

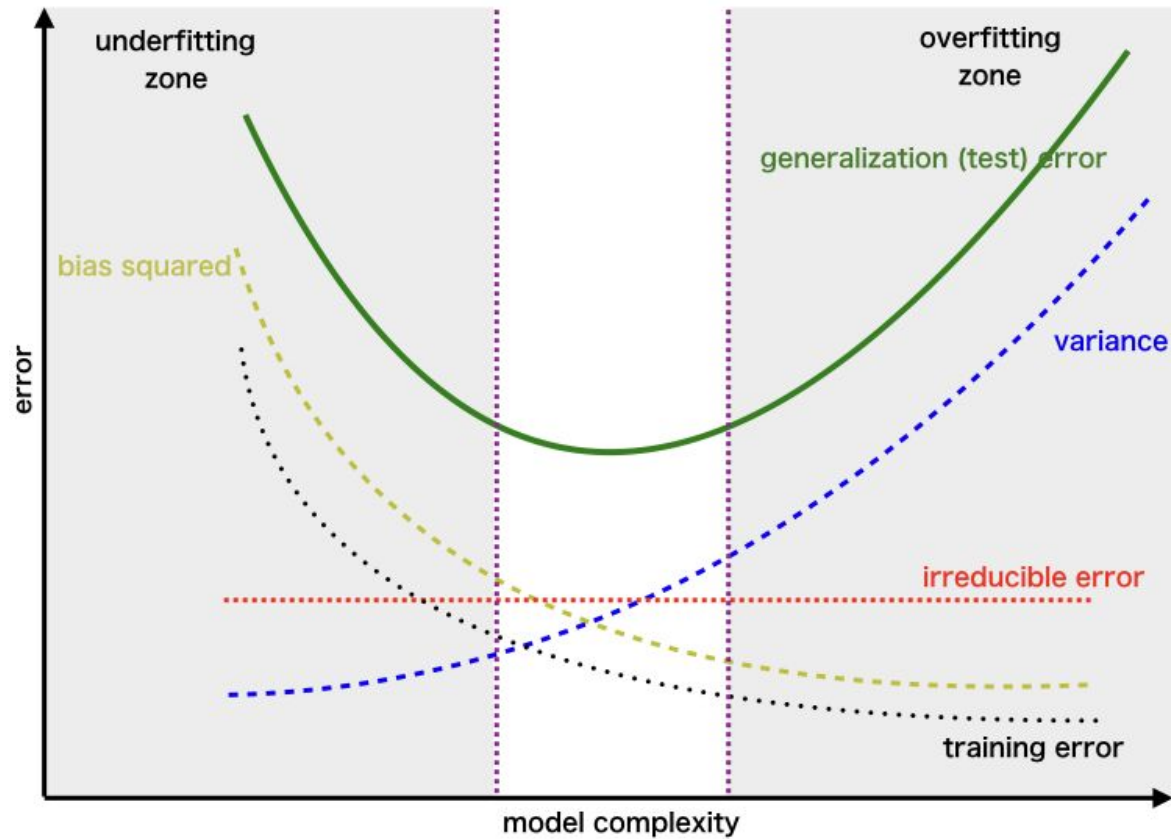
Introducción a la Inteligencia Artificial

