


Clase 4

Árboles - Bagging - Random forest

Manuel Benjamín

November 4, 2023

Universidad de Buenos Aires 

Métodos basados en árboles

- Describimos métodos de árboles para regresión y clasificación.
- Vamos a segmentar el espacio de variables explicativas en un numero finito de regiones disjuntas.
- La forma de segmentar el espacio se puede representar en un árbol de decisión, por lo que estas metodologías se conocen como métodos de árboles de decisión.

Pros y contras

- Los metodos de arboles son simples y de fácil interpretación.
- En general no tiene el mejor poder predictivo.
- Para mejorar su capacidad de predicción se utilizan técnicas de **bagging**, **random forest** y **boosting**. Estas metodologías combinan muchos arboles para hacer una única predicción.
- Combinar muchos arboles aumenta mucho la predicción pero disminuye la interpretabilidad.

Arboles - regresión

```
> head(Hitters)
```

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	League
-Andy Allanson	293	66	1	30	29	14	1	293	66	1	30	29	14	A
-Alan Ashby	315	81	7	24	38	39	14	3449	835	69	321	414	375	N
-Alvin Davis	479	130	18	66	72	76	3	1624	457	63	224	266	263	A
-Andre Dawson	496	141	20	65	78	37	11	5628	1575	225	828	838	354	N
-Andres Galarraga	321	87	10	39	42	30	2	396	101	12	48	46	33	N
-Alfredo Griffin	594	169	4	74	51	35	11	4408	1133	19	501	336	194	A
	Division	PutOuts	Assists	Errors	Salary	NewLeague								
-Andy Allanson	E	446	33	20	NA	A								
-Alan Ashby	W	632	43	10	475.0	N								
-Alvin Davis	W	880	82	14	480.0	A								
-Andre Dawson	E	200	11	3	500.0	N								
-Andres Galarraga	E	805	40	4	91.5	N								
-Alfredo Griffin	W	282	421	25	750.0	A								

Figure 1: Data set Hitters

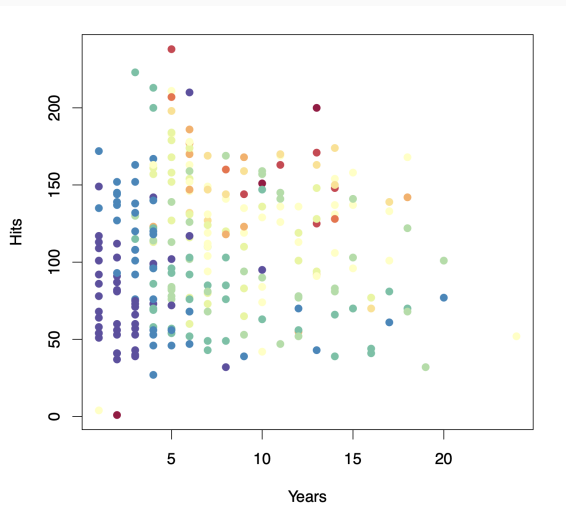


Figure 2: Salario de los jugadores, de bajo: azul y verde, a alto: amarillo y rojo



Figure 3: Arbol de decisión para el $\log(\text{Salary})$

- En cada nodo interno la etiqueta de la forma $(X_j < t_k)$ indica que la rama izquierda corresponde a esa partición y la rama derecha corresponde a $X_j \geq t_k$
- El árbol tiene dos nodos internos y tres nodos terminales (hojas). El número en cada hoja es el promedio de las observaciones que cayeron en la rama.
- El espacio de variables predictoras está dividido en tres regiones

