UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Una Introducción al Aprendizaje de Máquina basado en Árboles de Clasificación

MONOGRAFÍA DE TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE MATEMÁTICO
PROYECTO CURRICULAR DE MATEMÁTICAS

Oscar Guzmán y Santiago Vargas
Dirigido por: Luis Alejandro Masmela Caita

Bogotá DC
Junio de 2022

Resumen

En el área de Machine Learning, el modelo de Árboles de Clasificación es uno de los más populares en el ámbito de Aprendizaje Supervisado. Por lo tanto, basados el trabajo de (Breiman y col., 1984), se plantea describir de manera matemática la construcción de un modelo de árbol de clasificación usando la base de datos IRIS (Fisher, 1936). Además, se tratarán temas como el rendimiento del modelo, el podado y la validación cruzada con el fin de obtener un modelo de árbol eficiente.

Palabras clave: Árboles de Clasificación, Machine Learning, Aprendizaje Supervisado.

Clasificación AMS: 68T01

Agradecimientos: Un agradecimiento especial a nuestras familias; al docente Luis Alejandro Masmela por su apoyo y paciencia; también a la Universidad Distrital quien nos acogió en este proceso.

Introducción

El aprendizaje supervisado es una de las principales ramas del Machine Learning. Este consiste en el diseño de algoritmos matemáticos que en un conjunto de datos dado relacionen un grupo de variables independientes con una variable dependiente. Uno de los métodos más conocidos son los Árboles de Clasificación, pues su funcionamiento es simple y además se han usado para modelar diferentes proyectos (clasificación de letras en una imagen, predecir la enfermedad de un paciente basado en sus síntomas, etc.).

En este trabajo se estudiarán los árboles de clasificación planteados por Leo Breiman (Breiman y col., 1984) acompañado con las lecturas de (Choi, 2017). Cabe resaltar que según (Loh, 2014), dichos árboles de clasificación están basados en dos artículos que son los cimientos de los árboles de regresión en (Morgan y Sonquist, 1963), y clasificación en (Messenger y Mandell, 1972) respectivamente.

Ahora bien, el principio fundamental de los árboles de clasificación de Breiman es crear particiones en un conjunto de datos a partir de "preguntas" que tienen una respuesta cerrada, si o no. Por ejemplo, el siguiente árbol clasifica en una muestra de agua su potabilidad usando preguntas acerca de condiciones químicas en el líquido:

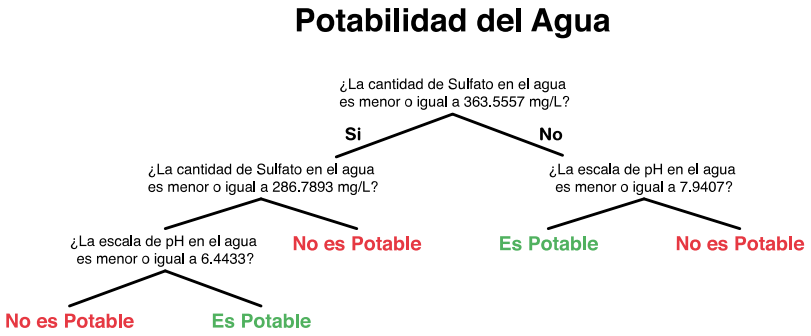


Figura 1: Árbol de Decisión creado a partir de los datos extraidos en: https://www.kaggle.com/datasets/adityakadiwal/water-potability

Aunque pueda parecer sencilla la construcción de los árboles, surgen interrogantes acerca de: Las preguntas formuladas, la cantidad de preguntas y el rendimiento del árbol. La solución a dichas cuestiones son los objetivos principales de este trabajo, que además se abordarán de manera matemática.

Por lo tanto, esta monografía se llevará a cabo en el siguiente orden: En el capítulo 1 se introducirá la base datos IRIS, y se construirá un primer árbol de clasificación basado en una porción de la base de datos; en el capítulo 2 se planteará el "podado", el cual crea un conjunto de árboles cada uno basado en el árbol del capítulo anterior; en el Capítulo 3 seleccionará el mejor árbol de la anterior sección usando la validación cruzada; y por último, en el Capítulo 4 se hallará el mejor árbol para la base de datos IRIS completa.