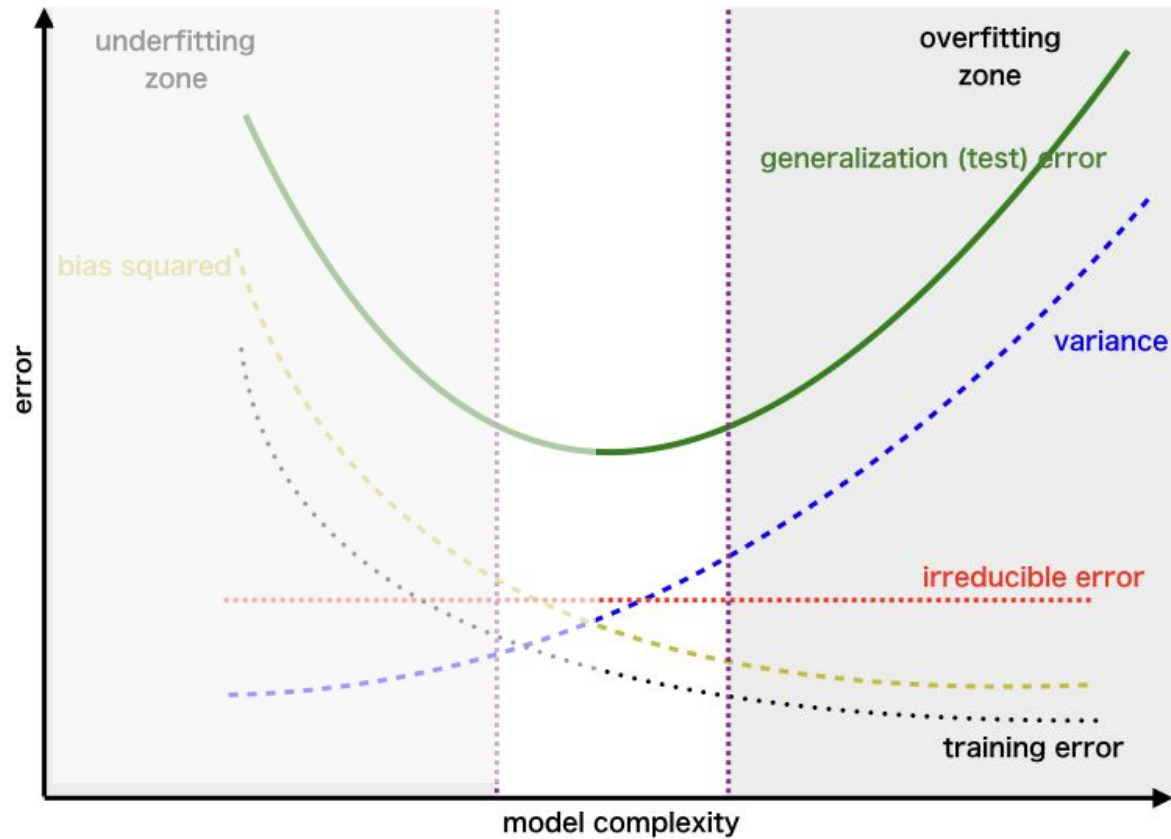
Clase 5



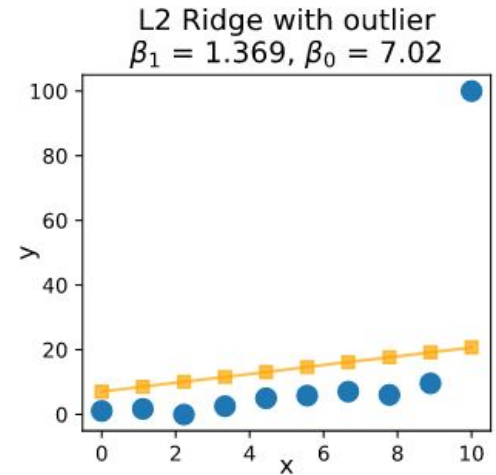
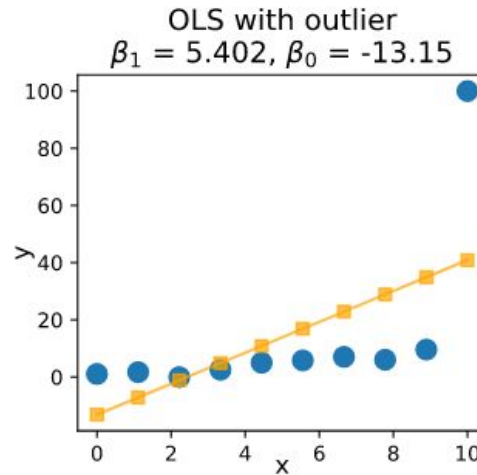
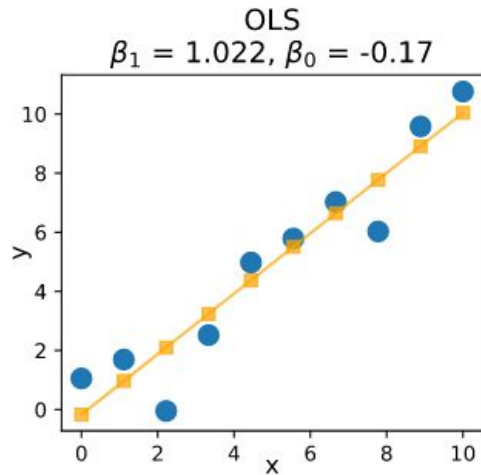
Clase 5

1. Regularización
 - a. Caso general
 - b. Ridge
 - c. Lasso
2. Gradient descent
 - a. GD
 - b. GD Estocástico
 - c. GD Mini-Batch
3. Entrenamiento de modelos
 - a. Selección de modelos
 - b. Cross-Validation





Regularización - Motivación



$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2$$

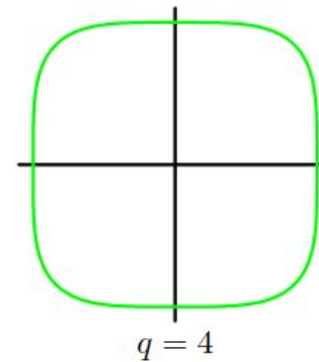
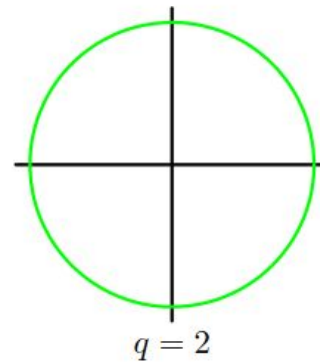
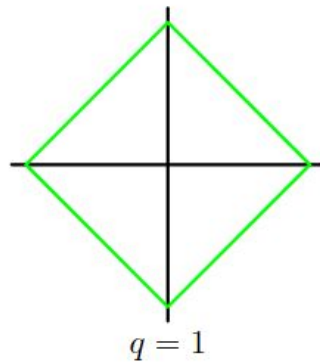
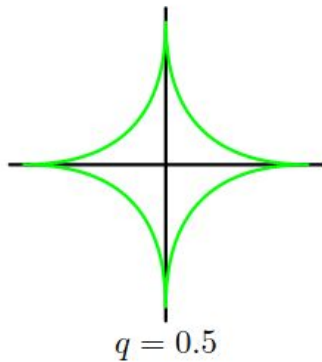
Observado - Predicción



\mathbf{w} está “libre”

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \boxed{\frac{\lambda}{2} \sum_{j=1}^M |w_j|^q}$$

