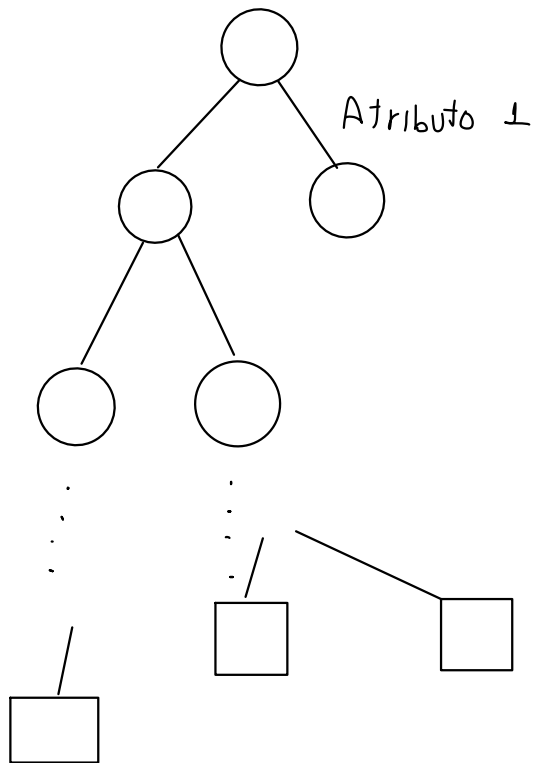
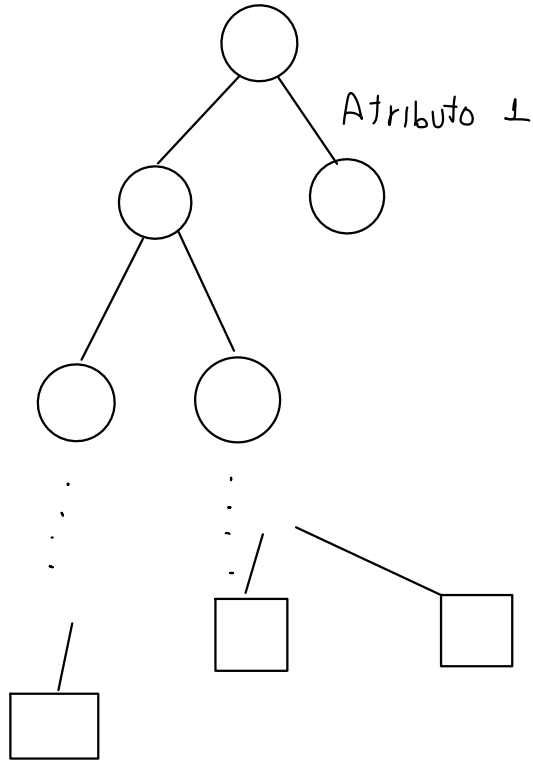




## Antes: Decision trees

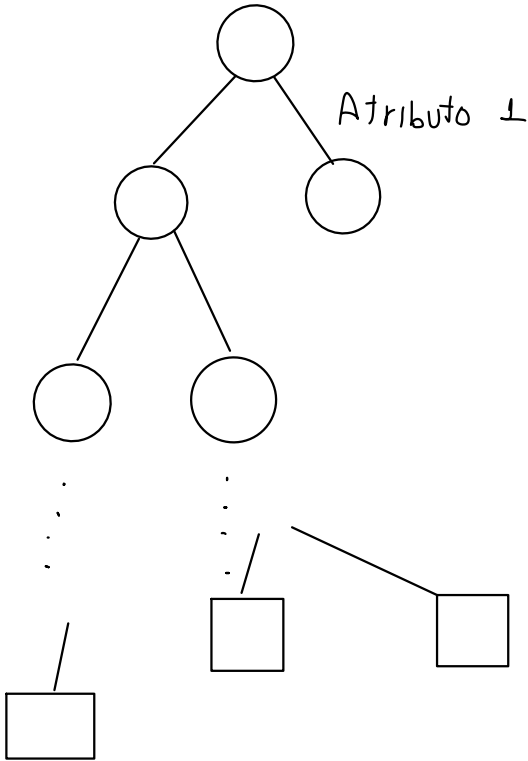


## Antes: Decision trees



Cómo elegimos el mejor atributo?

