Introducción a la Inteligencia Artificial
Clase 5

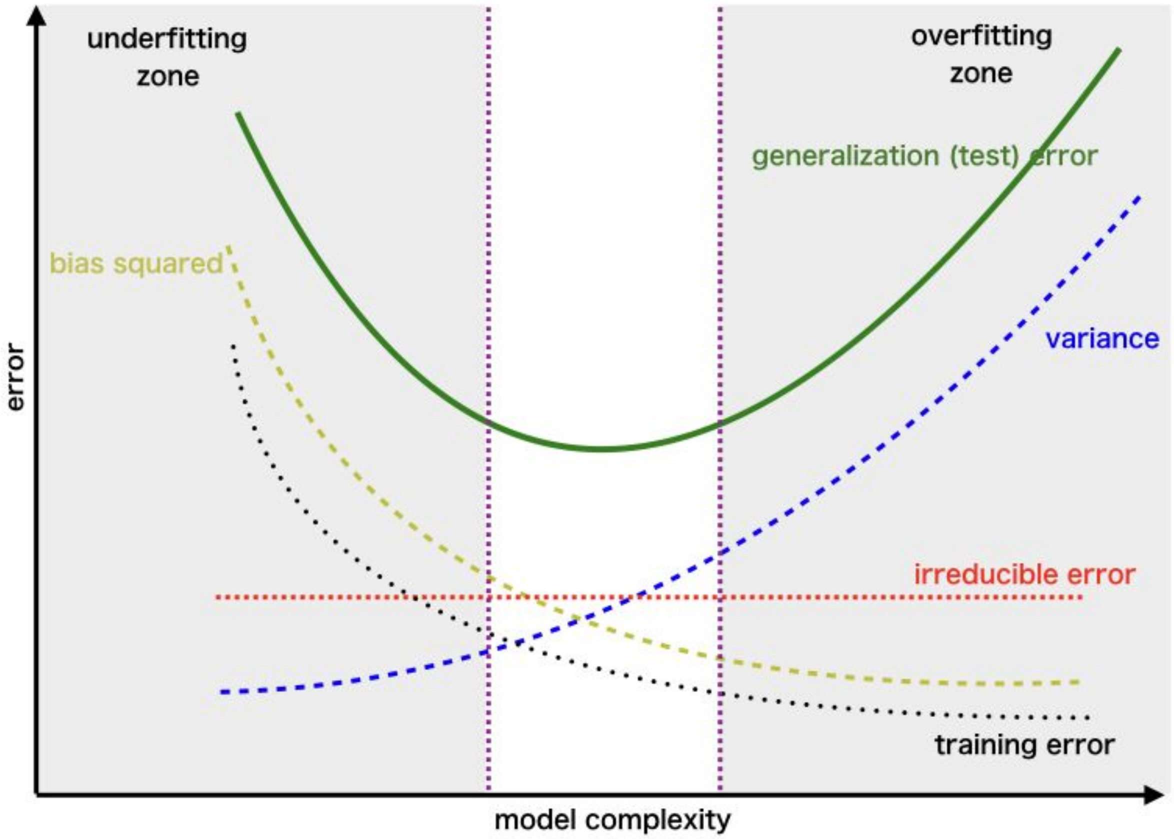
FACULTAD DE INGENIERÍA

Índice

Clase 5

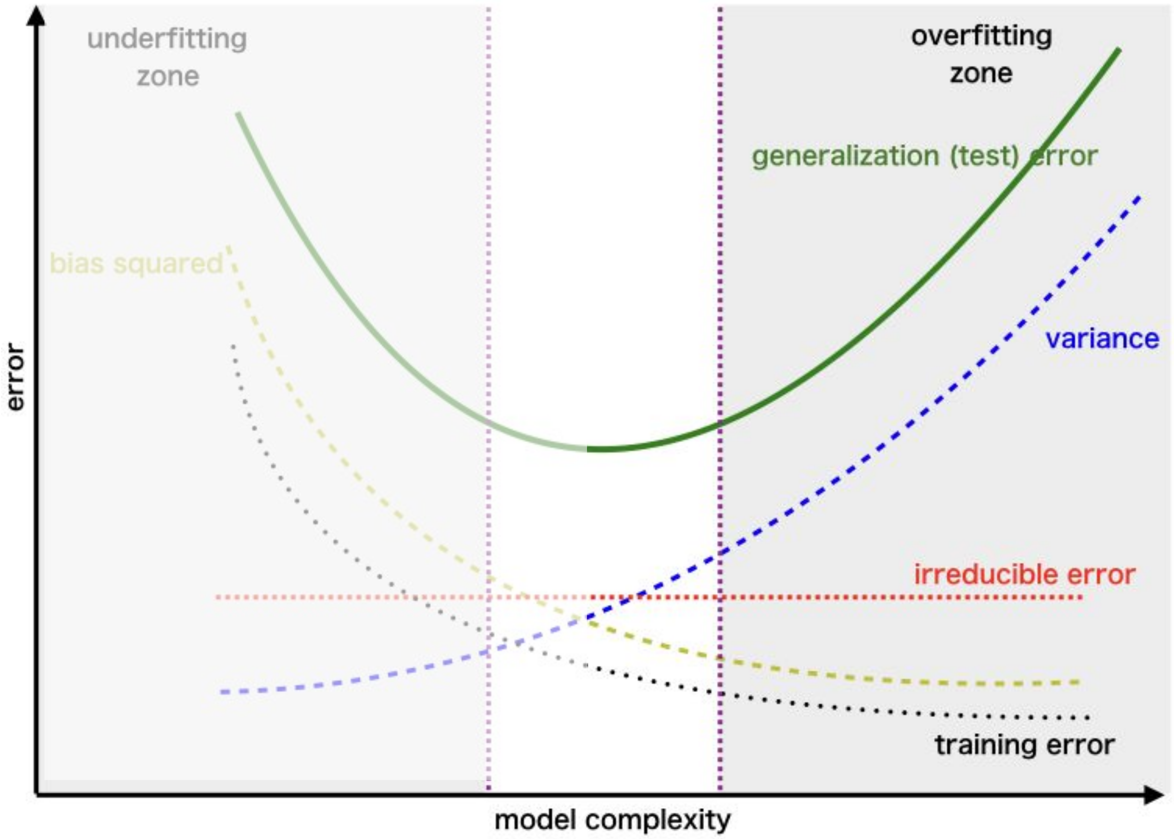
1. Regularización
   a. Caso general
   b. Ridge
   c. Lasso
2. Gradient descent
   a. GD
   b. GD Estocástico
   c. GD Mini-Batch
3. Entrenamiento de modelos
   a. Selección de modelos
   b. Cross-Validation