Término de regularización "weight decay" → w afecta la pérdida



Lasso

Ridge

$$\mathbf{w} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T \mathbf{y}$$

Maximum A Posteriori como regularización

Distribución “a priori” de los parámetros \longrightarrow Observar data \longrightarrow Actualizar distribución (Posterior)

$$p(w) \sim D(\theta)$$

$$(\mathcal{X}, \mathcal{Y})$$

$$p(w|\mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{X}, w)p(w)}{p(\mathcal{Y}|\mathcal{X})}$$

$$w_{map} = (\Phi^T \Phi + \frac{\sigma^2}{b^2} I)^{-1} \Phi^T y$$

Gaussian prior con varianza b^2

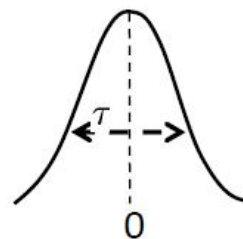
Maximum A Posteriori como regularización - Ridge (L2)

$$\hat{\beta}_{\text{MAP}} = \arg \max_{\beta} \underbrace{\log p(\{Y_i\}_{i=1}^n | \beta, \sigma^2, \{X_i\}_{i=1}^n)}_{\text{Conditional log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

I) Gaussian Prior

$$\beta \sim \mathcal{N}(0, \tau^2 \mathbf{I})$$

$$p(\beta) \propto e^{-\beta^T \beta / 2\tau^2}$$



$$\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - X_i \beta)^2 + \lambda \|\beta\|_2^2 \quad \text{Ridge Regression}$$

↓
constant(σ^2, τ^2)

$$\hat{\beta}_{\text{MAP}} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{Y}$$

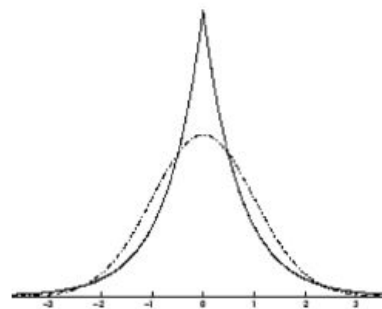
Maximum A Posteriori como regularización - LASSO (L1)

$$\hat{\beta}_{\text{MAP}} = \arg \max_{\beta} \underbrace{\log p(\{Y_i\}_{i=1}^n | \beta, \sigma^2, \{X_i\}_{i=1}^n)}_{\text{Conditional log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

II) Laplace Prior

$$\beta_i \stackrel{iid}{\sim} \text{Laplace}(0, t)$$

$$p(\beta_i) \propto e^{-|\beta_i|/t}$$

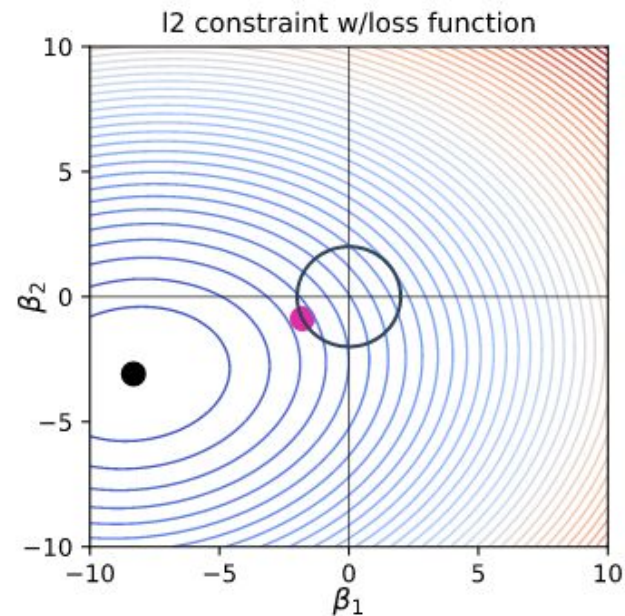
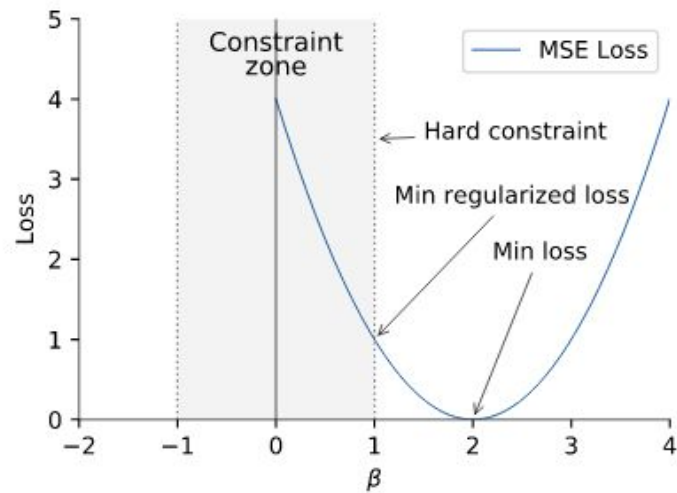


$$\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - X_i \beta)^2 + \lambda \|\beta\|_1$$

\downarrow
constant(σ^2, t)