1. Árbol Binario de Clasificadores Estructurados

1.1. Base de Datos

El conjunto de datos IRIS (Fisher, 1936) consta de 150 observaciones del ancho y alto, del pétalo y el sépalo de tres tipos de plantas: Iris Setosa, Iris Virginica e Iris Versicolor.

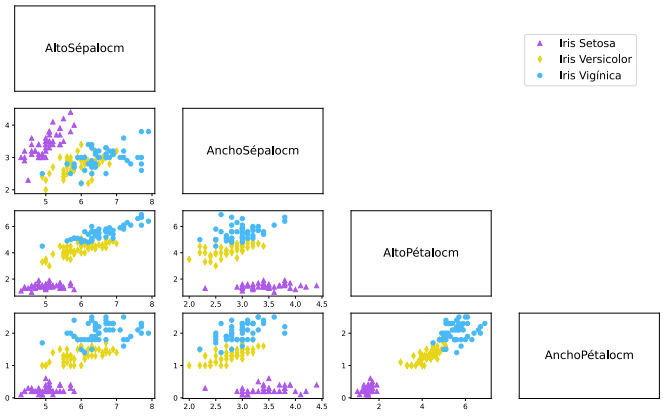


Figura 2: Interacción entre las diferentes variables que conforman la base de datos Iris

Aunque, uno de los objetivos de este trabajo es crear un árbol de clasificación para la base de datos IRIS, se usará una parte de dicha base de datos conformada por el alto y el ancho del pétalo, con el fin de ilustrar algunas definiciones y ejemplos.

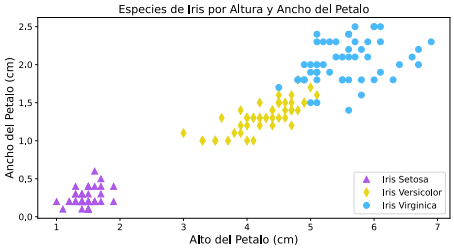


Figura 3: Gráfica de IRIS usando las variables de ancho y alto del pétalo

De manera general, se puede definir la muestra de aprendizaje (base de datos con la que se construye el modelo) como sigue:

Definición 1. *Una muestra de aprendizaje es un conjunto de la forma:*

$$\mathcal{L} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$$

Donde x_i pertenece al conjunto de variables independientes X, y y_i a su asociación en el conjunto de la variable dependiente $C = \{1, 2, \ldots, J\}$.

En nuestro caso, cada vector de $\mathcal{L}$ tiene como primera componente una dupla con las variables alto y ancho del pétalo en el conjunto X de variables independientes; y como segunda componente, a una asociación numérica de las clases de flor Iris (Setosa, virginica y versicolor) en el conjunto C.

Ahora bien, con la muestra de aprendizaje, se procederá a mostrar como formular preguntas que creen buenas divisiones en $\mathcal{L}$ para construir un árbol de clasificación.

1.2. La reducción de impureza y las divisiones más óptimas

Para comenzar, en la Figura 3, se puede observar que algunas clases tienen divisiones bastante claras. Por ejemplo, la clase Iris Setosa puede ser agrupada en el conjunto $t_1 \subset \mathcal{L}$ que toma los datos con el alto de pétalo menor o igual a $2.5cm$. Por otro lado, el conjunto restante t_2 tiene dos clases involucradas; sin embargo, puede realizarse una partición conformada por dos nodos: t_3, t_4.

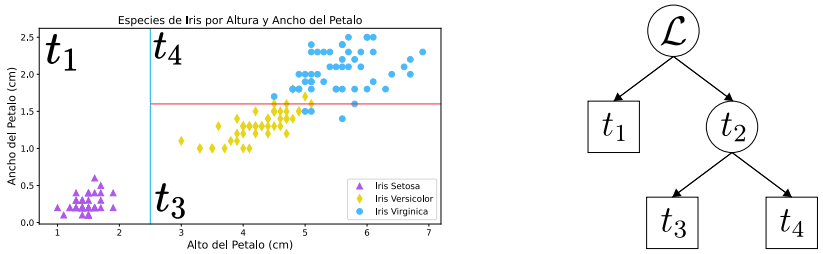


Figura 4: A izquierda el gráfico de los datos con las divisiones realizadas. A derecha, la visualización de las divisiones como un grafo de árbol donde el nodo raíz corresponde a $\mathcal{L}$; las hojas (o nodos terminales), corresponden a los nodos t_1, t_3, t_4; y las ramas, en este caso corresponde al nodo t_2.

