

Lecture 3: Regularizacion y Clasificacion

Aprendizaje y Minería de Datos para los Negocios

Ignacio Sarmiento-Barbieri

Universidad de los Andes

October 25, 2021

Agenda

- 1 Recap
 - Predicción
 - Overfit
- 2 Regularización
 - Lasso
 - Ridge
- 3 Classification
 - K vecinos cercanos (KNN)
 - Logit
- 4 Break
- 5 R para ML

Predicción y Error Predictivo

- ▶ El objetivo es predecir y dadas otras variables X .
- ▶ Asumimos que el link entre y and X esta dado por el modelo:

$$y = f(X) + u \quad (1)$$

- ▶ donde $f(X)$ es cualquier función,
- ▶ u una variable aleatoria no observable $E(u) = 0$ and $V(u) = \sigma^2$

Predicción y Error Predictivo

- ▶ En la práctica no conocemos $f(X)$
- ▶ Es necesario estimarla $\hat{y} = \hat{f}(X)$
- ▶ La medida de cuan bien funciona nuestro modelo es

$$MSE(y) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

- ▶ Notemos que podemos descomponer el MSE en dos partes

$$MSE(y) = MSE(\hat{f}) + \sigma^2 \quad (3)$$

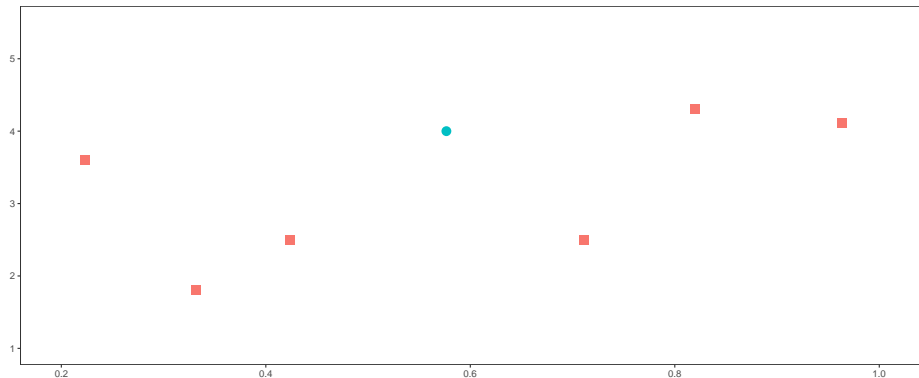
$$= Bias^2(\hat{f}) + V(\hat{f}) + \sigma^2 \quad (4)$$

- ▶ el error de estimar f con \hat{f} . (*reducible*)
- ▶ el error de no observar u, σ^2 . (*irreducible*)
- ▶ dilema entre sesgo y varianza

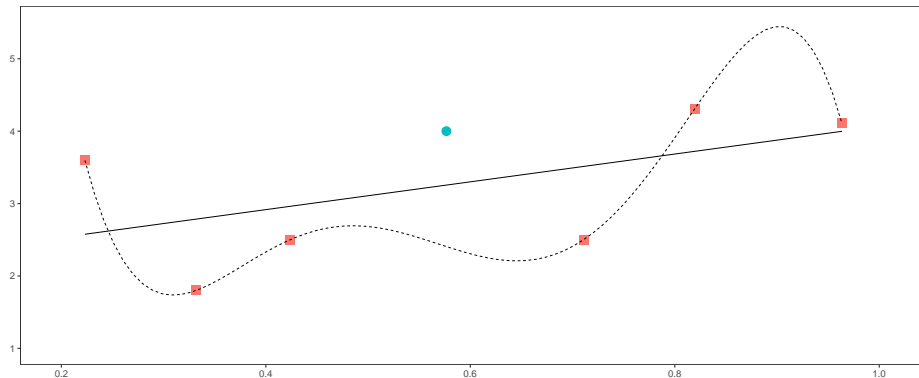
Overfit y Predicción fuera de Muestra

- ▶ ML nos interesa la predicción fuera de muestra
- ▶ Overfit: modelos complejos predicen muy bien dentro de muestra, pero tienden a hacer un trabajo fuera de muestra