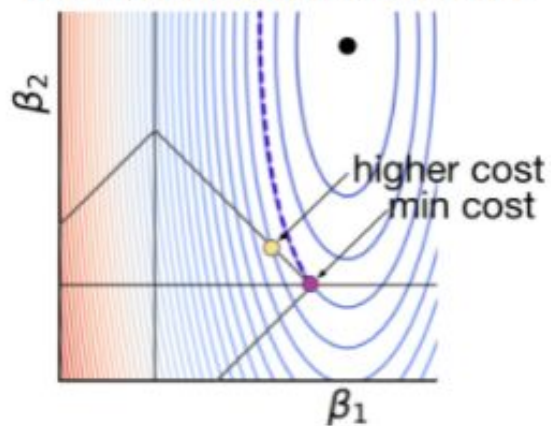
Lasso

Regularización

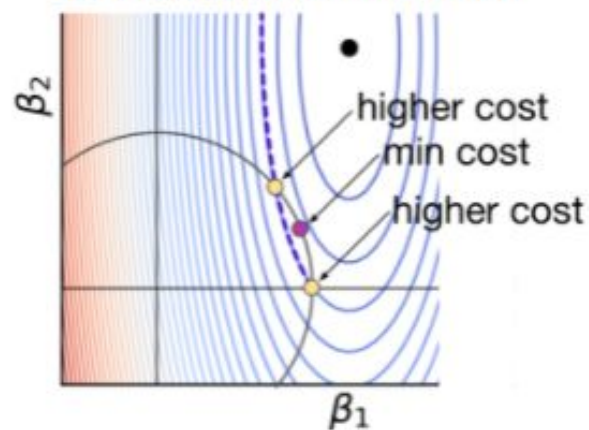


Regularización

(a) L1 Constraint Diamond



(b) L2 Constraint Circle

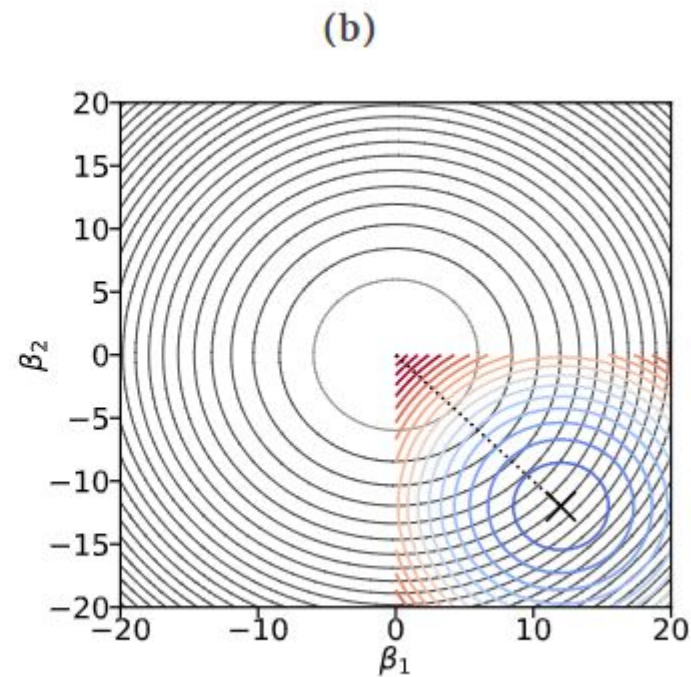
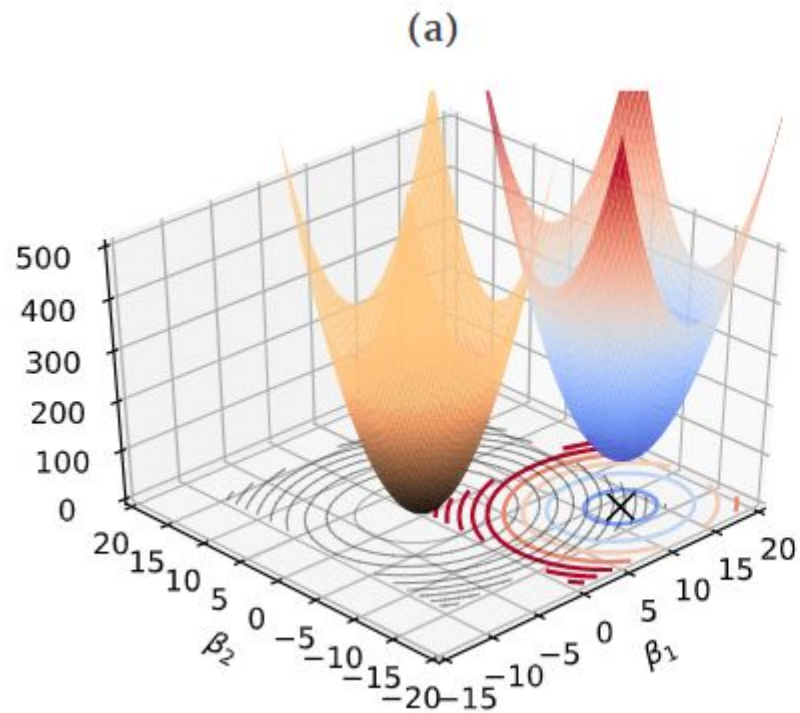


ElasticNet

$$(\alpha\lambda\|\beta\|_1 + \frac{1}{2}(1 - \alpha)\|\beta\|_2^2)$$

¿Qué β se reduce más?