$$R_1 = \{X|\text{Years} < 4.5\}$$

$$R_2 = \{X|\text{Years} \geq 4.5 \text{ y Hits} < 117.5\}$$

$$R_3 = \{X|\text{Years} \geq 4.5 \& \text{Hits} \geq 117.5\}$$

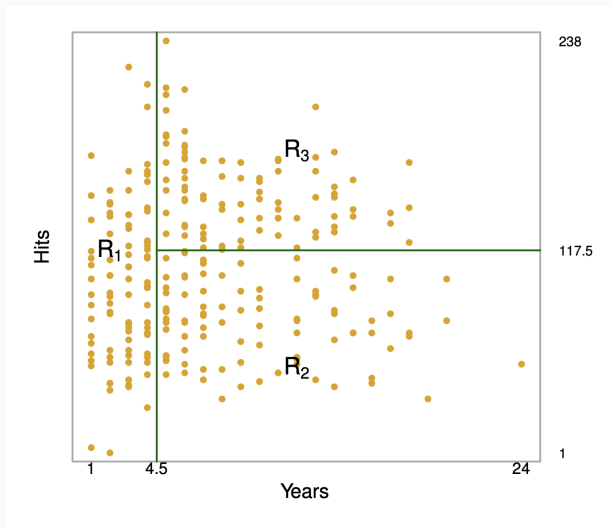


Figure 4: Partición del espacio de variables predictoras

- Las regiones R_1 , R_2 , R_3 se las conoce como nodos terminales.
- Los arboles de decisión se los suele dibujar con las hojas colgando.
- Los nodos del árbol donde se parte es espacio de covariables se conoce como nodos internos.

- **Years** es el factor mas importante para determinar el Salario. Jugadores con menos años ganan menos que jugadores con mas años.
- Para los jugadores menos experimentados la cantidad de **Hits** no influye en el salario.
- La cantidad de **Hits** si impactan en el salario de jugadores con 5 años o mas.
- Fácil de interpretar, de mostrar y de explicar.

Detalles sobre la construcción de árboles de regresión.

1. Dividimos el espacio de variables predictoras en rectángulos multidimensionales, R_1, \dots, R_J .
2. En cada rectángulo predecimos la respuesta con el promedio de la respuesta de las observaciones que caen en el mismo.
3. El objetivo es encontrar las cajas que minimizan el RSS dado por

$$RSS = \sum_{j=1}^J \sum_{i \in R_j} (y_i - \bar{y}_{R_j})^2$$

- No es computacionalmente razonable, resolver el problema de la partición óptima en J cajas.
- Por esta razón se utiliza un heurística golosa. De arriba para abajo haciendo partición binaria recursiva.
- Decimos de arriba para abajo porque empieza arriba del árbol (arriba en el gráfico no en la forma de un árbol de verdad) y va partiendo el espacio de manera sucesiva.
- Es goloso porque en cada paso elige el mejor split posible en vez de buscar por la partición que llevará a la solución óptima.

1. Iniciamos el árbol seleccionando la variable y su partición que minimiza el RSS.

$$R_1(j, s) = \{X|X_j < s\} \text{ y } R_2(j, s) = \{X|X_j \geq s\}$$

Buscamos la variable j y el punto de partición s que resuelven

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

2. Dado un árbol ya particionado: Buscamos en todos los nodos terminales la mejor partición y elegimos entre todos el que minimiza el RSS.
3. Seguimos particionando hasta alcanzar un criterio de parada, e.g Todos los nodos terminales con menos de 5 observaciones.

Predicciones

- Para una nueva observación predecimos la respuesta con el promedio de la región en la que cae.
- El siguiente gráfico muestra las predicciones para un árbol

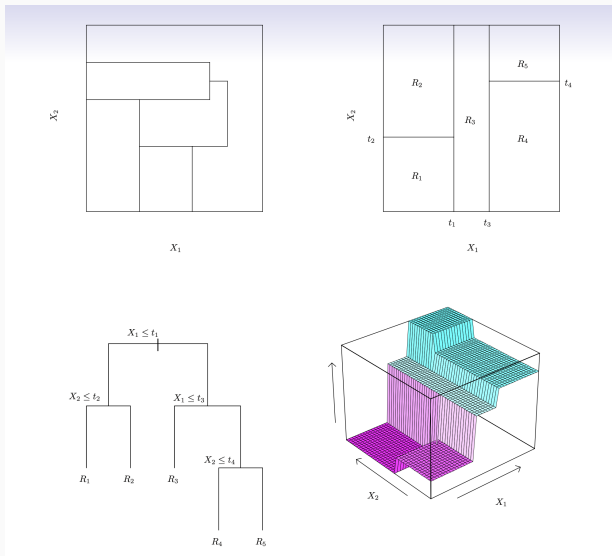


Figure 5: Arriba izquierda: Una región que no puede ser obtenida por el algoritmo descrito. Arriba derecha la región obtenida por el árbol de abajo a la izquierda. Abajo derecha: Predicciones

Jardinería de árboles

- El proceso descrito puede producir buenas predicciones en el set de entrenamiento... pero tiende a sobre-ajustar la data. ¿Por qué?