Overfit



Overfit



Overfit y Predicción fuera de Muestra

- ▶ ML nos interesa la predicción fuera de muestra
- ▶ Overfit: modelos complejos predicen muy bien dentro de muestra, pero tienden a hacer un trabajo fuera de muestra
- ▶ Hay que elegir el modelo que “mejor” prediga
 - ▶ Medidas Clásicas

$$AIC(j) = \log \left(\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right) - p_j \quad (5)$$

$$SIC(j) = \log \left(\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right) - p_j \frac{1}{2} \log(n) \quad (6)$$

- ▶ Métodos de Remuestreo
 - ▶ Validación cruzada en K-partes (5 o 10)
 - ▶ Usar el MSE

Regularización

Lasso

- Para un $\lambda \geq 0$ dado, consideremos el siguiente problema de optimización

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - \beta_0 - x_{i1}\beta_1 - \cdots - x_{ip}\beta_p)^2 + \lambda \sum_{j=1}^p |\beta_j| \quad (7)$$

Lasso

- ▶ Para un $\lambda \geq 0$ dado, consideremos el siguiente problema de optimización

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - \beta_0 - x_{i1}\beta_1 - \cdots - x_{ip}\beta_p)^2 + \lambda \sum_{j=1}^p |\beta_j| \quad (7)$$

- ▶ “LASSO’s free lunch”: selecciona automáticamente los predictores que van en el modelo ($\beta_j \neq 0$) y los que no ($\beta_j = 0$)
- ▶ Porque? Los coeficientes que no van son soluciones de esquina
- ▶ $L(\beta)$ es no differentiable

Lasso Intuición en 1 Dimension

► Lasso Intuición

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (8)$$

► Un solo predictor, un solo coeficiente

► Si $\lambda = 0$

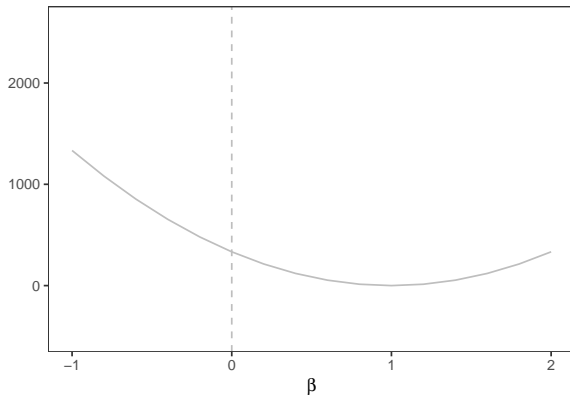
$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 \quad (9)$$

► y la solución es

$$\hat{\beta}_{OLS} \quad (10)$$

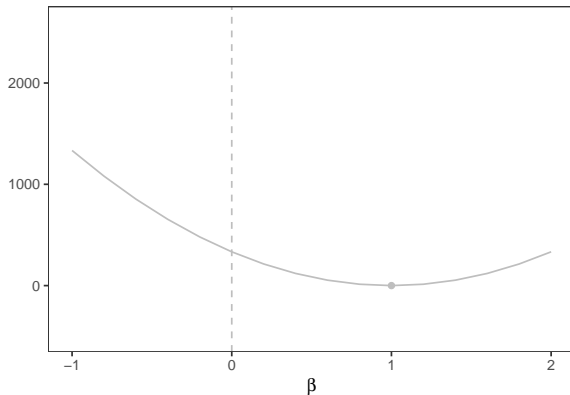
Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (11)$$



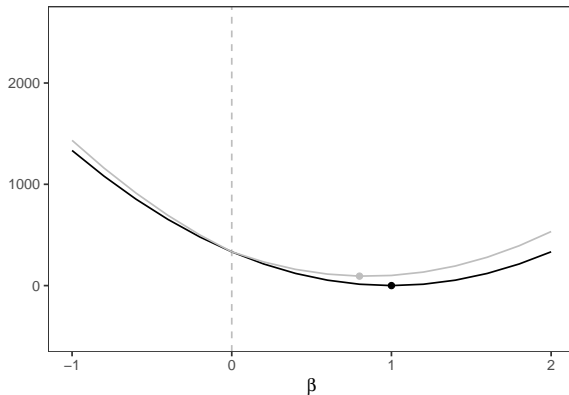
Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (12)$$