Regularización



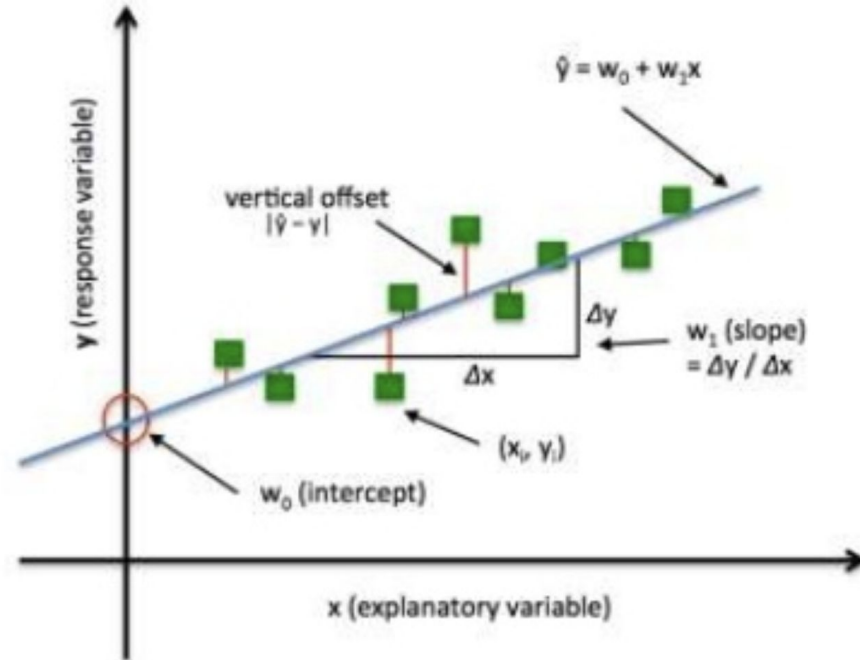
Gradiente Descendente

Implementación de Gradiente Descendente

Solucion analitica

$$\min_W \|Y - XW\|_2^2$$

$$W = (X^T X)^{-1} X^T Y$$

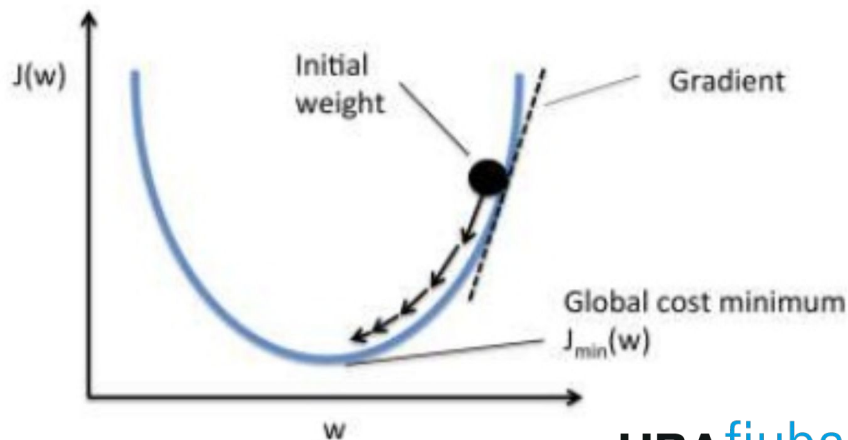


Implementación de Gradiente Descendente

Solución numérica

$$\min_W \|Y - XW\|_2^2 \implies \min_W \sum_i (y_i - X_i \cdot W)^2$$

$$W \leftarrow W - \alpha \nabla \left(\sum_i (y_i - X_i \cdot W)^2 \right)$$



Implementación de Gradiente Descendente

Solución numérica

$$\begin{aligned}\nabla_w J(w) &= \nabla_w \left(\sum_i (y_i - X_i W)^2 \right) \\ &= \sum_i \left(\nabla_w (y_i - X_i W)^2 \right) \\ &= \sum_i \left(\nabla_w (y_i - (x_{i1}w_1 + x_{i2}w_2 + \cdots + x_{im}w_m))^2 \right) \\ &= \sum_i \left(-2(y_i - \hat{y}_i)x_{ij} \right) \quad \forall j \in (1 \cdots m)\end{aligned}$$

Implementación de Gradiente Descendente

Solución numérica

$$\nabla \left(\sum_{\text{all samples}} (y_i - f_W(X_i))^2 \right)$$

Gradient Descent algorithm

for epoch in n_epochs:

- compute the predictions for **all the samples**
- compute the error between truth and predictions
- compute the gradient using **all the samples**
- update the parameters of the model

Implementación de Gradiente Descendente Estocástico

Solución numérica

$$\nabla ((y_i - f_W(X_i))^2)$$

Stochastic Gradient Descent algorithm

for epoch in n_epochs:

- shuffle the samples
- **for sample in n_samples:**
 - compute the predictions for **the sample**
 - compute the error between truth and predictions
 - compute the gradient using **the sample**
 - update the parameters of the model

Implementación de Gradiente Descendente Mini-Batch

Solución numérica

$$\nabla \left(\sum_{\text{batch samples}} (y_i - f_W(X_i))^2 \right)$$

Mini-Batch Gradient Descent algorithm

for epoch in n_epochs:

