

Árboles (CARTs)

Big Data y Machine Learning para Economía Aplicada

Ignacio Sarmiento-Barbieri

Universidad de los Andes

Agenda

1 Recap

2 Árboles

- Árboles de Regresión
- Árboles de Clasificación
- Sobreajuste

Recap

- Queremos predecir y en función de observables (\mathbf{x}_i)

$$y = f(\mathbf{x}_i) + u \quad (1)$$

- donde la estimación de f implica la que minimice el riesgo empírico (prediga mejor fuera de muestra):

$$\hat{f} = \underset{f}{\operatorname{argmin}} \left\{ \sum_{i=1}^n L(y_i, f(\mathbf{x}_i; \Theta)) \right\} \quad (2)$$

Recap

- Por ejemplo en regresión $f(x_i) = \mathbf{x}_i\beta$ y minimizamos

$$\hat{\beta} = \underset{\tilde{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^n (y_i - \mathbf{x}_i\beta)^2 \right\} \quad (3)$$

- Por ejemplo en clasificación logística $p(x_i) = \frac{1}{1+e^{-\mathbf{x}_i\beta}}$

$$\hat{\beta}^{MLE} = \underset{\beta}{\operatorname{argmin}} - \left[\sum_{i=1}^n \log \left(\frac{p_i}{1-p_i} \right)^{y_i} + \sum_{i=1}^n \log(1-p_i) \right] \quad (4)$$

Agenda

1 Recap

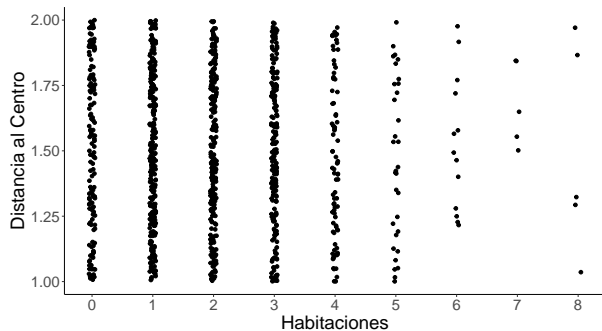
2 Árboles

- Árboles de Regresión
- Árboles de Clasificación
- Sobreajuste

Motivación

- Queremos predecir:

$$\text{Precio} = f(\text{habitaciones}, \text{Distancia al CBD}) \quad (5)$$



Motivación

- Podemos asumir f es lineal

$$f(\mathbf{x}_i; \beta) = \beta_0 + \beta_1 \text{Habitaciones}_i + \beta_2 \text{DCBD}_i + u_i \quad (6)$$

- o ajustar un modelo flexible e interpretable como son los arboles

$$f(x_i; \{R_j, \gamma_j\}_1^J) = T(x_i; \{R_j, \gamma_j\}_1^J) \quad (7)$$

Agenda

1 Recap

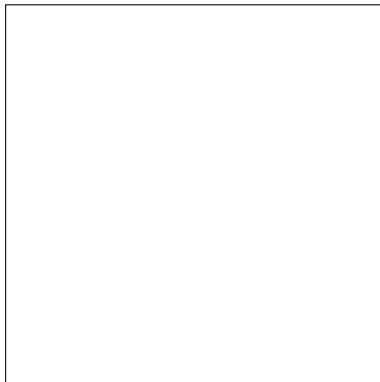
2 Árboles

- Árboles de Regresión
- Árboles de Clasificación
- Sobreajuste

¿Qué hacen?

