

# Árboles de Regresión y Clasificación

Álvaro Riascos  
Mónica Ribero

Universidad de Los Andes

February 23, 2016

# Contenido

Árboles de  
Regresión y  
Clasificación

Álvaro Riascos  
Mónica Ribero

Introducción

Árboles de Regresión

Árboles de Clasificación

Conclusiones

Introducción

Árboles de  
Regresión

Árboles de  
Clasificación

Conclusiones

# Introducción

- ▶ Se tiene un número  $n$  de observaciones (conjunto de entrenamiento)

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$$

$$\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{ip})$$

$$y_i = \text{Variable de Respuesta}$$

- ▶ Se tiene un número  $n$  de observaciones (conjunto de entrenamiento)

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$$

$$\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{ip})$$

$$y_i = \text{Variable de Respuesta}$$

- ▶ Crear modelo que, dado  $x_k$ , prediga el valor de  $y_k$ .

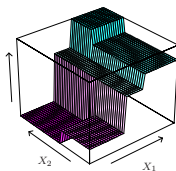
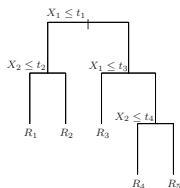
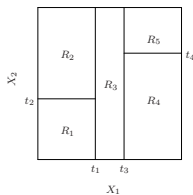
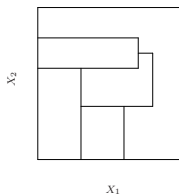
- ▶ Se tiene un número  $n$  de observaciones (conjunto de entrenamiento)

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$$
$$\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{ip})$$
$$y_i = \text{Variable de Respuesta}$$

- ▶ Crear modelo que, dado  $x_k$ , prediga el valor de  $y_k$ .
- ▶ Si  $y$  es categórica se habla de clasificación, de lo contrario de regresión.

# Planteamiento

- Suponga que se tiene una partición  $R_1, \dots, R_m$  del espacio de  $\mathbf{x}$



- Suponga que se tiene una partición  $R_1, \dots, R_m$  del espacio de  $\mathbf{x}$



- ▶ Suponga que se tiene una partición  $R_1, \dots, R_m$  del espacio de  $\mathbf{x}$
- ▶ Suponga que para  $k = 1, \dots, m$  tiene una constante  $c_k$  que aproxima *bien* el valor de  $y_k$  para  $\mathbf{x}_k \in R_k$

- ▶ Suponga que se tiene una partición  $R_1, \dots, R_m$  del espacio de  $\mathbf{x}$
- ▶ Suponga que para  $k = 1, \dots, m$  tiene una constante  $c_k$  que aproxima *bien* el valor de  $y_k$  para  $x_k \in R_k$
- ▶ Se puede utilizar el predictor

$$f(x) = \sum_{j=1}^m c_j I_{\{x \in R_j\}}(x)$$

## Procedimiento

1. Determinar las regiones  $R_k$
2. Determinar las constantes  $c_k$
3. Determinar el número de regiones  $m$

# Árboles de Regresión

- Para cada  $k = 1, \dots, m$  escoger la constante  $c_k$  más “cercana” a los  $y_i$  correspondientes a los  $\mathbf{x}_i$  en esa caja

$$\hat{c}_k = \min_c \sum_{i: \mathbf{x}_i \in R_k} (y_i - c)^2$$

- La solución para cada caja es el promedio

$$\hat{c}_k = \text{prom}(y_i | \mathbf{x}_i \in R_k)$$

# Determinación de los $R_k$

Árboles de  
Regresión y  
Clasificación

Álvaro Riascos  
Mónica Ribero

Introducción

Árboles de  
Regresión

Árboles de  
Clasificación

Conclusiones

- Separación binaria recursiva. Algoritmo “glotón” (Top down greedy algorithm).

# Determinación de los $R_k$

- ▶ Separación binaria recursiva. Algoritmo “glotón” (Top down greedy algorithm).
- ▶ Para cada variable  $X_j$  con  $j = 1, \dots, p$  y cada  $s \in \text{rango}(X_j)$  define

$$R_1(j, s) = \{X | X_j < s\}$$

$$R_2(j, s) = \{X | X_j \geq s\}$$

# Determinación de los $R_k$

- ▶ Separación binaria recursiva. Algoritmo “glotón” (Top down greedy algorithm).
- ▶ Para cada variable  $X_j$  con  $j = 1, \dots, p$  y cada  $s \in \text{rango}(X_j)$  define

$$R_1(j, s) = \{X | X_j < s\}$$

$$R_2(j, s) = \{X | X_j \geq s\}$$



- ▶ Se busca en cada paso

$$\min_{j,s} [\min_c \sum_{i: x_i \in R_1(j,s)} (y_i - c)^2 + \min_c \sum_{i: x_i \in R_2(j,s)} (y_i - c)^2] \quad (1)$$

- ▶ Para cada  $j$  es sencillo encontrar un  $s$ .
- ▶ Se encuentra  $s^*$  para cada  $j$  y se escoge la pareja  $(j^*, s^*)$  que minimiza el error.
- ▶ Continue el procedimiento.

# Determinar la Altura del Árbol

- ▶ Muy alto  $\rightarrow$  overfitting
- ▶ Muy bajo:  $\rightarrow$  No captura estructura
- ▶ Dos opciones:
  1. Separar un nodo solo si la reducción en el error es mayor que cierto margen  $t$ .
  2. Crear árbol *grande*  $T_0$  y luego podarlo

# Podar el Árbol

- ▶ Crear árbol *grande*  $T_0$  y luego podarlo.
- ▶ Grande: Hasta que haya máximo  $n_{min}$  observaciones en cada nodo terminal.
- ▶ Para determinar dónde podar el árbol se crea una medida para comparar árboles  $T \subset T_0$

- Se define para  $T \subset T_0$  y cada región  $R_k$ ,  $k = 1, \dots, m$  determinada por un nodo terminal de  $T$  :

$$N_k = |\{x_i | x_i \in R_k\}|$$

$$Q_k(T) = \frac{1}{N_k} \sum_{i: x_i \in R_k} (y_i - \hat{c}_k)^2$$

$$C_\alpha(T) = \sum_{i=1}^{|T|} N_i Q_i(T) + \alpha |T|$$

- ▶  $\alpha$  es un parámetro que compensa costo de complejidad y el error de clasificación.
- ▶  $\alpha$  se estima con crossvalidación

- ▶  $\alpha$  es un parámetro que compensa costo de complejidad y el error de clasificación.
- ▶  $\alpha$  se estima con crossvalidación
- ▶  $T_\alpha$  es único y se encuentra usando “weakest link pruning”

# Árboles de Clasificación

- ▶ Para cada nodo  $k$ ,  $\hat{c}_k$  se escoge como la clase más común en ese nodo.
- ▶ Defina la proporción de la clase  $k$  en el nodo  $m$ :

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I_{\{y_i=k\}}$$



$$\hat{c}_m = \operatorname{argmax}_k p_{mk}$$



- Error de clasificación:

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} I_{\{y_i \neq c_m\}} = 1 - \hat{p}_{mk}$$

- Índice de Gini:

$$\sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- Cross-entropy:

$$-\sum_{k=1}^K p_{mk} \log p_{mk}$$

# Conclusiones

# Conclusiones

- ▶ Fáciles de interpretar y modelar
- ▶ Mala capacidad predictiva