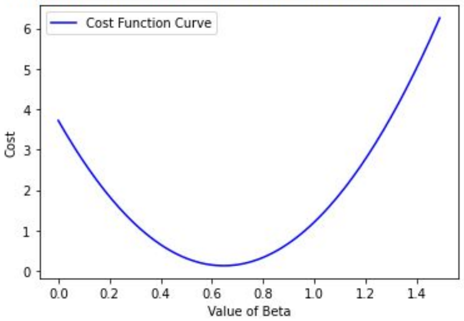
Cómo hago el proceso inverso?

Reconstrucción de ODEs

Regresión + Regularización

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





$$J(\hat{\beta}) = \overbrace{\frac{1}{n}\sum_{i=1}^{n}\left(\underbrace{y^{(i)} - \hat{y}^{(i)}}_{\text{error}}\right)^2}^{\text{Error cuadrático medio}} + \underbrace{\lambda \sum_{j=1}^{M} \left\|\hat{\beta}_j\right\|_p}_{\text{Penalización}}$$

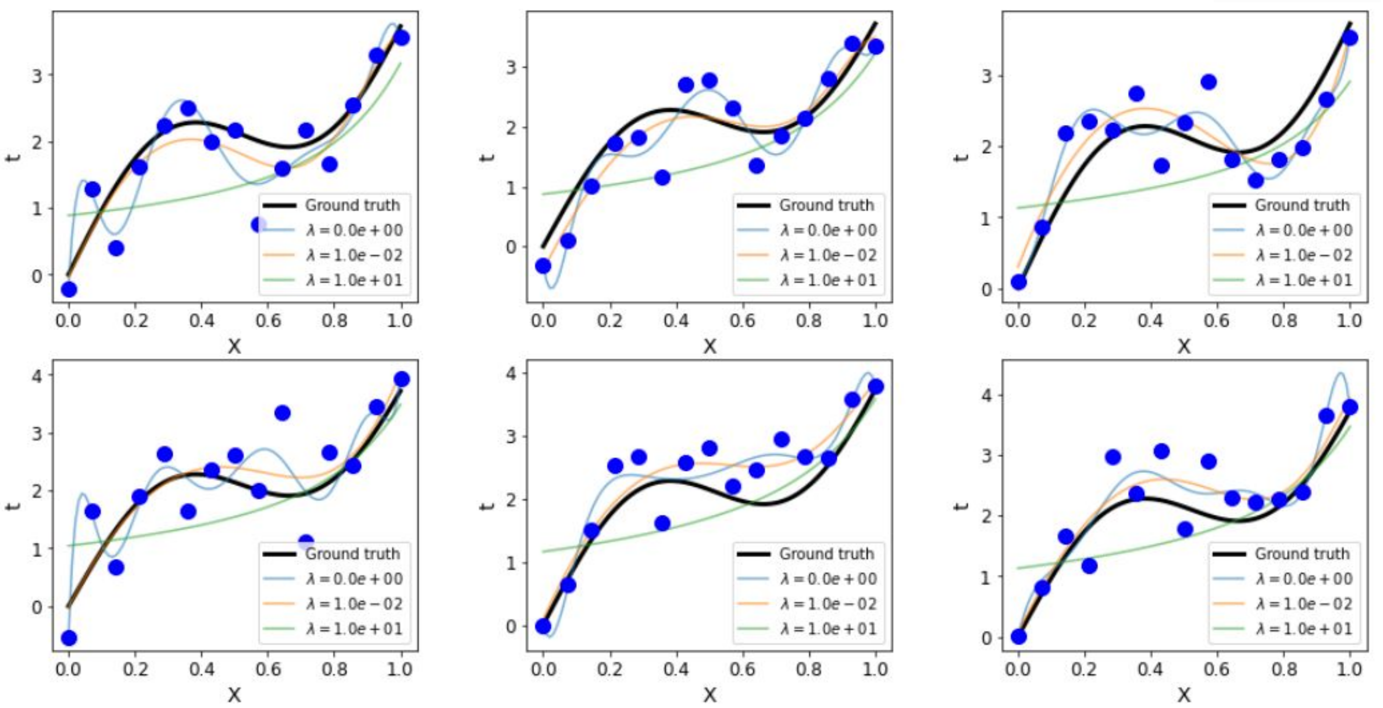
Función costo

Agrego a la **función costo** un **término de penalización**. Si un parámetro toma un valor muy alto, este término también. Minimizar la función costo penalizando ese parámetro es restringir para que no sea alto

Estoy incorporando un **hiperparámetro (lambda)** que regula la **magnitud de penalización**, asignando más importancia a ciertos parámetros por sobre otros

Permitirá regular la **flexibilidad del modelo**, haciendo que sea **menos sensible al ruido** y, por lo tanto, **reduciendo su varianza** (aunque no necesariamente siempre reduciendo la complejidad del modelo)