## Antes: Decision trees



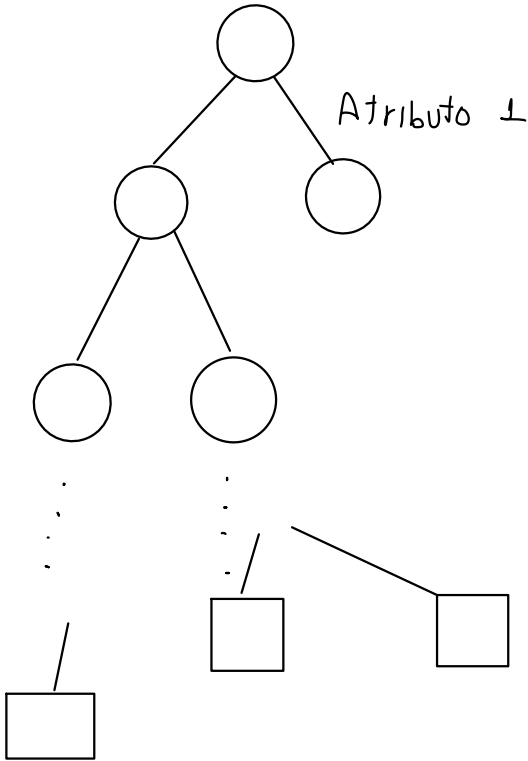
Cómo elegimos el mejor atributo?

Calculando la ganancia de información

$Gan(S, A)$  para cada atributo

Qué define la "complejidad" del árbol?

## Antes: Decision trees



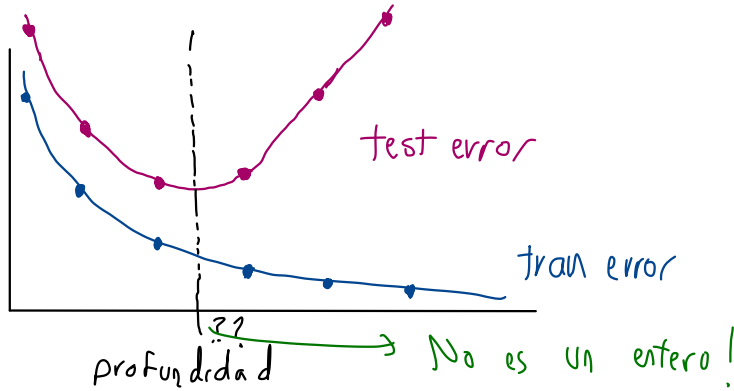
Cómo elegimos el mejor atributo?

Calculando la ganancia de información

$Gan(S, A)$  para cada atributo

Qué define la "complejidad" del árbol?

profundidad, número de hojas etc.



### Observaciones sobre los árboles de decisión

- El modelo óptimo no siempre se encuentra, pues los parámetros (profundidad, etc) son enteros
- Tienen problema de alta varianza: para cada muestra de entrenamiento el árbol puede cambiar mucho

# Bagging

(En el caso de árboles de decisión) Hacer crecer árboles grandes y reducir la varianza

Bagging se puede usar con cualquier tipo de algoritmo

Se utiliza para solucionar el problema de la **varianza**

Recordemos

$$\text{Err} = \overbrace{E_{x,D}[(h_D(x) - \bar{h}(x))^2]}^{\text{VARIANZA}} + \overbrace{E_x[(\bar{h}(x) - \bar{y}(x))^2]}^{\text{SES GO}^2} + E_{x,y}[(\bar{y}(x) - y)^2]$$

Diagram illustrating the decomposition of Error (Err) into three components:

- VARIANZA** (Variance):  $E_{x,D}[(h_D(x) - \bar{h}(x))^2]$ 
  - $h_D(x)$ : Modelo dado por A para D
  - $\bar{h}(x)$ : Modelo esperado de A
- SES GO<sup>2</sup>** (Bias squared):  $E_x[(\bar{h}(x) - \bar{y}(x))^2]$ 
  - $\bar{h}(x)$ : Promedio de predicciones del modelo
  - $\bar{y}(x)$ : Mediana de valores reales
- Bias**:  $E_{x,y}[(\bar{y}(x) - y)^2]$