“Recursive binary splitting”

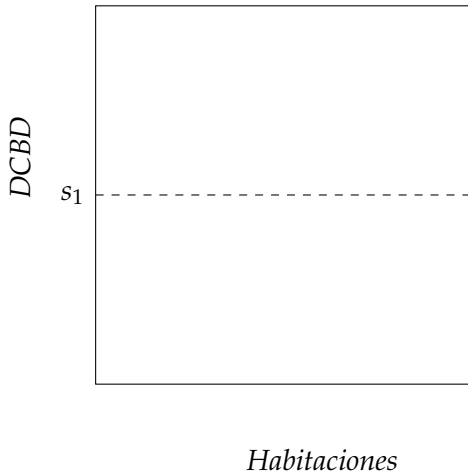
DCBD



Habitaciones

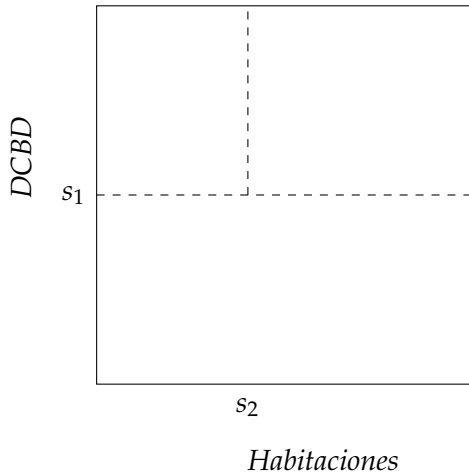
¿Qué hacen?

“Recursive binary splitting”

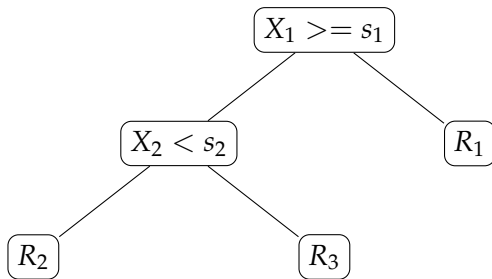


¿Qué hacen?

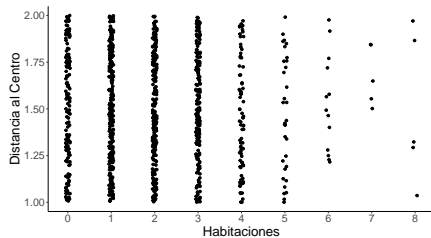
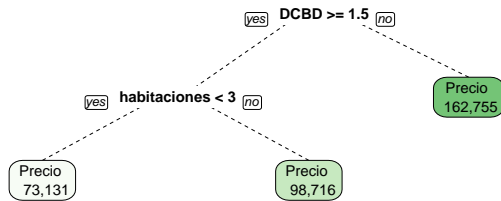
“Recursive binary splitting”



¿Qué hacen?



¿Qué hacen?



¿Cómo construimos un árbol?

- El problema de optimización

$$\hat{f} = \operatorname{argmin}_f \left\{ \sum_{i=1}^n L(y_i, f(\mathbf{x}_i; \Theta)) \right\} \quad (8)$$

- es ahora

$$\{\hat{R}_j, \hat{\gamma}_j\}_1^J = \operatorname{argmin}_{\{R_j, \gamma_j\}_1^J} \left\{ \sum_{i=1}^n L(y_i, T(\mathbf{x}_i; \{R_j, \gamma_j\}_1^J)) \right\} \quad (9)$$

¿Cómo construimos un árbol?

- ▶ Datos: $y_{n \times 1}$ y $X_{n \times k}$
- ▶ Definiciones
 - ▶ j es la variable que parte el espacio y s es el punto de partición
 - ▶ Defina los siguientes semiplanos

$$R_1(j, s) = \{X | X_j \leq s\} \ \& \ R_2(j, s) = \{X | X_j > s\} \quad (10)$$

- ▶ El problema: usando una “perdida cuadrática” buscar la variable de partición X_j y el punto s de forma tal que:

$$\min_{j, s} \left[\min_{\gamma_{R_1}} \sum_{x_i \in R_1(j, s)} (y - \gamma_{R_1})^2 + \min_{\gamma_{R_2}} \sum_{x_i \in R_2(j, s)} (y - \gamma_{R_2})^2 \right] \quad (11)$$

¿Cómo construimos un árbol?

- ▶ ¿Cuál es la solución?

¿Cómo construimos un árbol?