Regresión + Regularización

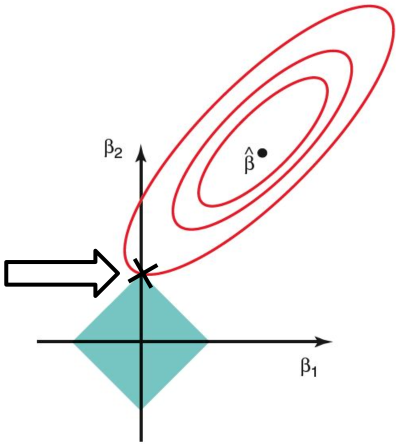


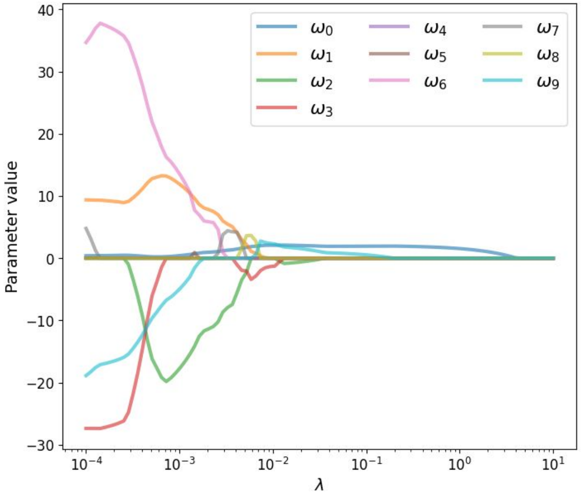
Regresión + Regularización

Regularización LASSO (p=1)

$$J(\hat{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\left(y^{(i)} - \hat{y}^{(i)}\right)^2}_{\text{error}} + \lambda \sum_{j=1}^{M}\left\|\hat{\beta}_j\right\|_p$$

Agregamos un término de **penalización** en la función costo

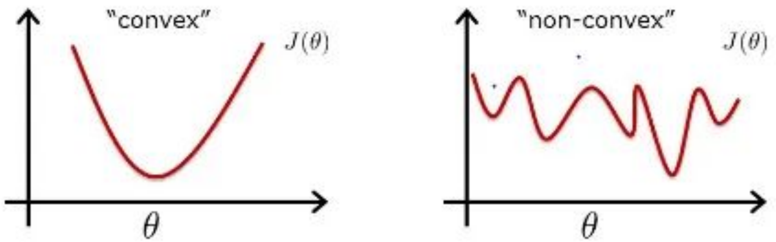




Cuando penalizo, es posible que parámetros se hagan cero (se cancelan términos)

Regresión + Regularización

La función costo en la mayoría de los casos **puede no ser convexa**



- El **descenso por el gradiente** puede **no encontrar el mínimo global**
- Pero la función costo se calcula de los **datos de entrenamiento**
- Entonces, el **mínimo global** puede dar un modelo con **sobreajuste**
- Notar que **regularizar** es buscar el mínimo de una **nueva función costo**, que incluye un **término de penalización**, por lo que **me aleja del mínimo global original**

SINDy

Sparse Identification of Nonlinear Dynamical systems

Sistema dinámico $\longrightarrow$ $$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t; \mu)$$

Puedo plantear el campo vector como un desarrollo de variables en una base de transformaciones

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi$$

Parámetros (coeficientes)

Derivadas

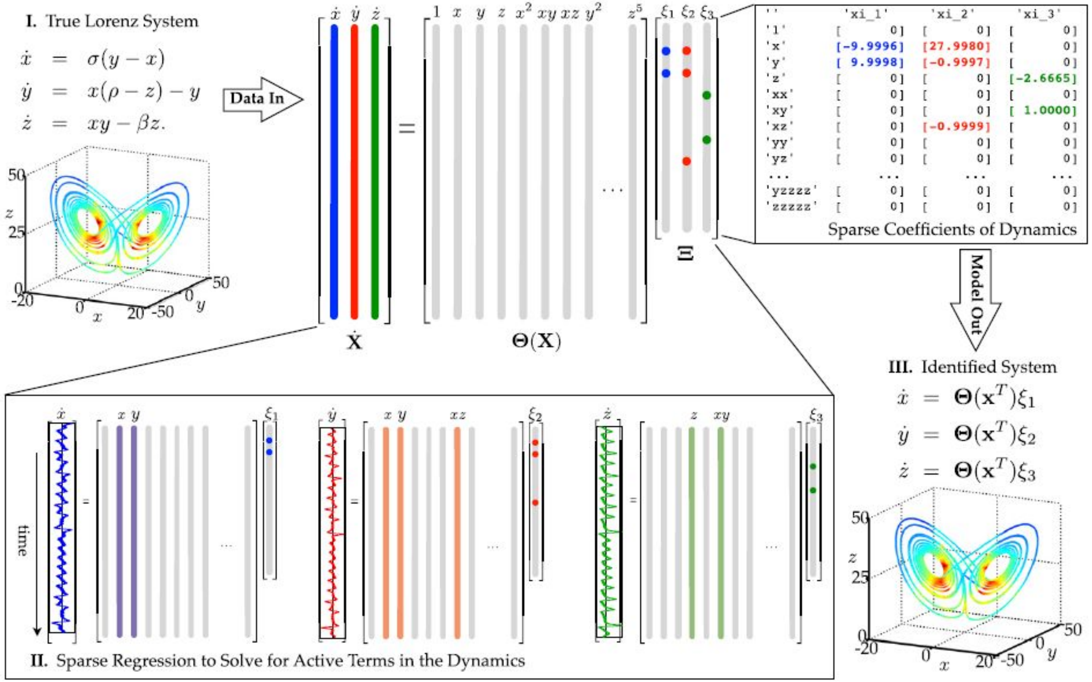
Variables transformadas (por ejemplo x^2.z)