Jardinería de árboles

- El proceso descrito puede producir buenas predicciones en el set de entrenamiento... pero tiende a sobre-ajustar la data. ¿Por qué?
- Un árbol mas chicos con menos regiones, puede tener menos varianza y mejor interpretabilidad a costas de un poco de sesgo.
- Podríamos solo hacer particiones en caso de que un split decrezca lo suficiente el RSS.
- Si bien esta estrategia resulta en arboles mas chicos tiene poca vista global. (Pensar un ejemplo)

Podado de arboles

Jardinería de árboles

- Una mejor estrategia es crear un árbol muy grande T_0 y después podarlo hasta obtener un sub árbol apropiado.
- Podado por costo de complejidad o podado por el eslabón mas débil.
- Consideramos una secuencia de arboles indexadas por un hiper-parametro α . Para cada valor de α existe un sub arbol $T \subseteq T_0$ que minimiza

$$\sum_{i=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \bar{y}_{R_m})^2 + \alpha |T|.$$

Aca $|T|$ indica el numero de nodos terminales (regiones) del árbol T .

Podado de árbol - Mejor sub árbol

El parámetro α controla un balance entre la complejidad del subárbol y el ajuste de los datos de entrenamiento.

Elegimos $\hat{\alpha}$ por validación cruzada.

La metodología incluye la selección de α y el ajuste del sub árbol correspondiente.

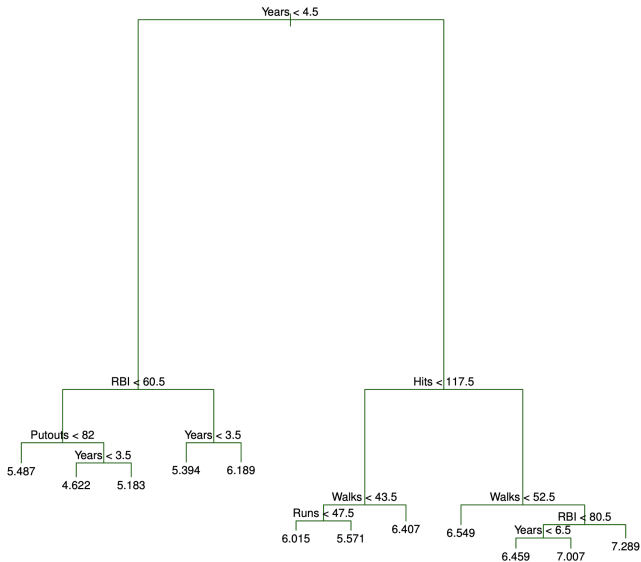
Resumen: Algoritmo para ajuste de un árbol

1. Con la metodología descrita cultivamos un árbol grande con la data de entrenamiento, parando cuando cada nodo terminal tiene menos de un numero de observaciones **max node**.
2. Podamos el árbol de manera de obtener una secuencia de sub árboles como función de α .
3. Estimamos el error de validación cruzada para cada α en una grilla.
4. Devolver el árbol del paso 2 que corresponda al valor elegido de α .

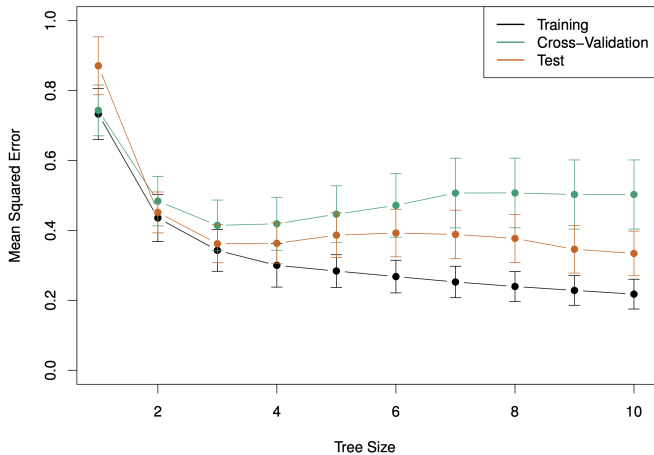
Ejemplo - Baseball

- Partimos el dataset a la mitad para entrenamiento y testeo.
- Construimos T_0 con la data de entrenamiento y hallamos la secuencia de sub arboles para cada α
- Mediante validación cruzada con $K = 6$ cruces elegimos el valor de α