Regularización



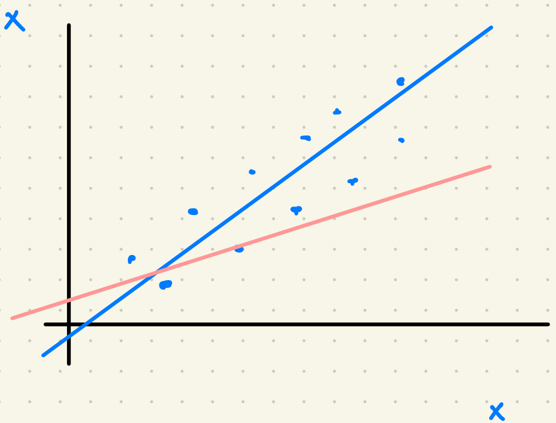
.UBAfiuba FACULTAD DE INGENIERÍA

Regularización



.UBAfiuba FACULTAD DE INGENIERÍA

Robustez



• datos s/ outliers

× outliers

— Regresion normal

— regresión c/ outliers

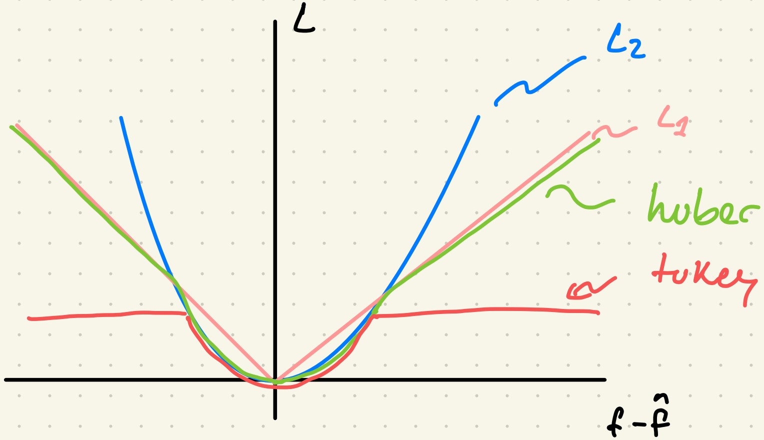
$$R(f) = \mathbb{E}\left(L\left(f,\hat{f}\right)\right) \rightarrow L(f,\hat{f}) = (f-\hat{f})^2 \rightarrow \text{función } L2$$

Riesgo empírico

(perdida cuadrática)

$$L_1 = |f-\hat{f}|$$

$$L_1 = \mathbb{I}_{|f-\hat{f}|,\,c} = \begin{cases} 0 & |f-\hat{f}| \leq c \\ 1 & |f-\hat{f}| > c \end{cases}$$



Regularización:

Con la regularización buscamos minimizar el error de estimación (Riesgo empírico) al mismo tiempo que restringimos ó limitamos el comportamiento de los parametros (parameter shrinkeage) ⇝ nos ayuda a disminuir el error de generalización.

Partimos de $\hat{y} = \beta_0 + \sum_i \beta_i x_i \rightsquigarrow \hat{\beta} = \arg\min_{\beta} \sum_i \left(y_i - \beta_0 - \sum_j \beta_j x_{ij}\right)^2$

Vamos a condicionar: $\sum_i \beta_j^2 \leq t \quad \left(\|\beta_j\|^2 \leq t \right)$

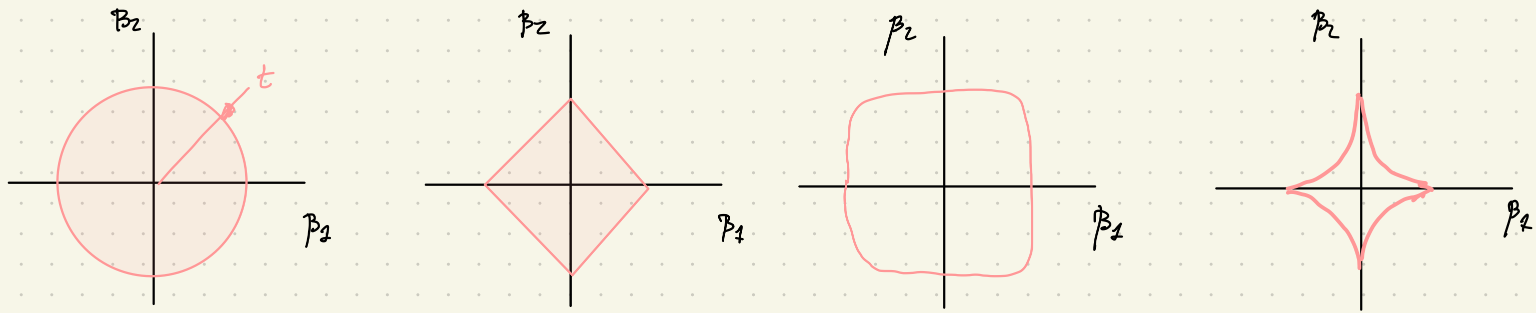
$$\hat{\beta} = \arg\min_{\beta} \sum_i \left(y - \left(\beta_0 - \sum_j \beta_j x_{ij}\right)\right)^2 - \lambda \sum_i \beta_j^{q}$$

orden de la norma

restricción

Parametro de complejidad

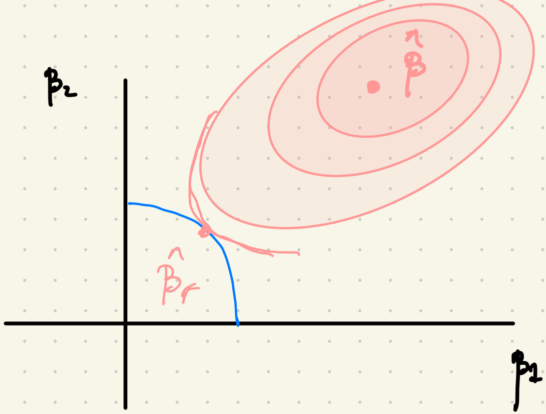
Graficamente la restricción determina la región de posibles $\hat{\beta}_j$, esa forma depende de q.

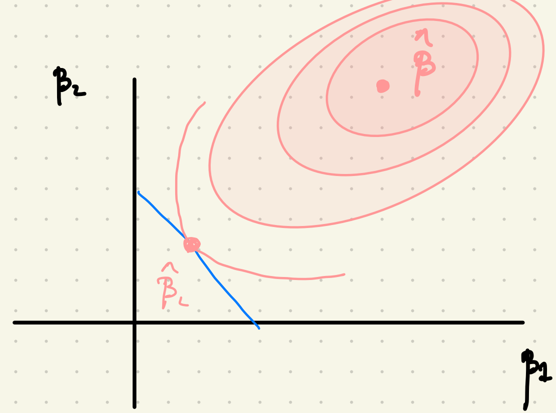


$q=2$	$q=1$	$q=4$	$q=0{,}5$
$\|\|\beta_1+\beta_2\|\|^2 \leq t$	$\|\beta_1+\beta_2\| \leq t$		

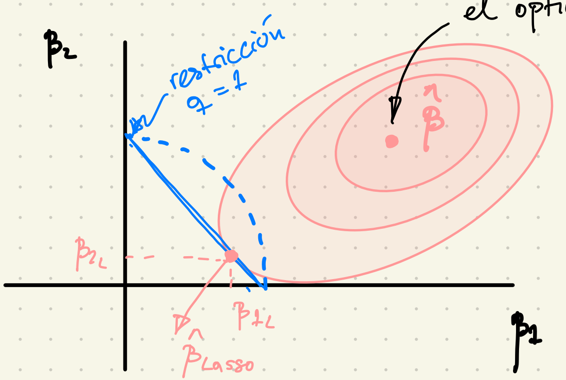
$q=2 \rightarrow$ regularización ridge

$q=1 \rightarrow$ regularización Lasso





Recordemos que $\hat\beta$ en general tiene curvas de nivel elípticas, esto viene por construcción del estimador. Nosotros vamos a buscar el punto donde la restricción corta la elipse.