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

Agenda

1 Recap

2 Árboles

- Árboles de Regresión
- Árboles de Clasificación
- Sobreajuste

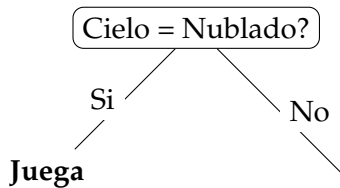
Árboles: Problema

- Jugamos al tenis?

Cielo	Humedad	Tenis?
Sol	Alta	No
Sol	Alta	No
Nublado	Alta	Sí
Sol	Alta	No
Sol	Normal	Sí
Nublado	Alta	Sí
Nublado	Normal	Sí

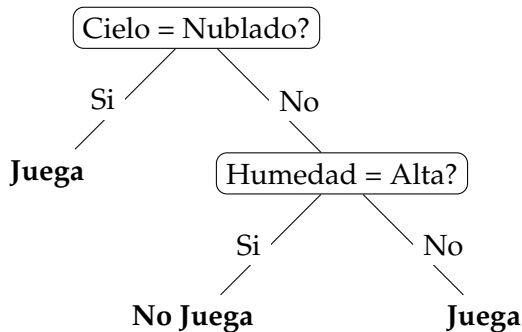
Árboles: Problema

Cielo	Humedad	Tenis?
Sol	Alta	No
Sol	Alta	No
Nublado	Alta	Sí
Sol	Alta	No
Sol	Normal	Sí
Nublado	Alta	Sí
Nublado	Normal	Sí



Árboles: Problema

Cielo	Humedad	Tenis?
Sol	Alta	No
Sol	Alta	No
Nublado	Alta	Sí
Sol	Alta	No
Sol	Normal	Sí
Nublado	Alta	Sí
Nublado	Normal	Sí



¿Cómo construimos un árbol de decisión?

- ▶ Regiones lo más “puras” posibles
 - ▶ **Regresión:** mínima varianza
 - ▶ **Clasificación:** ?

¿Cómo construimos un árbol de decisión?

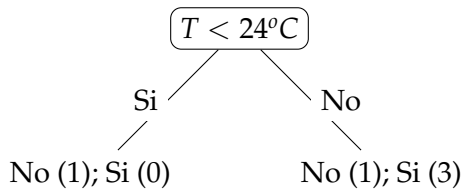
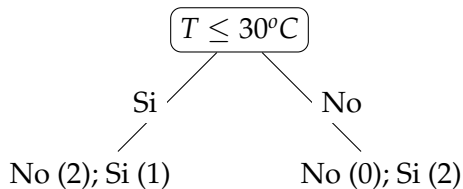
Problemas de clasificación

Temperatura °C	Llovió
23	NO
24	NO
29	SI
31	SI
33	SI

¿Cómo construimos un árbol de decisión?

Problemas de clasificación

- ¿Cuál de los dos cortes es mejor?



¿Cómo construimos un árbol de decisión?

Problemas de clasificación. Medidas de Impureza

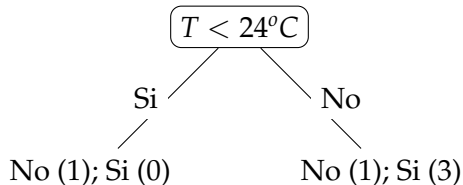
- ▶ Medidas de impureza dentro de cada hoja:
 - ▶ Índice de Gini : $G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$
 - ▶ Entropía : $-\sum_{k=1}^K \hat{p}_{mk} \log(\hat{p}_{mk})$
- ▶ Se define la impureza de un árbol por el promedio ponderado de las impurezas de cada hoja. El ponderador es la fracción de observaciones en cada hoja.

¿Cómo construimos un árbol de decisión?

Problemas de clasificación. Impureza

- ¿Cuál de los dos cortes es mejor?

Temperatura °C	Llovió
31	SI
24	NO
29	SI
33	SI
23	NO

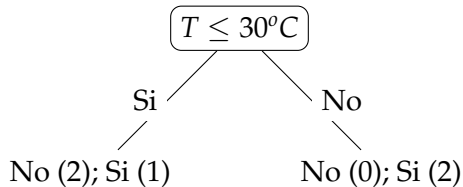


¿Cómo construimos un árbol de decisión?