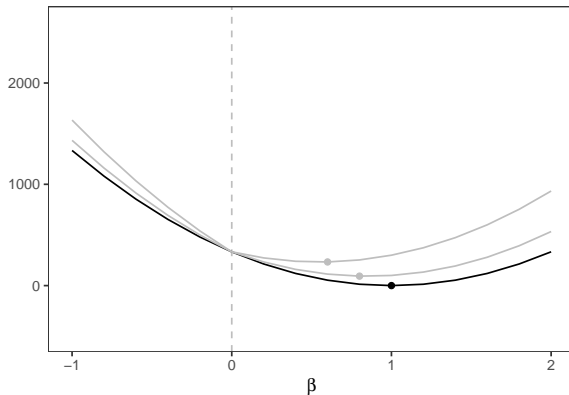
Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (13)$$



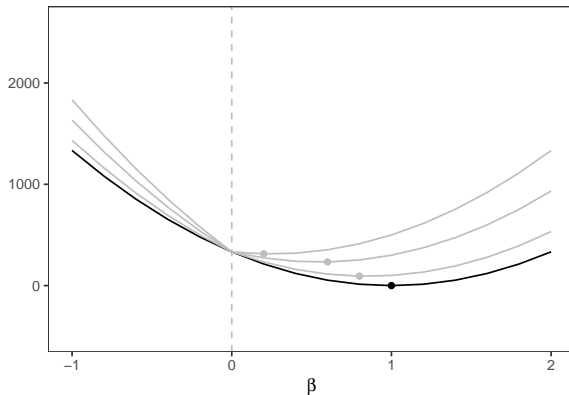
Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (14)$$



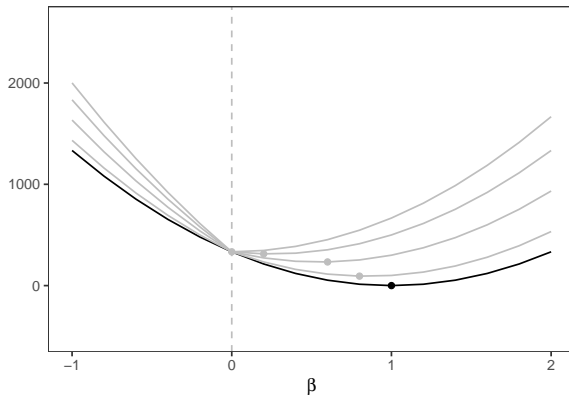
Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (15)$$



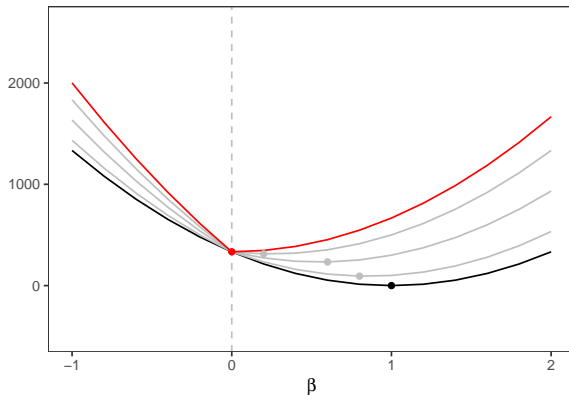
Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (16)$$



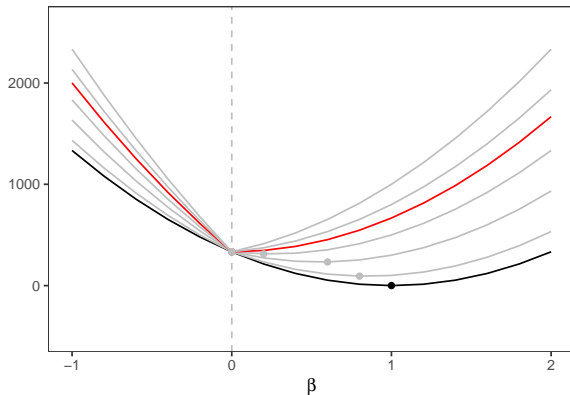
Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (17)$$



Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (18)$$



Intuición en 1 Dimension

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (19)$$

la solución analítica es

$$\hat{\beta}_{lasso} = \begin{cases} 0 & \text{si } \lambda \geq \lambda^* \\ \hat{\beta}_{OLS} - \frac{\lambda}{2} & \text{si } \lambda < \lambda^* \end{cases} \quad (20)$$

Ridge

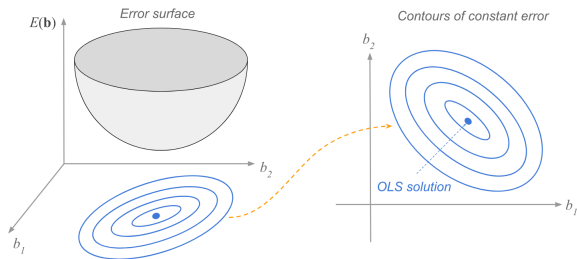
- ▶ Para un $\lambda \geq 0$ dado, consideremos ahora el siguiente problema de optimización

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - \beta_0 - x_{i1}\beta_1 - \cdots - x_{ip}\beta_p)^2 + \lambda \sum_{j=1}^p (\beta_j)^2 \quad (21)$$

- ▶ La intuición es similar a lasso, pero la vamos a extender a 2-Dim

Intuición en 2 Dimensiones (OLS)

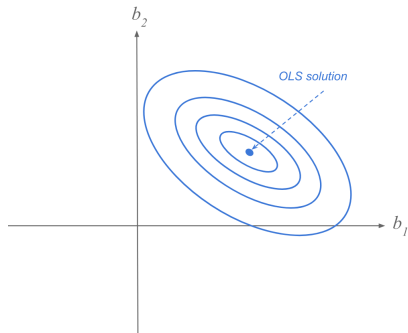
$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 \quad (22)$$



Fuente: <https://allmodelsarewrong.github.io>

Intuición en 2 Dimensiones (OLS)

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 \quad (23)$$



Fuente: <https://allmodelsarewrong.github.io>

Intuición en 2 Dimensiones (Ridge)

- ▶ Al problema

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - \beta_0 - x_{i1}\beta_1 - \cdots - x_{ip}\beta_p)^2 + \lambda \sum_{j=1}^p (\beta_j)^2 \quad (24)$$

- ▶ podemos escribirlo como

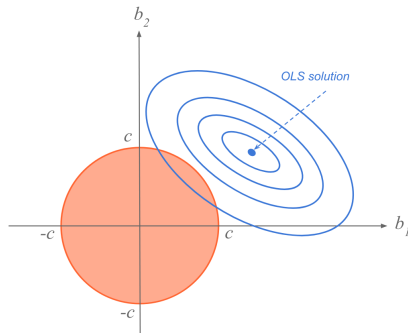
$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i1}\beta_2)^2 \quad (25)$$

sujeto a

$$((\beta_1)^2 + (\beta_2)^2) \leq c$$

Intuición en 2 Dimensiones (Ridge)

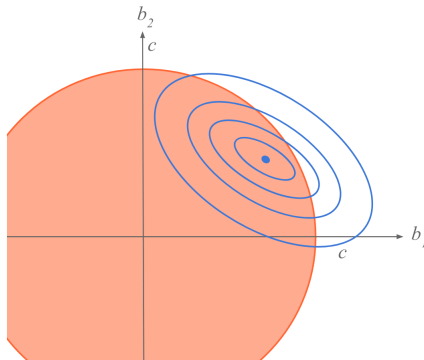
$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 \text{ s.a. } ((\beta_1)^2 + (\beta_2)^2) \leq c \quad (26)$$



Fuente: <https://allmodelsarewrong.github.io>

Intuición en 2 Dimensiones (Ridge)