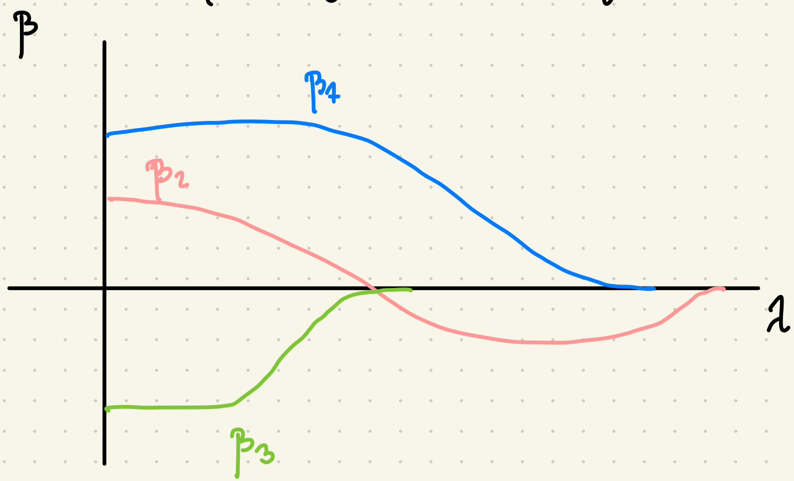
el óptimo más óptimo (global)

$$B(f) = E\{f - \hat f\}$$

Reg Lasso decimos que es autoselectora de variables.

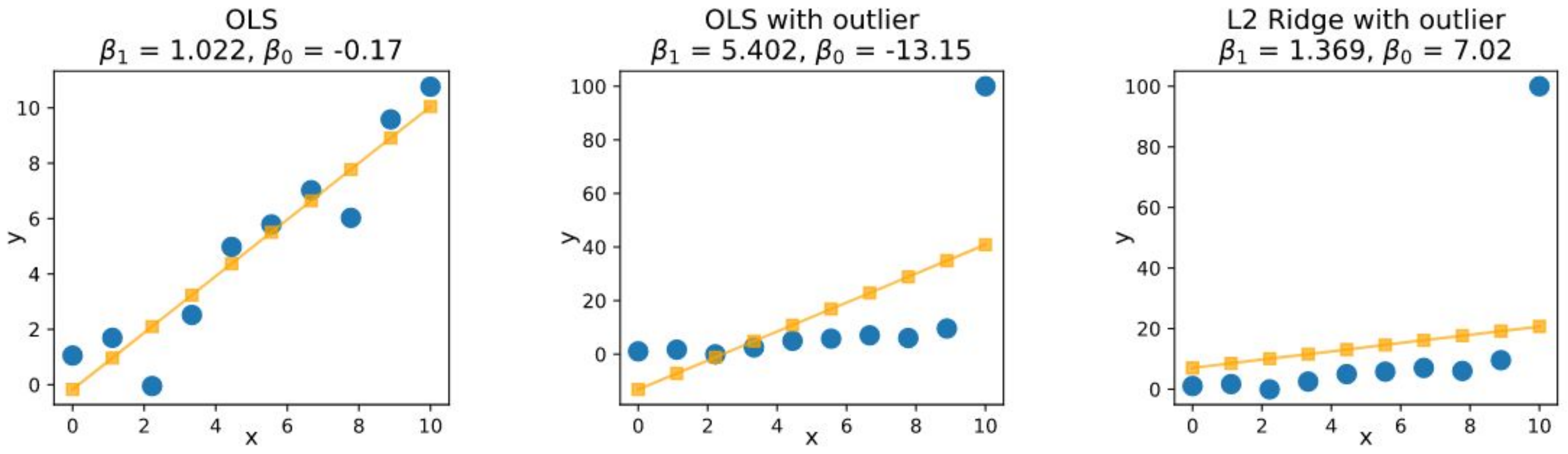
¿Cómo funciona?

1. plantear y definir q.
2. settear un vector de λ's ($\lambda \uparrow \Rightarrow$ más penalidad)
3. optimizamos con un λ_i $\rightarrow$ $RSS(\lambda; q) = \|Y - X\beta\|^2 + \lambda \|\beta\|_q$
4. Calculamos métricas (bondad de ajuste, calidad de la información)
5. Comparamos y elegimos el mejor.



Regularización - Motivación

$(\beta_0 + \beta_1)^2 \leq \lambda$



$\lambda = 1,45$

Regularización

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

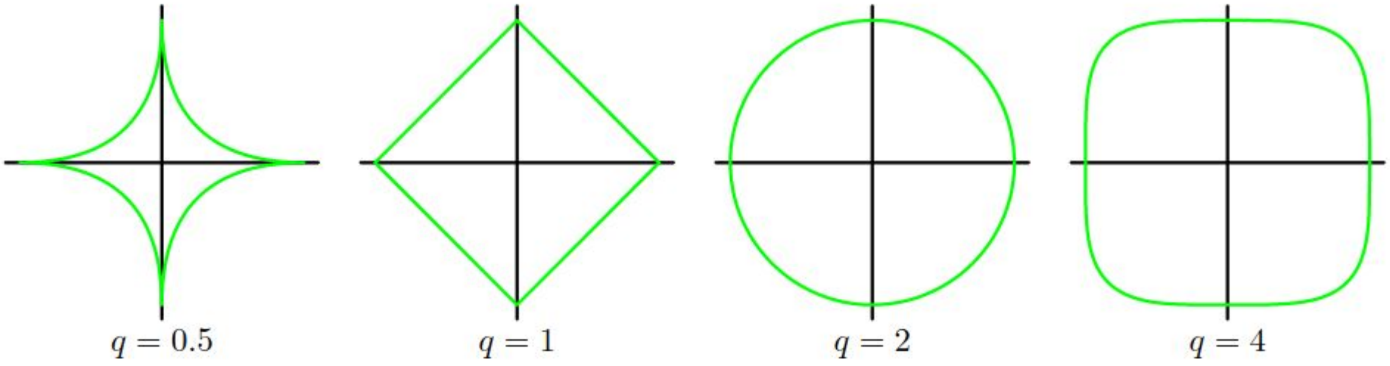
Observado - Predicción

w está "libre"

Regularización

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

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



Lasso

Ridge

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

Maximum A Posteriori como regularización

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

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

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

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

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

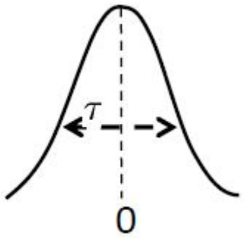
Gaussian prior con varianza b2

Maximum A Posteriori como regularización - Ridge (L2)

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

I) Gaussian Prior

$$\beta \sim \mathcal{N}(0, \tau^2\mathbf{I}) \qquad p(\beta) \propto e^{-\beta^T\beta/2\tau^2}$$



$$\widehat{\beta}_{\mathrm{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \underbrace{\lambda}_{\text{constant}(\sigma^2, \tau^2)} \|\beta\|_2^2$$