## Bagging

(En el caso de árboles de decisión) Hacer crecer árboles grandes y reducir la varianza

Bagging se puede usar con cualquier tipo de algoritmo

Se utiliza para solucionar el problema de la **varianza**

Recordemos

$$\text{Err} = \underbrace{E_{x,D} \left[ \left( h_D(x) - \bar{h}(x) \right)^2 \right]}_{\text{VARIANZA}} + \underbrace{E_x \left[ \left( \bar{h}(x) - \bar{y}(x) \right)^2 \right]}_{\text{SESGO}^2} + E_{x,y} \left[ \left( \bar{y}(x) - y \right)^2 \right]$$

Diagram illustrating the decomposition of the error (Err) into three components:

- VARIANZA** (Variance):  $E_{x,D} \left[ \left( h_D(x) - \bar{h}(x) \right)^2 \right]$ 
  - $h_D(x)$ : Modelo dado por  $A$  para  $D$
  - $\bar{h}(x)$ : Modelo esperado de  $A$
- SESGO<sup>2</sup>** (Squared Bias):  $E_x \left[ \left( \bar{h}(x) - \bar{y}(x) \right)^2 \right]$ 
  - $\bar{h}(x)$ : Promedio de predicciones del modelo
  - $\bar{y}(x)$ : Media de valores reales
- Bias**:  $E_{x,y} \left[ \left( \bar{y}(x) - y \right)^2 \right]$

Queremos reducir este término

En otras palabras, acercar el modelo al modelo promedio. ¿Cómo?



Consideremos múltiples conjuntos de entrenamiento

$D_1, D_2, \dots, D_m$

Consideremos múltiples conjuntos de entrenamiento

$$D_1, D_2, \dots, D_m$$

Para cada uno de estos conjuntos entrenamos un modelo

$$h_1, h_2, \dots, h_m.$$

Luego, nuestro modelo será el promedio de los  $h_i$ .

Consideremos múltiples conjuntos de entrenamiento

$$D_1, D_2, \dots, D_m$$

Para cada uno de estos conjuntos entrenamos un modelo

$$h_1, h_2, \dots, h_m.$$

Luego, nuestro modelo será el promedio de los  $h_i$ .

$$h(x) = \frac{1}{m} \sum_{j=1}^m h_j(x)$$

$$\text{si } m \rightarrow \infty$$

$$h(x) \longrightarrow \bar{h}(x)$$

entre más grande  $m$ , nuestro modelo se acerca a  $\bar{h}(x)$ , así la varianza se reduce !

¿Es inconveniente?

De dónde sacamos los conjuntos de entrenamiento  $D_1, \dots, D_m$ ?

¿El inconveniente?

De dónde sacamos los conjuntos de entrenamiento  $D_1, \dots, D_m$ ?

Bootstrapping



¿Es inconveniente?

De dónde sacamos los conjuntos de entrenamiento  $D_1, \dots, D_m$ ?

## Bootstrapping

Sea  $D$  el conjunto de entrenamiento.  $|D| = N$

Tomamos de  $D$  una muestra de  $N$  puntos (con reemplazo!)



$$|D_i| = N$$

Los  $D_i$  son probablemente diferentes de  $D$

La probabilidad de sacar el mismo:  $\left(\frac{1}{N}\right)^N$

Luego sí entrenamos en los  $D_i$  y promediamos

# Random Forest

Hacer bagging con árboles de decisión. Con una modificación:

# Random Forest

Hacer bagging con árboles de decisión. Con una modificación:

Al entrenar cada árbol:

$D_j$         $\rightarrow$  Primer atributo?

Lo elegimos entre un subconjunto  
aleatorio de  $K$  atributos

$\rightarrow$  nuevo hiperparámetro

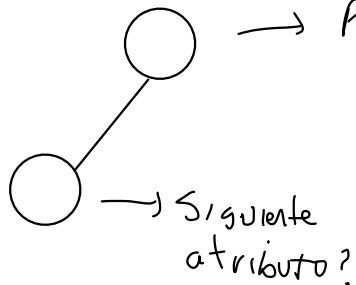


# Random Forest

Hacer bagging con árboles de decisión. Con una modificación:

Al entrenar cada árbol:

$D_j$  → Primer atributo?