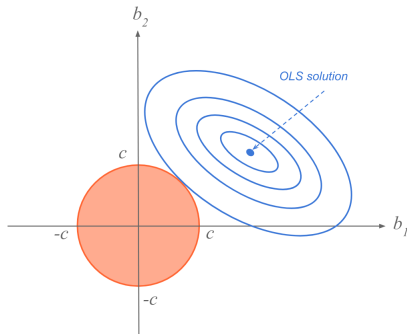
$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 \text{ s.a. } ((\beta_1)^2 + (\beta_2)^2) \leq c \quad (27)$$



Fuente: <https://allmodelsarewrong.github.io>

Intuición en 2 Dimensiones (Ridge)

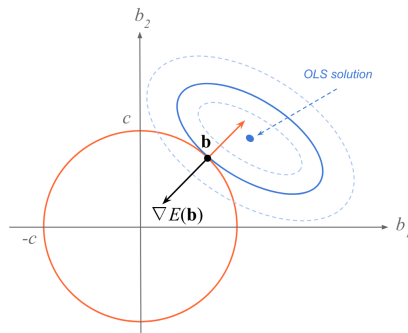
$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 \text{ s.a } ((\beta_1)^2 + (\beta_2)^2) \leq c \quad (28)$$



Fuente: <https://allmodelsarewrong.github.io>

Intuición en 2 Dimensiones (Ridge)

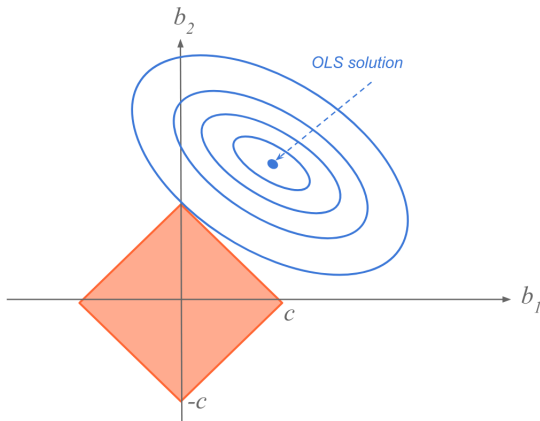
$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 \text{ s.a. } ((\beta_1)^2 + (\beta_2)^2) \leq c \quad (29)$$



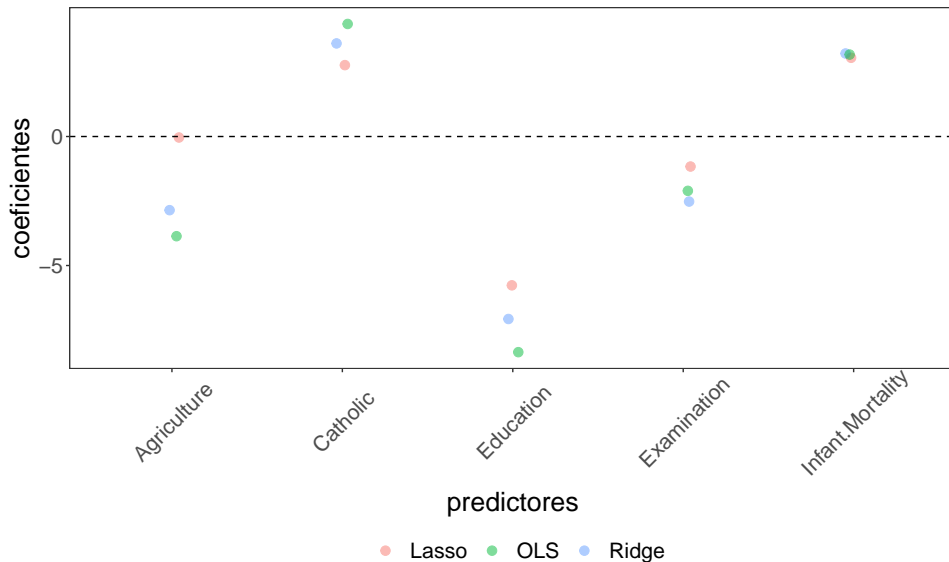
Fuente: <https://allmodelsarewrong.github.io>