Ridge Regression

$$\widehat{\beta}_{\mathrm{MAP}} = (\boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{I})^{-1} \boldsymbol{A}^\top \boldsymbol{Y}$$

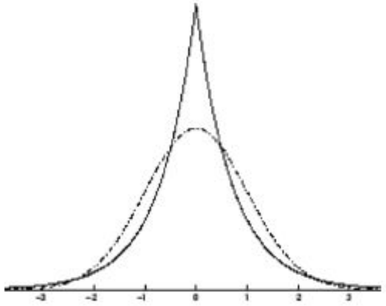
Machine Learning 10-315, Carnegie Mellon

Maximum A Posteriori como regularización - LASSO (L1)

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

II) Laplace Prior

$$\beta_i \overset{iid}{\sim} \text{Laplace}(0, t) \qquad\qquad p(\beta_i) \propto e^{-|\beta_i|/t}$$



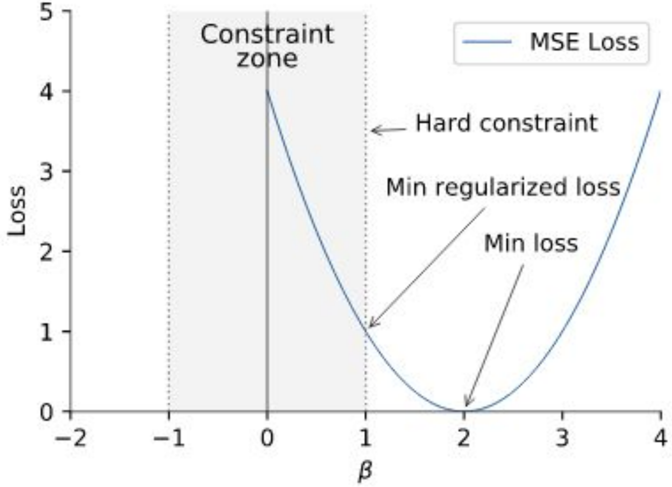
$$\widehat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \underset{\substack{\downarrow \\ \text{constant}(\sigma^2, t)}}{\lambda} \|\beta\|_1$$