Ejemplo - Baseball - T_0



Ejemplo - Baseball



Arboles - Clasificación

Árboles para clasificación

- Muy similar a arboles de regresión. Salvo que ahora no miramos RSS si no accuracy.
- En cada nodo terminal predecimos con la clase con mayor cantidad de ocurrencias.

Detalles

- Usamos partición binaria para cultivar el árbol.
- Una alternativa al RSS miramos la proporción de clasificaciones incorrectas. Es decir la fracción de observaciones que no pertenecen a la clase mayoritaria

$$E_m = 1 + \max_k \hat{p}_{mk}.$$

Donde \hat{p}_{mk} es la proporción de observaciones de la clase k en la región m .

- En la práctica se utilizan otras medidas ya que el error de clasificación no es lo suficientemente sensible (ni derivable).

Gini index y Entropía

- Índice Gini de la región m

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

Una medida de la varianza total a través de las K clases. El índice toma valores cerca de cero solo todos los \hat{p}_{mk} están cerca de cero o uno.

- El índice es una medida de pureza del nodo. Un valor bajo indica que la mayoría de las observaciones son de una única clase
- **Cross-entropy**

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

- Numéricamente ambas medidas suelen ser similares.

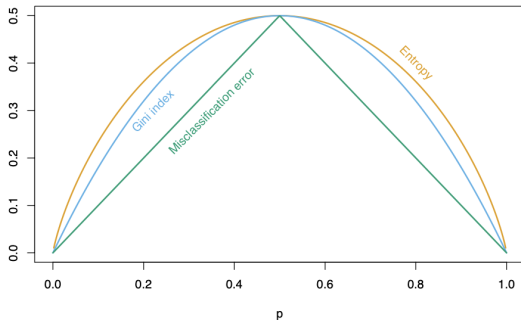


Figure 6: Medidas de impureza de nodos para un problema de clasificación de dos clases como función de la proporción p de la clase 2. La entropía cruzada está escalada para que pase por el $(0.5, 0.5)$.

Ejercicio

Mostrar que el Índice de Gini es la probabilidad de que al seleccionar dos elementos con reposición sean de clases distintas.

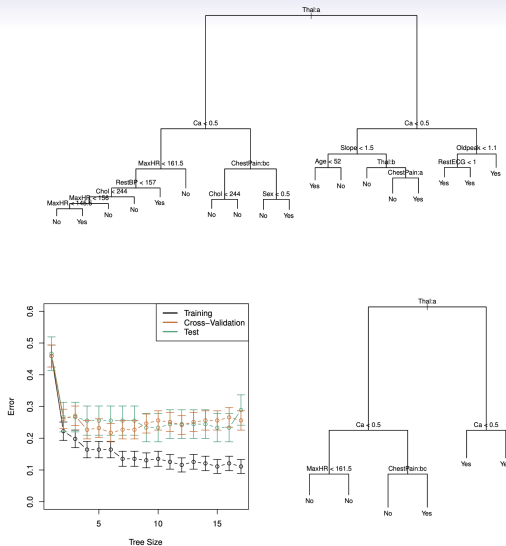


Figure 7: Presencia de en enfermedad del criazón con 13 variables predictoras que incluye, edad, sexo biológico y colesterol