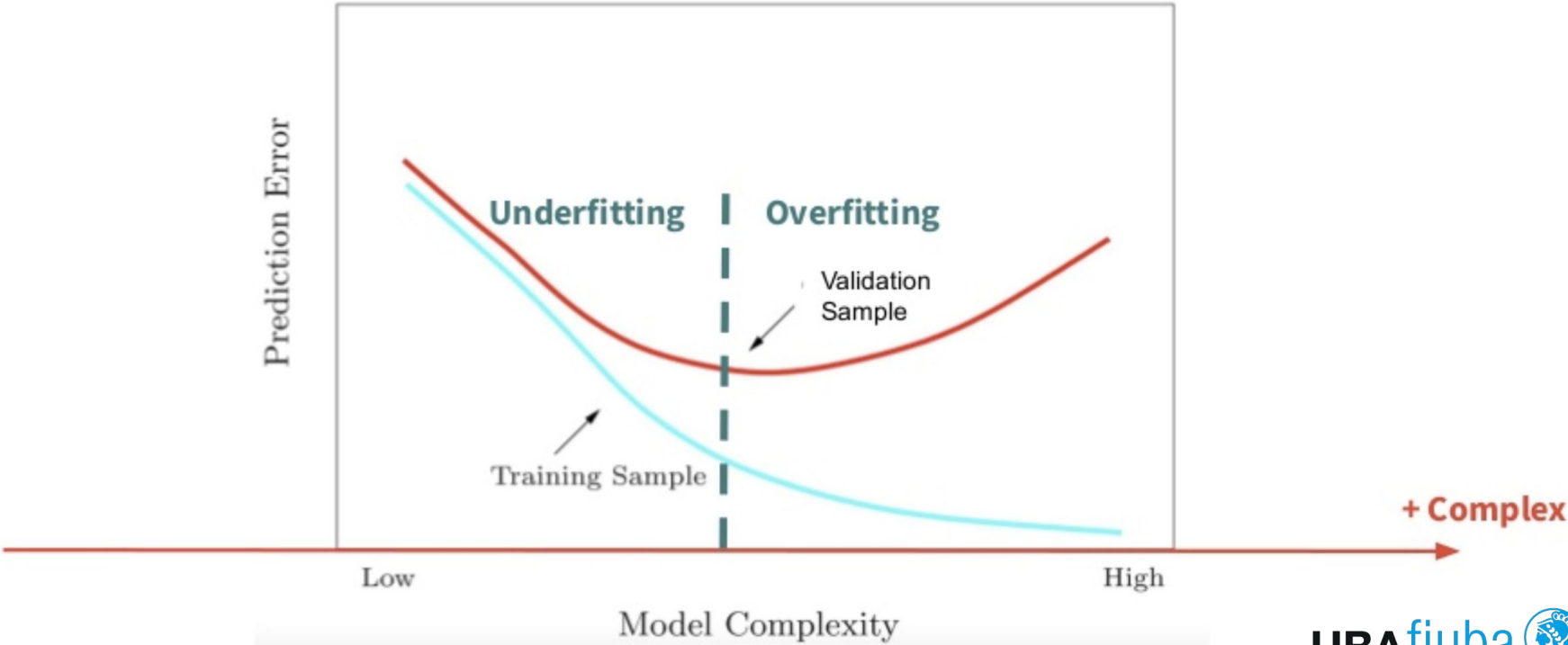
- shuffle the batches
- **for batch in n_batches:**
 - compute the predictions for **the batch**
 - compute the error for the batch
 - compute the gradient for **the batch**
 - update the parameters of the model

Comparativa de gradientes

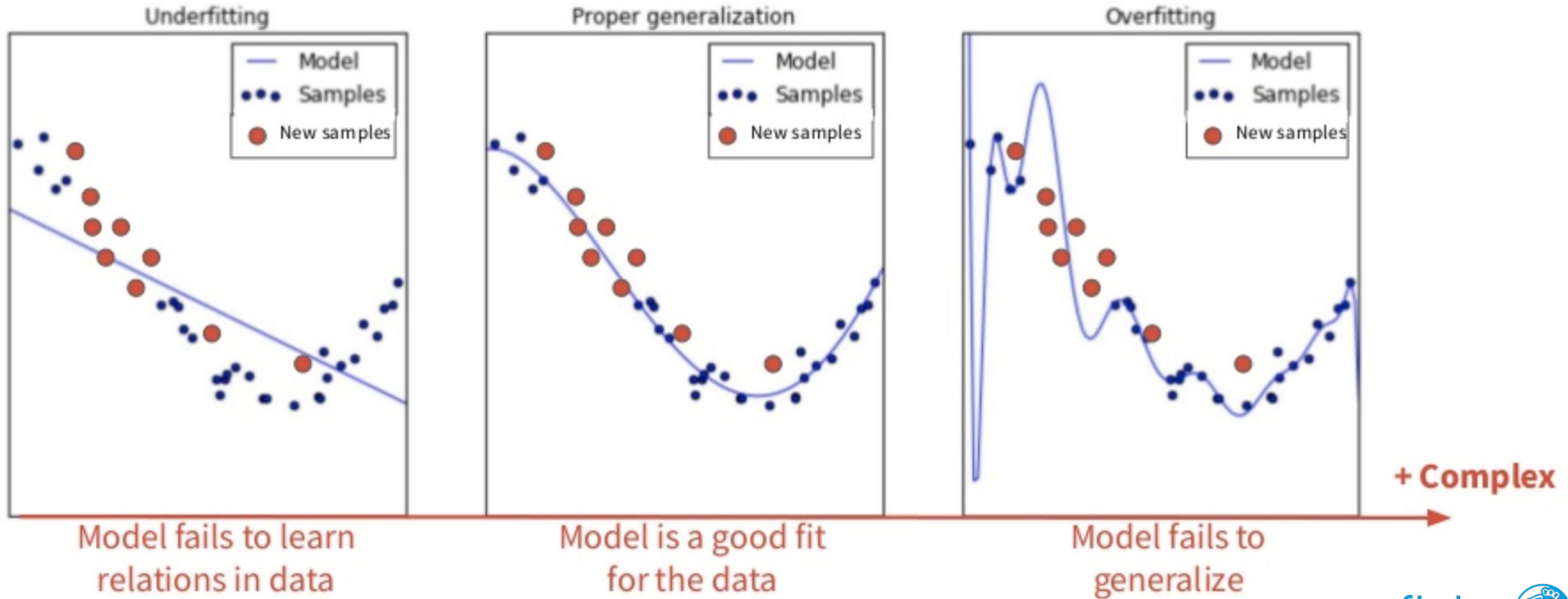
| | Gradient Descent | Stochastic Gradient Descent | Mini-Batch Gradient Descent |
|---------------------------|--|--|--|
| Gradient | $\nabla \left(\sum_{\text{all samples}} (y_i - f_W(X_i))^2 \right)$ | $\nabla \left((y_i - f_W(X_i))^2 \right)$ | $\nabla \left(\sum_{\text{batch samples}} (y_i - f_W(X_i))^2 \right)$ |
| Speed | Very Fast (vectorized) | Slow (compute sample by sample) | Fast (vectorized) |
| Memory | O(dataset) | O(1) | O(batch) |
| Convergence | Needs more epochs | Needs less epochs | Middle point between GD and SGD |
| Gradient Stability | Smooth updates in params | Noisy updates in params | Middle point between GD and SGD |