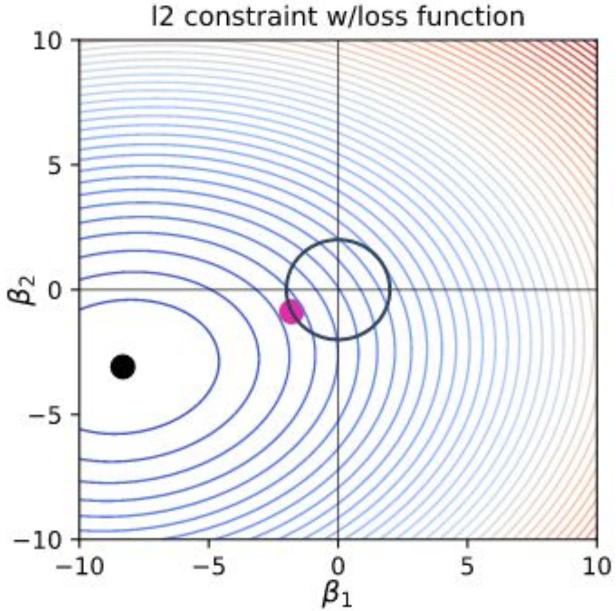
Lasso

Machine Learning 10-315, Carnegie Mellon

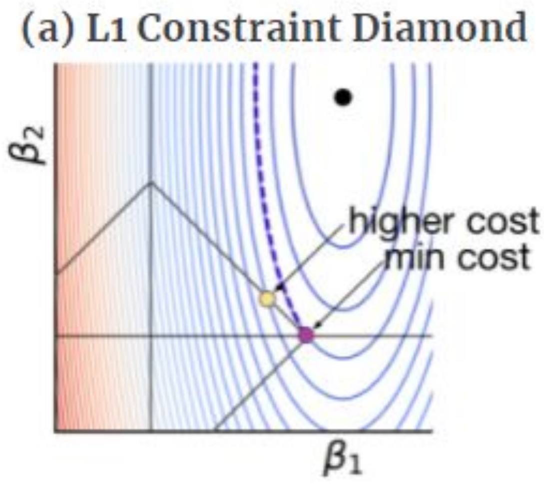
.UBAfiuba FACULTAD DE INGENIERÍA

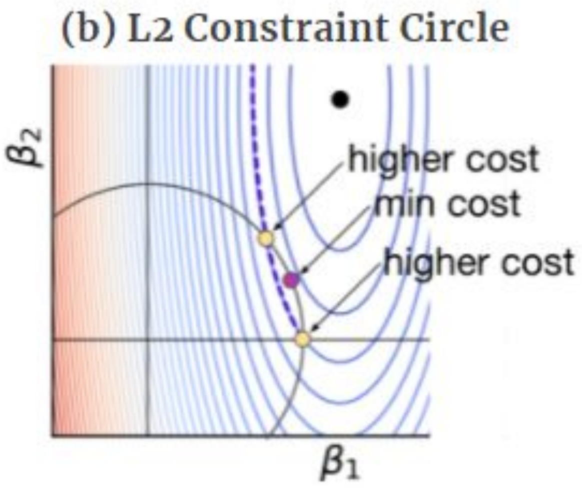
Regularización





Regularización



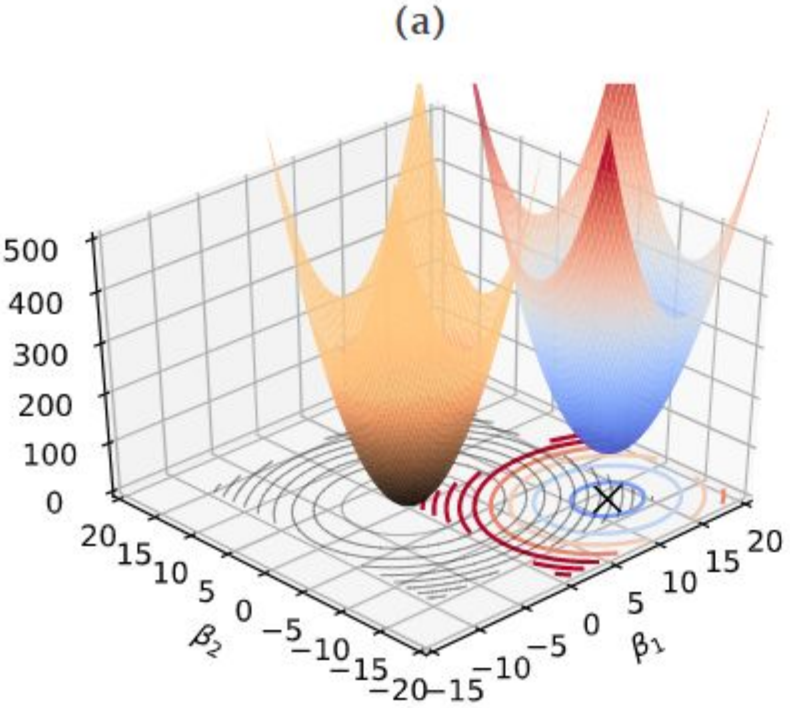


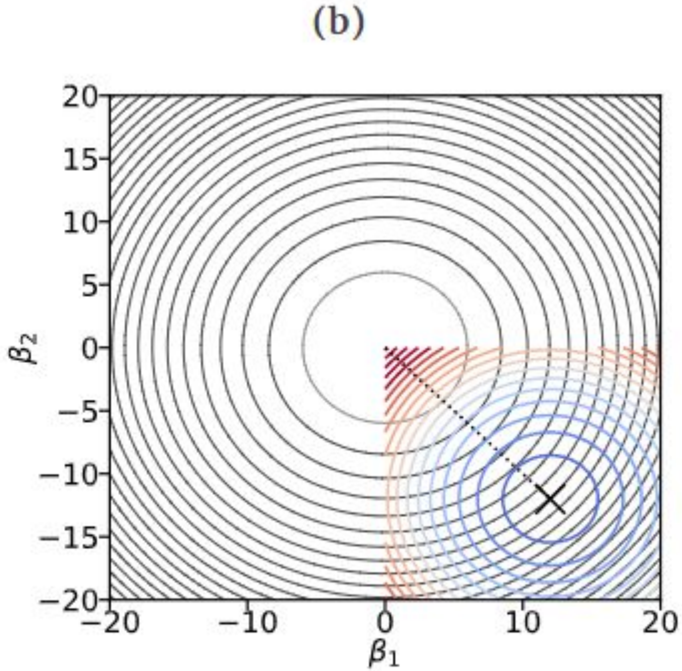
ElasticNet

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

¿Qué β se reduce más?

Regularización





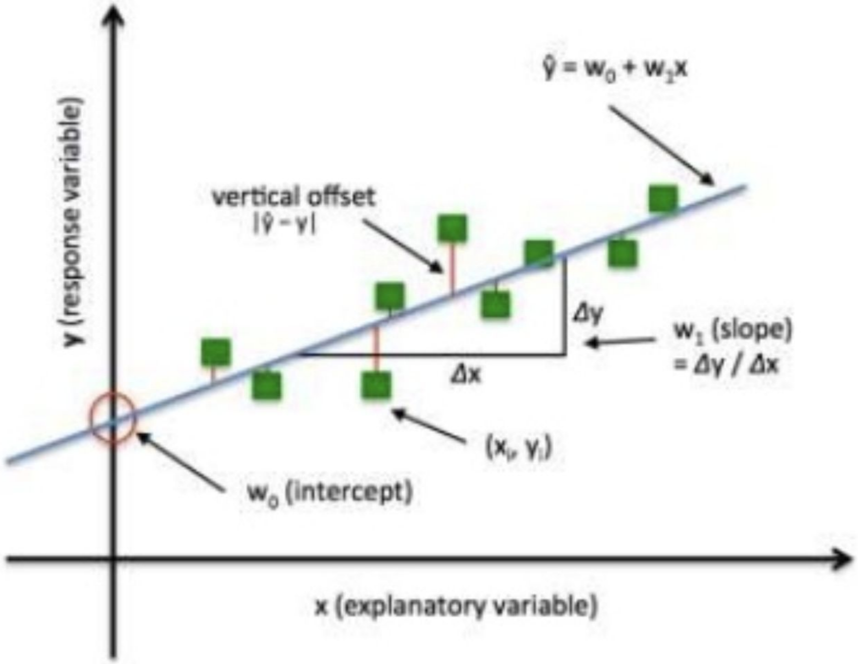
Gradiente Descendente

Implementación de Gradiente Descendente

Solucion analitica

$$\min_{W} \|Y - XW\|_2^2$$

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

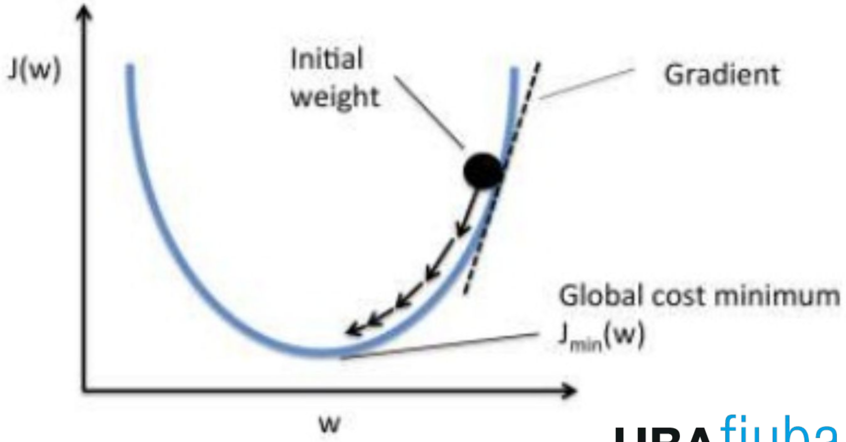


Implementación de Gradiente Descendente

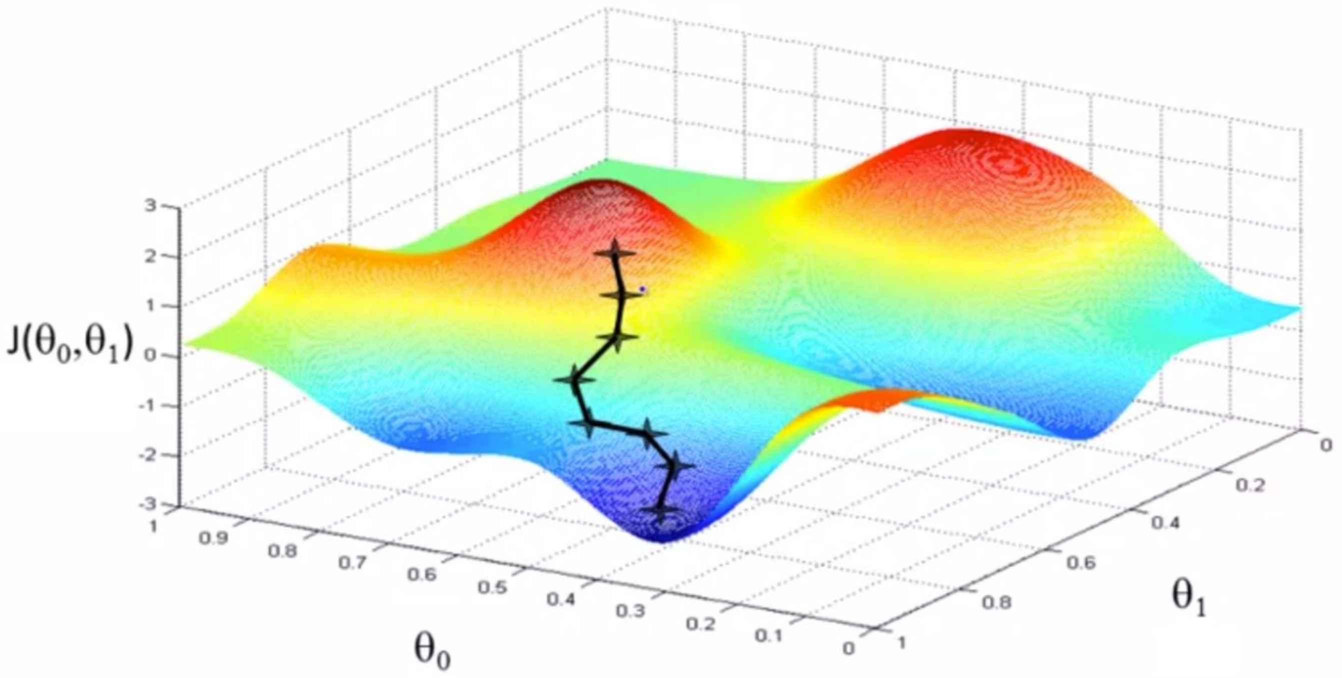
Solución numérica

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

$$W \longleftarrow W - \underbrace{\alpha}_{\text{Learning rate}} \nabla \left(\sum_i (y_i - X_i \cdot W)^2 \right)$$