Intuición en 2 Dimensiones (Lasso)

$$\min_{\beta} E(\beta) = \sum_{i=1}^n (y_i - x_{i1}\beta_1 - x_{i2}\beta_2)^2 \text{ s.a } (|\beta_1| + |\beta_2|) \leq c \quad (30)$$



Lasso y Ridge Ejemplo



Comentarios técnicos

- ▶ Lasso y ridge son sesgados, pero las disminuciones en varianza pueden compensar esto y llevar a un MSE menor
- ▶ Lasso encoje a cero, Ridge no tanto
- ▶ Importante para aplicación:
 - ▶ Estandarizar los datos (media 0, y varianza 1)
 - ▶ Como elegimos λ ?

Comentarios técnicos: selección de λ

- ▶ Como elegimos λ ?
- ▶ λ es un parámetro y lo elegimos usando validación cruzada
 - 1 Partimos la muestra de entrenamiento en K Partes: $M_{train} = M_{fold 1} \cup M_{fold 2} \cdots \cup M_{fold K}$
 - 2 Cada conjunto $M_{fold K}$ va a jugar el rol de una muestra de evaluación $M_{eval k}$. Entonces para cada muestra
 - ▶ $M_{train-1} = M_{train} - M_{fold 1}$
 - ▶ \vdots
 - ▶ $M_{train-k} = M_{train} - M_{fold k}$
 - 3 Luego hacemos el siguiente loop
 - 1 Para $\lambda_i = 0, 0.001, 0.002, \dots, \lambda_{max}$
 - Para $k = 1, \dots, K$
 - Ajustar el modelo $m_{i,k}$ con λ_i en $M_{train-k}$
 - Calcular y guardar el $MSE(m_{i,k})$ usando M_{eval-k}
 - fin para k
 - Calcular y guardar $MSE_i = \frac{1}{K} MSE(m_{i,k})$
 - 2 fin para λ
 - 4 Encontrar el menor MSE_i y usar ese $\lambda_i = \lambda^*$

Classification

Classification: Motivation

- ▶ Muchas de las preguntas predictivas suelen ser de clasificación

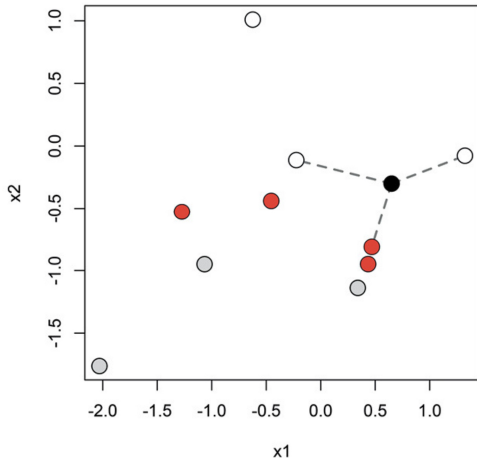
Classification: Motivation

- ▶ Muchas de las preguntas predictivas suelen ser de clasificación
- ▶ Predecir si un usuario va a hacer click o no
- ▶ Decidir si alguien va a “default” el credito
- ▶ La afiliación política basado en los discursos
- ▶ La diferencia es que ahora y representa la pertenencia a una clase
- ▶ $y \in \{0, 1, \dots, m\}$

La pregunta es: dado un nuevo conjunto de predictores X es nuestra mejor predicción de la categoría a la que pertenece

K vecinos cercanos (KNN)

- K vecinos cercanos predice la clase \hat{y} a partir de x preguntandose *Cual es la clase mas común para la observación alrededor x ?*



Source: Taddy (2019)

K vecinos cercanos (KNN)