Selección de modelos

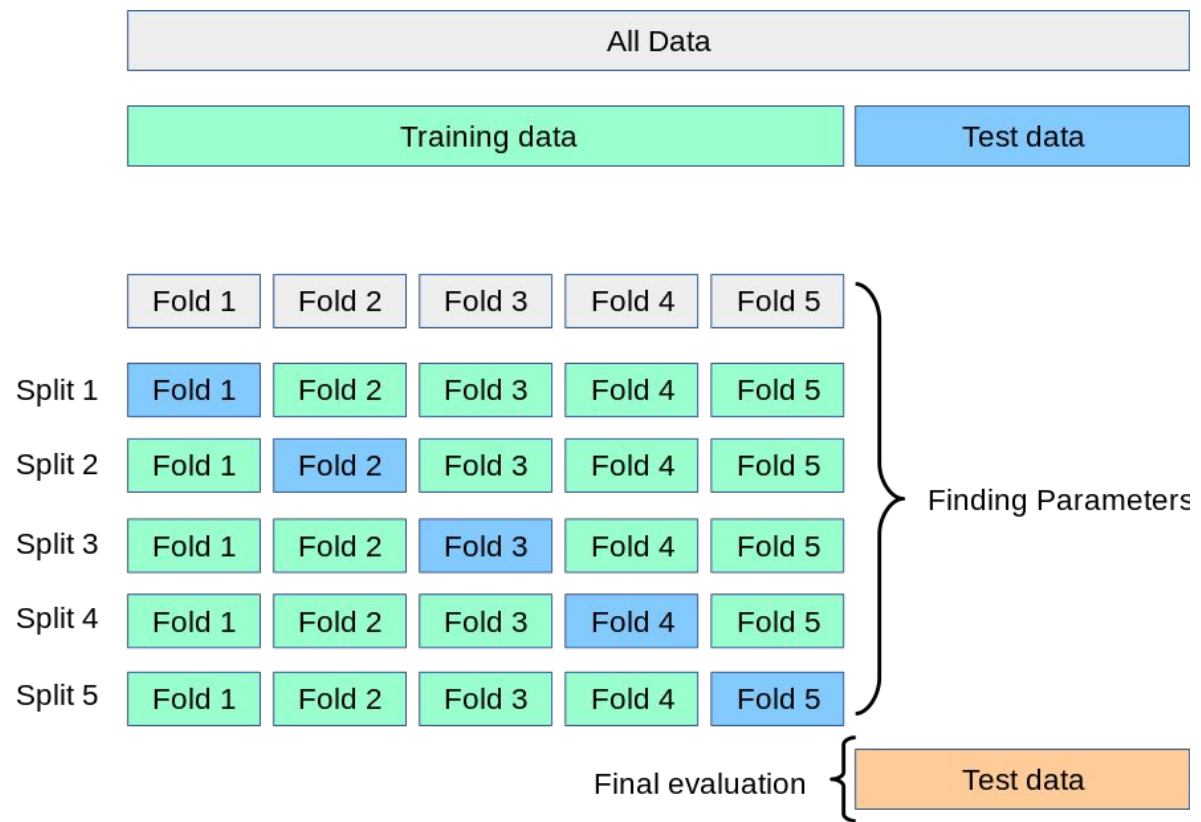
Selección de modelos



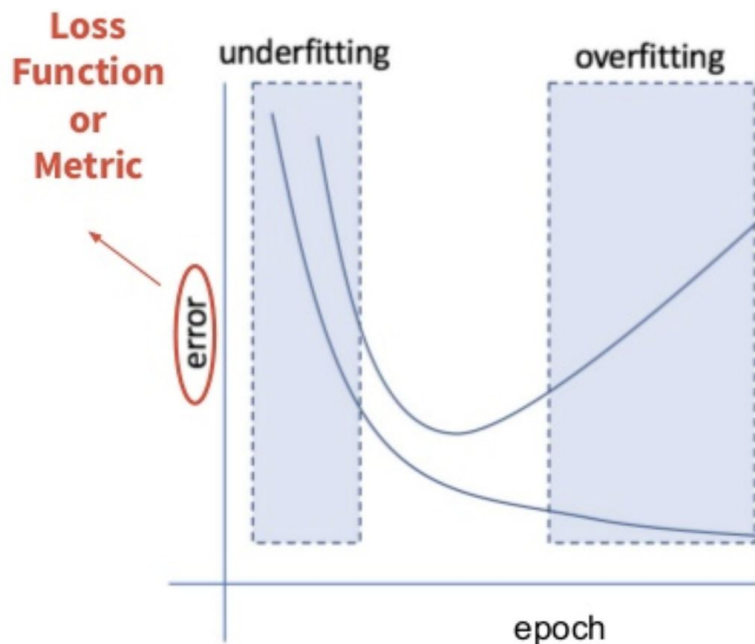
Selección de modelos



Cross-Validation



Entrenamiento numérico del modelo seleccionado - Obtención de parámetros

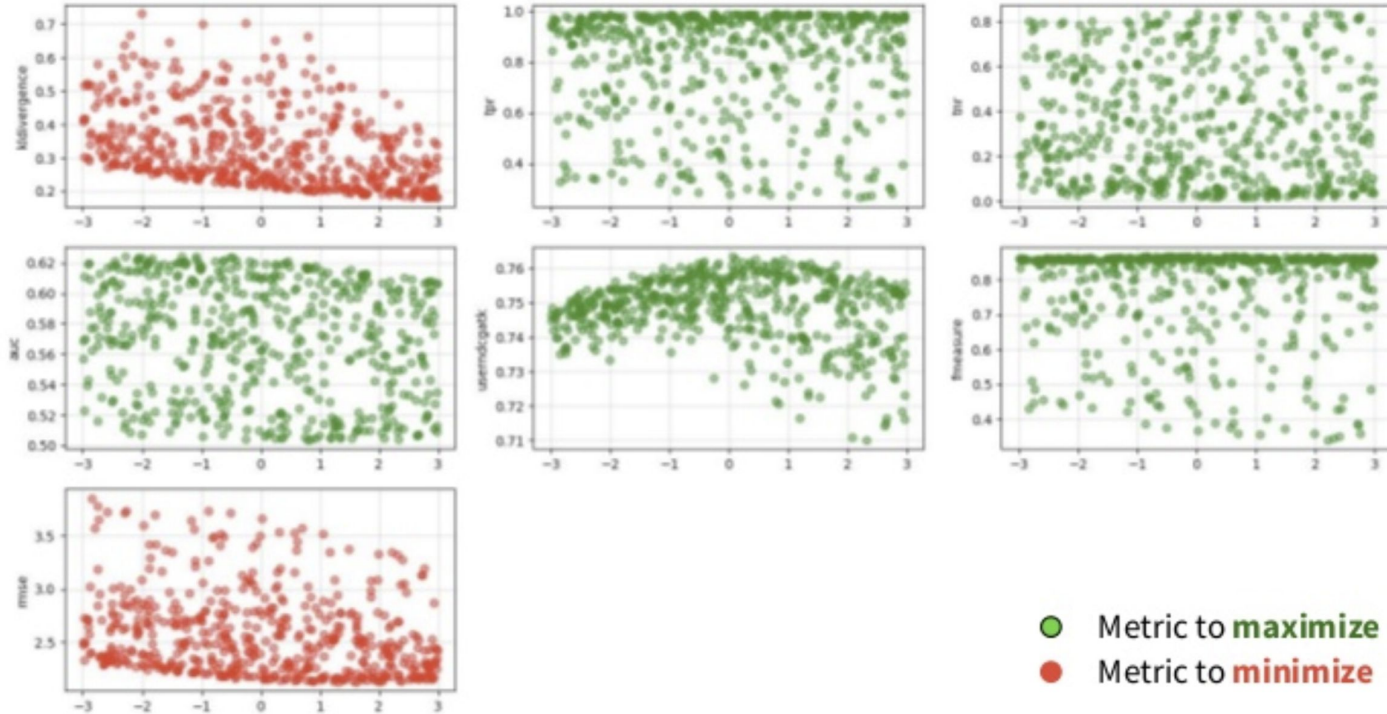


Mini-Batch Gradient Descent

for epoch in n_epochs:

- shuffle the batches
- **for batch in n_batches:**
 - compute the predictions for **the batch**
 - compute the error for the batch
 - compute the gradient for **the batch**
 - update the parameters of the model
- plot error vs epoch

Selección de los hiper parámetros



Grid Search

Random Search

Bibliografía

- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | <https://www.deeplearningbook.org/>
- Mathematics for Machine Learning | Deisenroth, Faisal, Ong
- Artificial Intelligence, A Modern Approach | Stuart J. Russell, Peter Norvig
- A visual explanation for regularization of linear models - Terence Parr
- A Complete Tutorial on Ridge and Lasso Regression in Python - Aarshay Jain