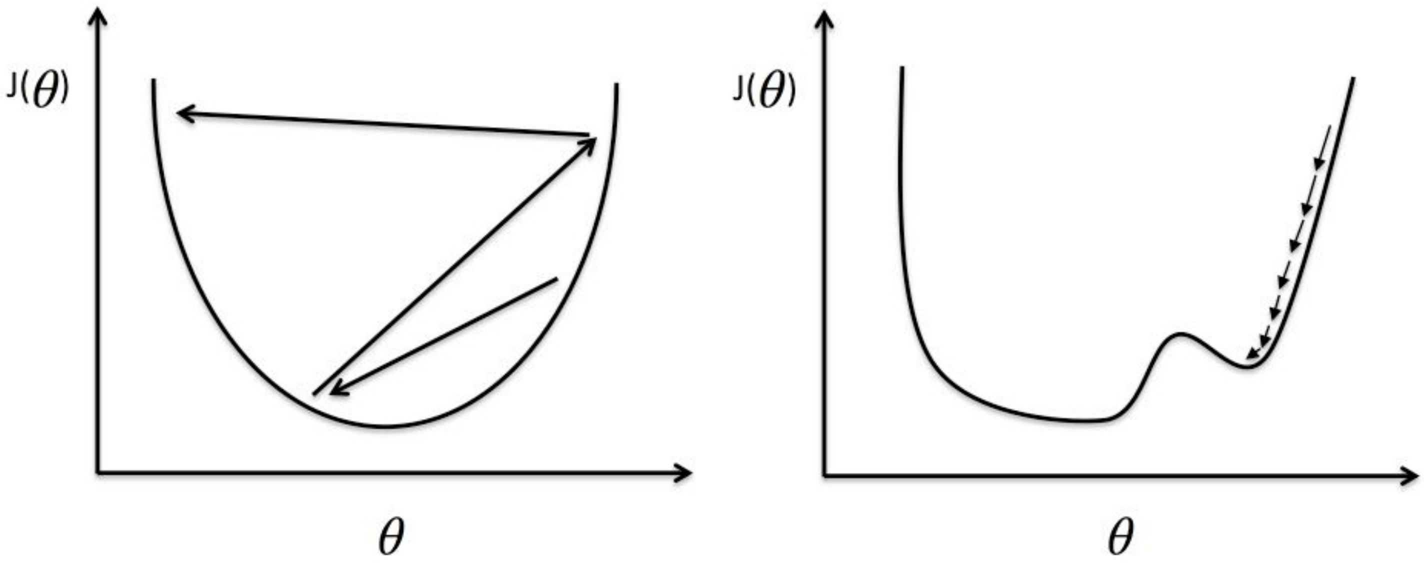


Gradiente Descendente



Andrew Ng

Gradiente Descendente



Large learning rate: Overshooting.

Small learning rate: Many iterations until convergence and trapping in local minima.

Implementación de Gradiente Descendente

Solución numérica

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

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

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

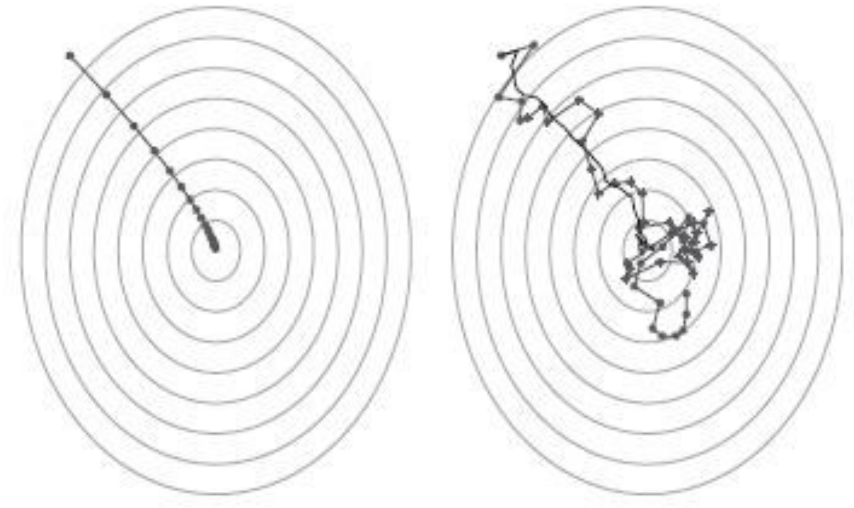
Stochastic Gradient Descent algorithm

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 Estocástico

Solución numérica



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

Gradient Descent

\sum_i \Big(-2 (y_i - \hat{y}_i) x_{ij}
\Big)