Problemas de clasificación. Impureza

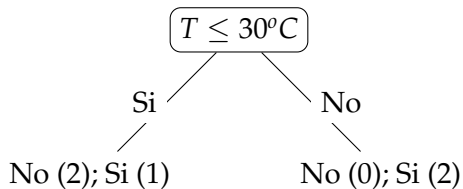
- ¿Cuál de los dos cortes es mejor?

Temperatura °C	Llovió
31	SI
24	NO
29	SI
33	SI
23	NO



¿Cómo construimos un árbol de decisión?

Problemas de clasificación. Predicción



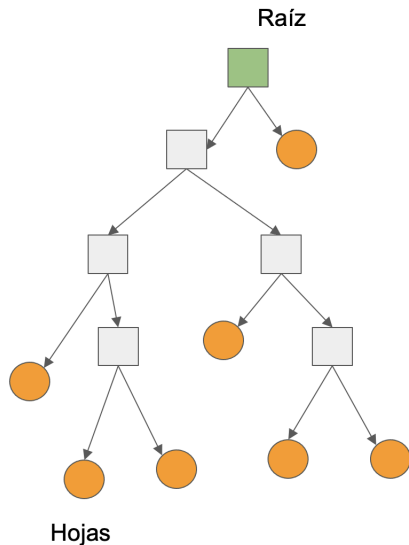
Agenda

1 Recap

2 Árboles

- Árboles de Regresión
- Árboles de Clasificación
- **Sobreaajuste**

Sobreajuste



Sobreajuste. Algunas soluciones

- ▶ Fijar la profundidad del árbol.
- ▶ Fijar la mínima cantidad de datos que están contenidos dentro de cada hoja.
- ▶ Pruning (poda).
 - ▶ Dejar crecer un árbol muy grande T_0
 - ▶ Luego cortarlo obteniendo sub-árbol (*subtree*)
 - ▶ Como cortarlo?

Pruning (poda)

- ▶ No es posible calcular el error de predicción usando cross-validation para cada sub-árbol posible
- ▶ Solución: *Cost complexity pruning* (cortar las ramas mas débiles)
 - ▶ Indexamos los arboles con T .
 - ▶ Un sub-árbol $T \in T_0$ es un árbol que se obtuvo colapsando los nodos terminales de otro árbol (cortando ramas).
 - ▶ $[T]$ = número de nodos terminales del árbol T

Pruning (poda)

- Cost complexity del árbol T

$$C_\alpha(T) = \sum_{m=1}^{[T]} n_m Q_m(T) + \alpha [T] \quad (12)$$

- donde $Q_m(T) = \frac{1}{n_m} \sum_{x_i \in R_m} (y_i - \hat{y}_m)^2$ para los árboles de regresión
- $Q_m(T)$ penaliza la heterogeneidad dentro de la regresión y α el número de regiones
- Objetivo: para un dado α , encontrar el pruning óptimo que minimice $C_\alpha(T)$

Pruning (poda)

- Mecanismo de búsqueda para T_α (pruning óptimo dado α).

Resultado: para cada α hay un sub-árbol único T_α que minimiza $C_\alpha(T)$.

- Eliminar sucesivamente las ramas que producen un aumento mínimo en $\sum_{m=1}^{[T]} n_m Q_m(T)$
- Se colapsa hasta el nodo inicial pero va a través de una sucesión de árboles
- T_α pertenece a esta secuencia. (Breiman et al., 1984)

Pruning (poda)

Algoritmo Completo

- 1 Utilizamos particiones recursivas binarias para hacer crecer el árbol
- 2 Para un dado α , aplicamos *cost complexity pruning* al árbol para obtener la secuencia de los subárboles como α .
- 3 Utilizamos K-fold cross-validation para elegir α .
- 4 Tenemos entonces una secuencia de subárboles para distintos valores de α
- 5 Elegimos el α y el subárbol que tienen el menor error de predicción.

Ejemplo



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