En la anterior figura se observa que los datos han quedado lo suficientemente bien divididos por los subconjuntos t_1, t_3, t_4, ya que la desigualdad de las clases es casi total o absolutamente total en cada hoja del árbol. La desigualdad de las clases en los nodos es fundamental al momento de escoger la mejor decisión, esta es medida en base a la proporción de las clases que hay en un nodo.

Dada la importancia de las proporciones de las clases en los nodos, se mostrarán algunas notaciones para estas; pues, serán necesarias para cálculos posteriores:

Sin perdida de generalidad, supongamos que la base de datos $\mathcal{L}$ tiene tamaño N, entonces:

Definición 2.

$N_j = \lvert\{(x_i, y_i) \in \mathcal{L} \mid y_i = j\}\rvert$	*N_j es la cantidad de datos en $\mathcal{L}$ que tienen a la clase $j \in C$*
$\mathcal{L}(t) = \{(x_i, y_i) \in \mathcal{L} \mid (x_i, y_i) \in t\}$	*$\mathcal{L}(t)$ son los datos en $\mathcal{L}$ que pertenecen al nodo t*
$N(t) = \lvert\mathcal{L}(t)\rvert$	*$N(t)$ es la cantidad de datos en $\mathcal{L}$ que hay en el nodo t*
$N_j(t) = \lvert\{(x_i, y_i) \in t \mid y_i = c_j\}\rvert$	*$N_j(t)$ es la cantidad de datos en $\mathcal{L}$ que tienen la clase $j \in C$ en el nodo t*

La proporción de datos en $\mathcal{L}$ que tienen la clase j es:

$$\pi(j) = \frac{N_j}{N}$$

La proporción de datos en $\mathcal{L}$ que tengan la clase j y están en un nodo t es:

$$p(j,t) = \frac{N_j(t)}{N}$$

La proporción de datos que están en un nodo t es:

$$p(t) = \sum_{j=1}^{k} p(j,t) = \frac{N(t)}{N}$$

La proporción de datos en un nodo t que tengan la clase j es:

$$p(j \mid t) = \frac{N_j(t)}{N(t)}$$

Con las anteriores proporciones definidas, se mostrará la función de impureza la cual dará un puntaje al nivel de desigualdad que hay en un nodo.

Definición 3. *Se dice que una función ϕ definida en el conjunto de vectores $(p_1, p_2, \ldots, p_J)$ de tamaño J, tal que $p_j \geq 0$; $j = 0, 1, \ldots, J$ y $\sum_{j=1}^{J} p_j = 1$, es una **función de impureza** si:*

(i) *la función ϕ toma el valor máximo únicamente en el punto $(\frac{1}{J}, \frac{1}{J}, \ldots, \frac{1}{J})$*

(ii) *Para dos números enteros i, j en $\{1, 2, \ldots, J\}$ se cumple que: $\phi(\ldots, p_i, \ldots, p_j, \ldots) = \phi(\ldots, p_j, \ldots, p_i, \ldots)$*

(iii) La función ϕ toma el valor mínimo únicamente en los puntos $(1,0,\ldots,0),(0,1,\ldots,0),\ldots,(0,0,\ldots,1)$

De la anterior función se puede observar que a los nodos totalmente desiguales como t_1 en la Figura 4 se le otorga el mínimo valor, mientras que a los nodos con igualdad en las proporciones de sus clases como $\mathcal{L}$ o t_2 tendrán el mayor valor de impureza. Algunos ejemplos de función de impureza serán mostrados a continuación:

Teorema 1. *La función:*

$$G(p_1,p_2,\ldots,p_J) = 1 - \sum_{i=1}^{J}(p_i)^2$$

Es una función de impureza (conocida como ***Índice de Impureza de Gini)***

Demostración:
Sea $(p_1,p_2,\ldots,p_J)$ un vector que cumple con las condiciones dadas en la definición 7, entonces:

(i) Para hallar el máximo de la función, se utilizara el método de multiplicadores de Lagrange. Donde la función f a optimizar y la función g que restringe a f están dadas por:

$$f(p_1,p_2,\ldots,p_J) = 1 - \sum_{i=1}^{J} p_i^2 \qquad g(p_1,p_2,\ldots,p_J) = p_1 + p_2 + \ldots p_J$$