Comparativa de gradientes

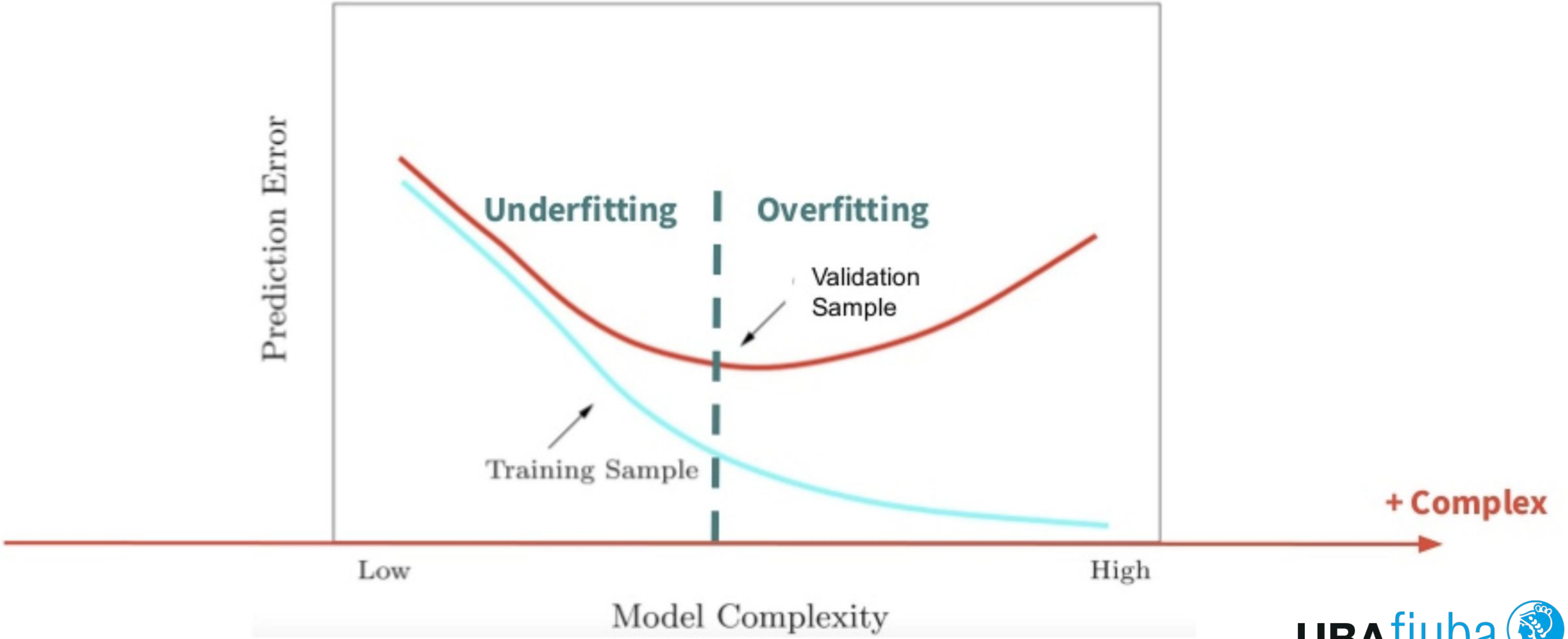
$$\sum \Big(-2 \, (y_i - \hat{y}_i) \, x_{ij} \Big)$$