- ▶ K vecinos cercanos predice la clase \hat{y} a partir de x preguntandose *Cual es la clase mas común para la observación alrededor x ?*
- ▶ Algoritmo: dado el insumo x_f donde nos gustaría la clase:
 - ▶ Encontrar los K vecnos mas cercanos, donde cercanos lo definimos a partir de una distancia, la más común es la euclideana

$$d(x_i, x_f) = \sqrt{\sum_{j=1}^p (x_{ij} - x_{fj})^2} \quad (31)$$

- ▶ Esto nos da los K vecinos cercanos:

$$[x_{i1}, y_{i1}], \dots, [x_{iK}, y_{iK}] \quad (32)$$

- ▶ La clase predicha es la más común

$$\hat{y}_f = mode\{y_{i1}, \dots, y_{iK}\} \quad (33)$$

K vecinos cercanos (KNN)

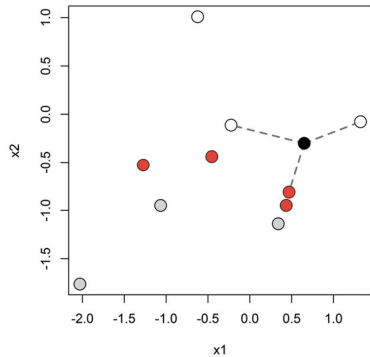
- Hay algunos problemas en aplicaciones
 - Las predicciones son inestables como función de K

$$K = 1 \implies \hat{p}(\text{white}) = 0$$

$$K = 2 \implies \hat{p}(\text{white}) = 1/2$$

$$K = 3 \implies \hat{p}(\text{white}) = 2/3$$

$$K = 4 \implies \hat{p}(\text{white}) = 1/2$$



Source: Taddy (2019)

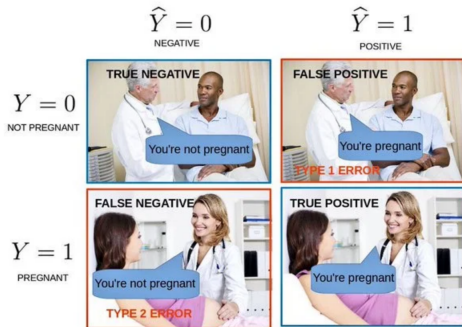
K vecinos cercanos (KNN)

- ▶ Hay algunos problemas en aplicaciones
 - ▶ Las predicciones son inestables como función de K
 - ▶ Esto hace que sea muy difícil elegir el K óptimo y validación cruzada no funciona muy bien
 - ▶ Dado que cada predicción x requiere un proceso de cuenta computacionalmente muy costoso, KNN no es muy útil cuando trabajamos con Big Data
 - ▶ KNN es una buena idea, pero demasiado crudo para ser útil en la práctica

Probabilidades, Costos, y Clasificación

Probabilidades, Costos, y Clasificación

- ▶ Dos estados de la naturaleza $y \rightarrow n \in \{0, 1\}$
- ▶ Dos acciones $(\hat{y}) \rightarrow a \in \{0, 1\}$
- ▶ Los aciertos y errores que cometemos al clasificar podemos resumirlos en



Source: <https://dzone.com/articles/understanding-the-confusion-matrix>

Probabilidades, Costos, y Clasificación

- ▶ Dos estados de la naturaleza $y \rightarrow n \in \{0, 1\}$
- ▶ Dos acciones $(\hat{y}) \rightarrow a \in \{0, 1\}$
- ▶ Probabilities
 - ▶ $p = Pr(y = 1|X)$
 - ▶ $1 - p = Pr(y = 0|X)$
- ▶ Las acciones tienen costos
- ▶ y estas puede haber distintos costos asociados

Probabilidades, Costos, y Clasificación

- ▶ Para tomar una decisión óptima, necesitamos estimar las probabilidades de los resultados posibles
- ▶ Estas probabilidades son las que permiten evaluar la pérdida esperada
- ▶ La pérdida esperada por tomar la acción a es

$$E[Loss(a)] = \sum_n p_n L(a, n) \quad (34)$$

- ▶ Al conocer las probabilidades de los distintos resultados podemos evaluar la pérdida esperada

Probabilidades, Costos, y Clasificación

- ▶ Afortunadamente sabemos como estimar probabilidades
 - ▶ K-vecinos cercanos
 - ▶ Modelo logístico
 - ▶ entre otras