Lo elegimos entre un subconjunto aleatorio de  $K$  atributos

→ nuevo hiperparámetro

así sucesivamente.

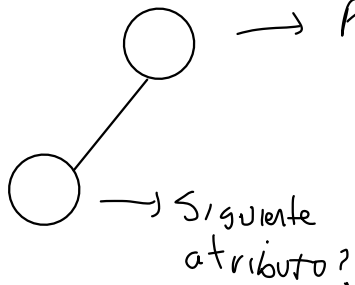
→ Lo elegimos entre un nuevo subconjunto de tamaño  $K$  de los que quedan

# Random Forest

Hacer bagging con árboles de decisión. Con una modificación:

Al entrenar cada árbol:

$D_j$  → Primer atributo?



Lo elegimos entre un subconjunto aleatorio de  $K$  atributos

→ nuevo hiperparámetro

→ Lo elegimos entre un nuevo subconjunto de tamaño  $K$  de los que quedan

así sucesivamente.

$K < d$  → dimension (# total de atributos)

□ Por qué  $k < d$ ? esta aleatoriedad hace que los árboles se diferencien (aún) más entre ellos

- ☐ Por qué  $k < d$ ? esta aleatoriedad hace que los árboles se diferencien (aún) más entre ellos
- ☐ Cómo elegir  $k$ ? se ha encontrado que  $k = \sqrt{d}$  funciona bien.

- ☐ Por qué  $k < d$ ? esta aleatoriedad hace que los árboles se diferencien (aún) más entre ellos
- ☐ Cómo elegir  $k$ ? se ha encontrado que  $k = \sqrt{d}$  funciona bien.
- ☐ Cuántos conjuntos  $D_i$  tomar? Tantos como se quiera!

aumentar la cantidad de modelos a promediar ( $h_i$ )

reduce la varianza sin aumentar el sesgo



- Por qué  $k < d$ ? esta aleatoriedad hace que los árboles se diferencien (aún) más entre ellos
- Cómo elegir  $k$ ? se ha encontrado que  $k = \sqrt{d}$  funciona bien.
- Cuántos conjuntos  $D_i$  tomar? Tantos como se quiera!

aumentar la cantidad de modelos a promediar ( $h_i$ )

reduce la varianza sin aumentar el sesgo

$$E_x \left[ \overbrace{(\bar{h}(x) - \bar{g}(x))}^{\text{sesgo}^2} \right]^2$$

→ No depende de los  $h_i$ !



## Ventajas de random forest


- Fácil de usar, no necesita preprocesamiento como normalización o estandarización de los datos, no necesita ajuste de nuevos hiperparámetros ( $k = \sqrt{d}$ ,  $m \rightarrow \infty$ )  
    ↪ default: 100

## Ventajas de random forest

- Fácil de usar, no necesita preprocesamiento como normalización o estandarización de los datos, no necesita ajuste de nuevos hiperparámetros ( $k = \sqrt{d}$ ,  $m \rightarrow \infty$ )  
    → default: 100
- Ha demostrado funcionar bastante bien



# Ventajas de random forest

- ☐ Fácil de usar, no necesita preprocesamiento como normalización o estandarización de los datos, no necesita ajuste de nuevos hiperparámetros ( $k = \sqrt{d}$ ,  $m \rightarrow \infty$ )  
→ default: 100
- ☐ Ha demostrado funcionar bastante bien
- ☐ Podemos medir el rendimiento sin usar conjunto de validación 

## Cómo hacer validación sin un conjunto de validación?

dado un punto  $(x_i, y_i)$ , éste no aparece en todos los conjuntos

$D_i$ . (se estima que en 30%-40% de los modelos no aparecerá)

así podemos validar midiendo el error en dichos modelos

error

$$\varepsilon = \frac{1}{N} \sum_{i=1}^N \frac{1}{z_i} \sum_{\substack{j \text{ t.q.} \\ (x_i, y_i \notin D_j)}} l(h_j(x_i), y_i)$$