	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

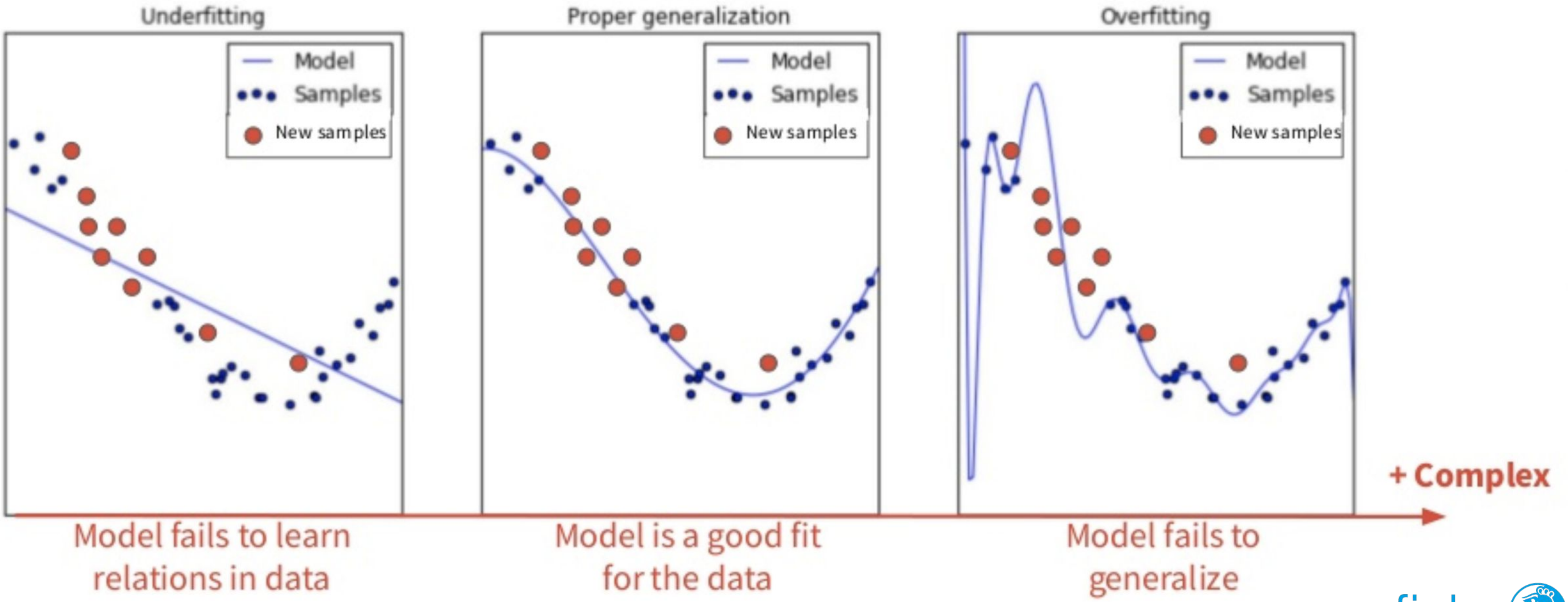
.UBAfiuba FACULTAD DE INGENIERÍA

Selección de modelos

Selección de modelos



Selección de modelos

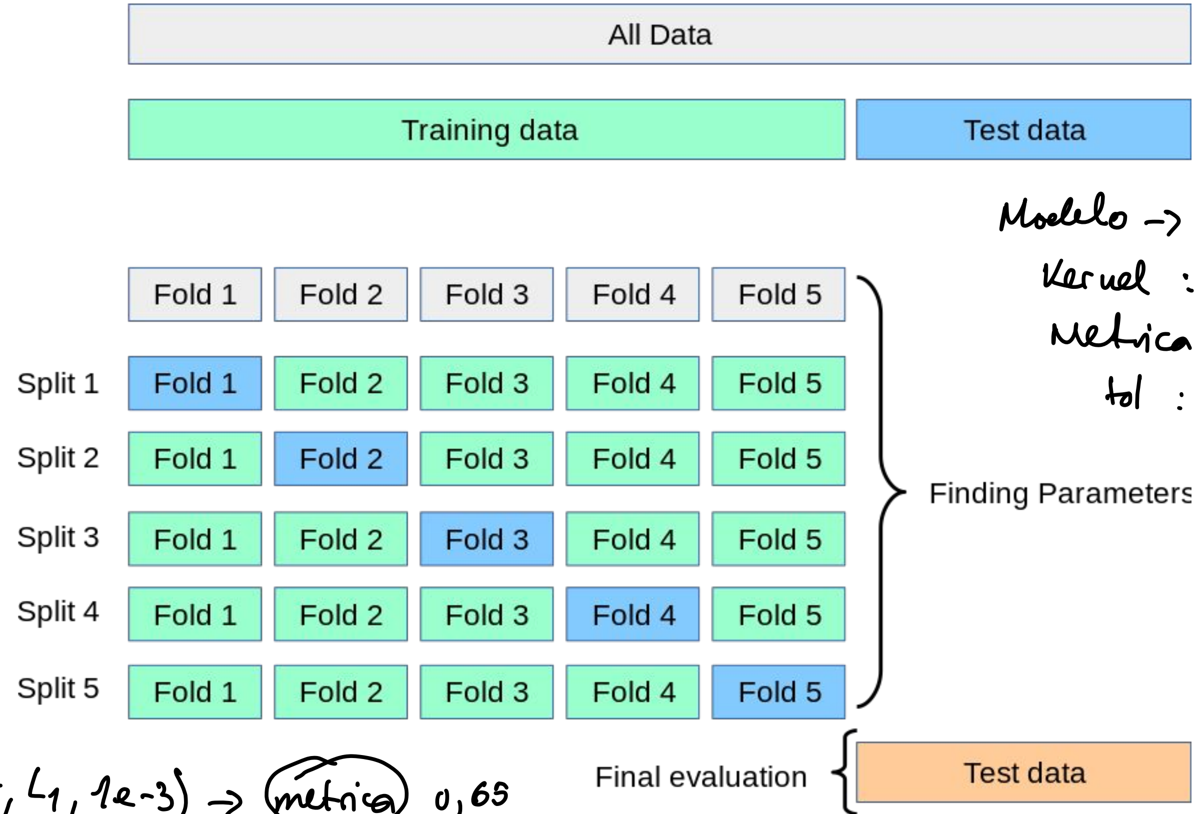


Entrenamiento de modelos - Cross-Validation

Cross-Validation

$\overline{Ac}$

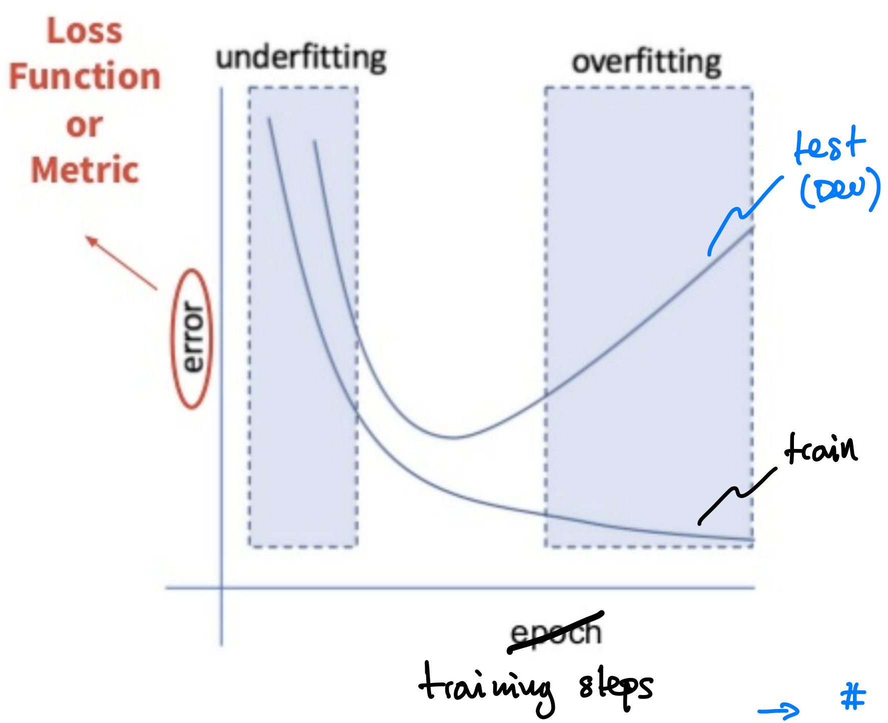
Ac_1
Ac_2
⋮
Ac_5



Modelo → SVM
Kernel : linear, rbf
Metrica : L_1, L_2
tol : 1e-3, 1e-4

M_1(SVM, Linear, L_1, 1e-3) → metrica 0,65
M_2(- - -) → metrica: 0,62

Entrenamiento numérico del modelo seleccionado - Obtención de parámetros



Mini-Batch Gradient Descent

for epoch in n_epochs:

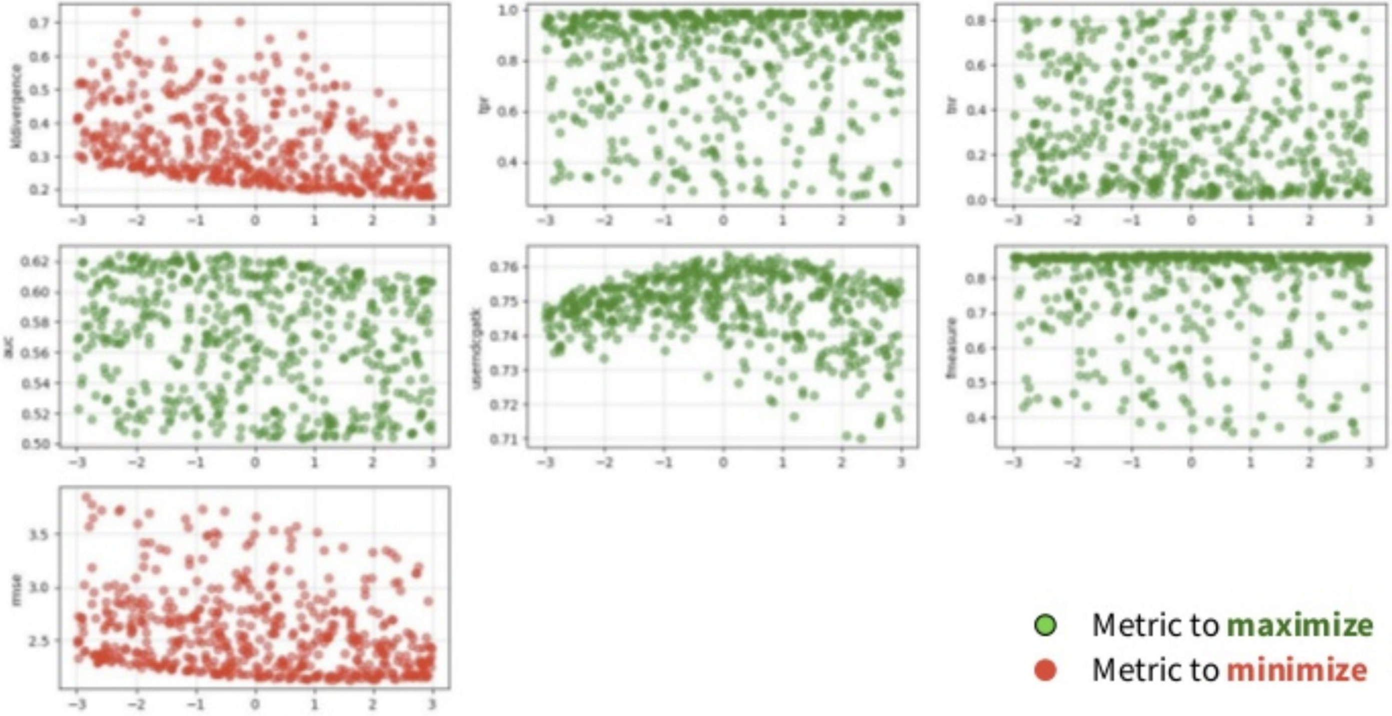
- shuffle the batches
- **for batch in n_batches:**
  - compute the predictions for **the batch**
  - compute the error for the batch
  - compute the gradient for **the batch**
  - update the parameters of the model
- plot error vs epoch

→ early stop → monitoreo del error de test.

→ # training steps fijos → Análisis a posterior

Entrenamiento de modelos - Hiper parámetros

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