Modelo lineal versus arboles de decisión

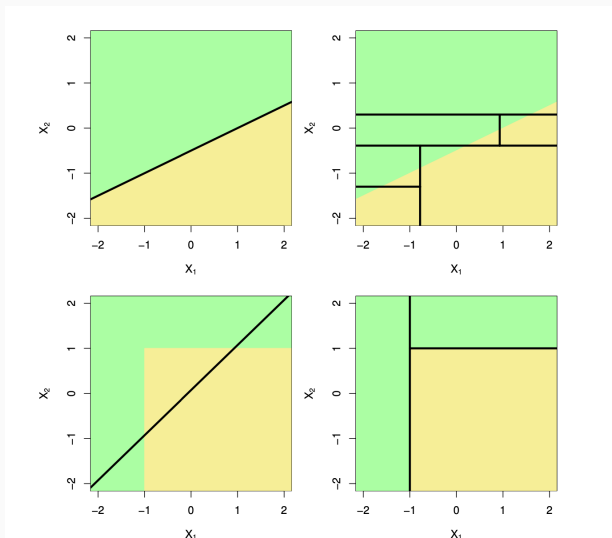


Figure 8: Columna izquierda: Ajustes lineales. Columna derecha ajuste con arboles. Fila superior: Modelo lineal. Fila inferior: Modelo no lineal.

Agregacion de modelos

Bagging

- **Agregación por bootstrap o bagging**, es una metodología utilizada para reducir la varianza de una metodología de aprendizaje estadístico. La introducimos con arboles pero puede usarse en otros contextos.
- Si uno tiene n observaciones independientes Z_1, \dots, Z_n con varianza σ^2 , la varianza del promedio es σ^2/n .
- Promediar un conjunto de observaciones reduce la varianza. (En la practica no tenemos múltiples sets de entrenamiento)

Bagging

- Podemos bootstrapear, tomando muestras repetidas del set de entrenamiento.
- Generamos B training sets bootstrapados. Entrenamos la metodología con el set b de entrenamiento bootstrapado y obtenemos $\hat{f}^{*b}(x)$. Promediamos las predicciones

$$\hat{f}_{bag} = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x).$$

- En caso de clasificación, para cada observación a predecir tomamos el voto mayoritario de los B modelos bootstrapado. Es decir, la predicción mas frecuente.

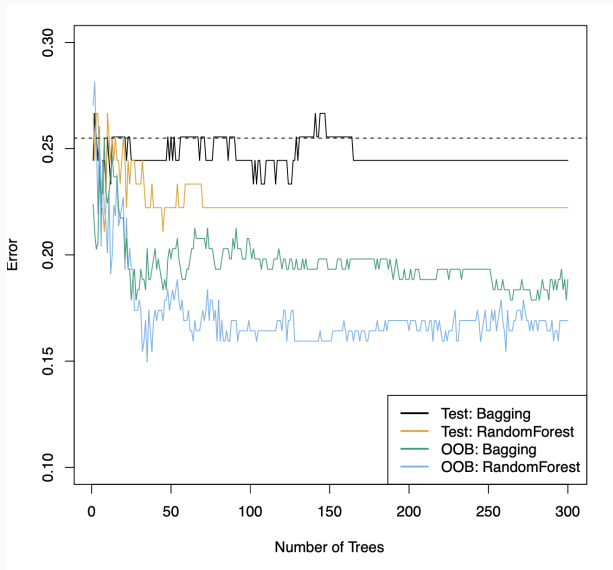
Estimación del error - Out of Bag

- Hay una forma muy sencilla de estimar el error de testeo para un modelo 'metido en bolsa'
- Cuando se toma una muestra con reposición, aproximadamente un tercio de las observaciones quedarán afuera.
- Este tercio de observaciones que no fueron usadas para ajustar el modelo baggeado se llama 'out -of bag' (OOB).
- Podemos predecir en las observaciones fuera de la bolsa y promediar los errores de predicción al cuadrado (o los errores de clasificación).
- Promediando estos promedio obtenemos **OOB error estimation**.

Random Forests

- Random forest mejora bagging para arboles con una pequeña modificación que busca volver menos dependientes a los arboles reduciendo la varianza del promedio.
- Como en bagging cultivamos arboles de decisión bootstrapeando el set de entrenamiento.
- Al crear los arboles de decisión, cada vez que vamos a considerar un split, elegimos al azar m variables explicativas del total de p . La partición solo se permite usando alguna de las m predictoras.
- Se vuelven a sortear los m predictoras para cada split y generalmente se elige $m \approx \sqrt{p}$.

Bagging y Random forest para Heart Data



Ejercicios

- Leer y seguir sección 8.3.3 Del ISL para aprender a utilizar Bagging y Random forest.
- Realizar los ejercicios pertinentes de las paginas 363 y 364 del ISLR.