Si tengo las variables y sus derivadas, puedo hacer una regresión lineal

En particular, regresión de tipo LASSO, agrego penalización que selecciona parámetros y los hace 0

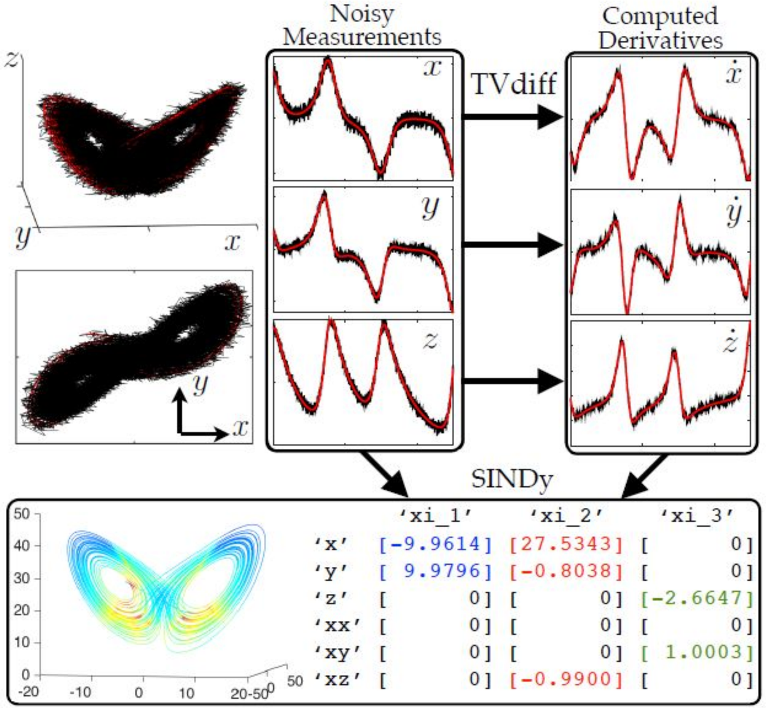
Puedo obtener una representación esparsa de las relaciones y reconstruir las ecuaciones del sistema

SINDy



Brunton, Proctor, & Kutz (2016). *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*. PNAS, *113*(15), 3932-3937.

SINDy



Funciona también

- para datos ruidosos
- cuando no tengo la derivada pero la puedo calcular numéricamente con cierta confianza

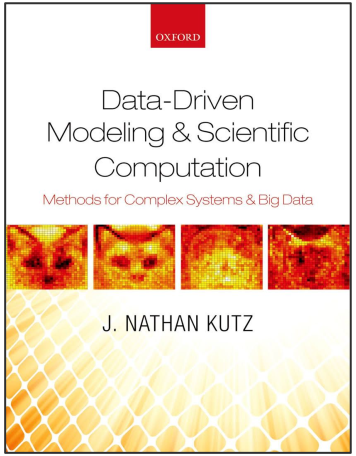
Qué limitaciones tiene esto?

Brunton, Proctor, & Kutz (2016). *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*. PNAS, *113*(15), 3932-3937.

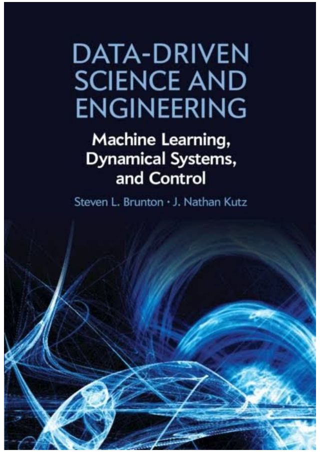
Vamos a hacer en el Colab

- Reconstrucción de ODEs
  - Atractor de Lorenz
    - LASSO
    - SINDy
  - Oscilador amortiguado (video)
  - Oscilador de relajación de Van der Pol
    - Dificultades del método?

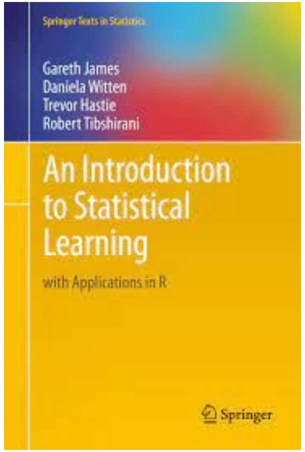
Bibliografía recomendada



Kutz 2013



Brunton & Kutz 2019



James et al 2017

Google

stackoverflow

towards data science

YouTube