Logit

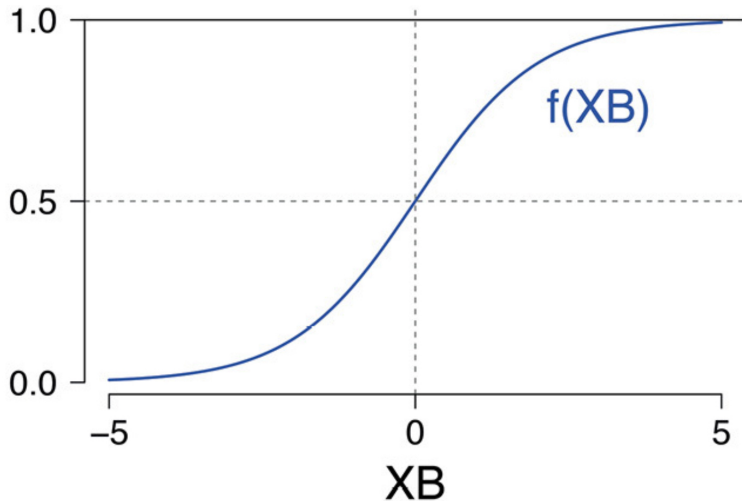
- Podemos modelar la relación entre $p(X)$ y X

$$Pr(y = 1|X) = f(X'\beta) \quad (35)$$

- usando la función logística (sigmoid, softmax)

$$p(y = 1|X) = \frac{e^{X'\beta}}{1 + e^{X'\beta}} = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)} \quad (36)$$

Logit



Source: Taddy (2019)

Logit

- ▶ Tenemos la probabilidad condicional

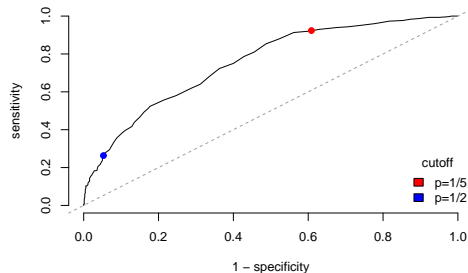
$$Pr(y = 1|X) = f(X'\beta) \quad (37)$$

- ▶ Estimamos los parámetros usando MLE
- ▶ Podemos recobrar fácilmente las predicciones:

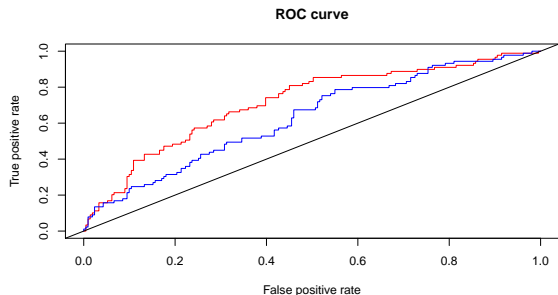
$$p(y = 1|X) = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p)} \quad (38)$$

ROC

- ▶ ROC ilustra el trade-off de las reglas de clasificación
- ▶ ROC nos da el *locus* de los *TPR* y *FPR* para todos los posibles $c \in [0, 1]$
 - ▶ *Sensibilidad*: Tasas de Verdaderos Positivos
 - ▶ *1-Especificidad*: Tasa de Falsos Positivos
- ▶ Nos da la habilidad
 - ▶ Medir la capacidad predictiva del modelo
 - ▶



- ▶ ROC ilustra el trade-off de las reglas de clasificación
- ▶ ROC nos da el *locus* de los *TPR* y *FPR* para todos los posibles $c \in [0, 1]$
 - ▶ *Sensibilidad*: Tasas de Verdaderos Positivos
 - ▶ *1-Especificidad*: Tasa de Falsos Positivos
- ▶ Nos da la habilidad
 - ▶ Medir la capacidad predictiva del modelo
 - ▶ Comparar entre modelos



Revisión

Hoy

- ▶ Regularización
 - ▶ Lasso
 - ▶ Ridge
- ▶ Clasificación
 - ▶ Vecinos Cercanos
 - ▶ Logit
 - ▶ ROC

Volvemos en 5 mins con R

R para ML



photo from <https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/>