Desde luego se puede aplicar Lagrange en $(0,1]^n$ pues en esa región las funciones f,g son continuas y su primera derivada es continua. Entonces, se hallará el punto $(p_1,p_2,\ldots,p_J)$ que satisfaga:

$$\begin{cases} \nabla(f(p_1,p_2,\ldots,p_J)) = \lambda \cdot \nabla(g(p_1,p_2,\ldots,p_J)) & \lambda \in \mathbb{R} \\ g(p_1,p_2,\ldots,p_J) = 1 \end{cases}$$

Reemplazando:

$$\begin{cases} \left(-2\cdot p_1,-2\cdot p_2,\ldots,-2\cdot p_J\right) = \lambda\cdot(1,1,\ldots,1) \\ p_1+p_2+\cdots+p_J = 1 \end{cases}$$

Por lo tanto:

$$-2\cdot p_j = \lambda \implies p_j = -\frac{\lambda}{2}; \quad \forall j \in \{1,2,\ldots,J\}$$

Así $p_1 = p_2 = \cdots = p_J = -\frac{\lambda}{2}$, entonces:

$$p_1 + p_2 + \cdots + p_J = 1 \implies p_1 = \frac{1}{J}$$

Dando como conclusión que $(\frac{1}{J}, \frac{1}{J}, \cdots, \frac{1}{J})$ es un punto máximo o mínimo. Sin embargo, dicho punto es máximo pues se cumple que $G(\frac{1}{J}, \frac{1}{J}, \cdots, \frac{1}{J}) > 0$ si $J > 1$ y además en la condición iii se mostrará un punto que en su evaluación en G no sobrepasa al encontrado anteriormente.

(ii) Sean i, j en $\{1, 2, \ldots, J\}$, entonces:

$$G(\ldots, p_i, \ldots, p_j, \ldots) = 1 - (\cdots + p_j + \cdots + p_i + \ldots) = G(\ldots, p_j, \ldots, p_i, \ldots)$$

(iii) La función G es no negativa, pues para cualquier componente p_i del vector de probabilidades $(p_1, p_2, \ldots, p_J)$ se cumple que:

$$0 \leq 1 - \sum_{j=1}^{J} (p_j)^2 \implies 0 \leq G(p_1, \ldots, p_i, \ldots, p_J)$$

Por lo tanto, el mínimo valor posible que puede tomar G es 0. Por otra parte, se tiene que:

$$G(1, 0, \ldots, 0) = 1 - \left((1)^2 + 0^2 + \cdots + (0)^2\right) = 0$$

Entonces, la condición 2 implica que:

$$G(1, 0, \ldots, 0) = G(0, 1, \ldots, 0) = \cdots = G(0, 0, \ldots, 1) = 0$$

Por lo tanto, los vectores $(1, 0, \ldots, 0), (0, 1, \ldots, 0), \cdots, (0, 0, \ldots, 1)$ son mínimos en G. ■

De manera similar a la anterior prueba, se puede mostrar que el indice de entropía de Shannon tambien es una función de impureza:

Teorema 2. *La función H de Entropía de Shannon dada por:*

$$H(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} g(p_i)$$

Donde $g(p_i)$ esta definida como:

$$g(p_i) = \begin{cases} 0 & ; p_i = 0 \\ p_i \cdot \log_2(p_i) & ;\textit{en otro caso} \end{cases}$$

Es una función de impureza.

En particular, si se tiene el vector de probabilidad $(p, 1-p)$ con $0 \leq p \leq 1$, los valores de las anteriores funciones en ese vector se pueden visualizar como sigue:

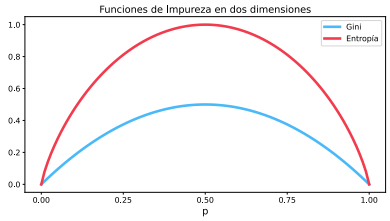


Figura 5: Comparación de funciones de impureza en el vectores de probabilidad $(p, 1-p)$. En la gráfica se observa que ambas funciones tienen el máximo valor cuando $p = \frac{1}{2}$

Ahora bien, la función de impureza aplicada a las proporciones de las clases en los nodos de un árbol, puede se define como:

Definición 4. *Dada una función de impureza ϕ, se define la medida de impureza $i(t)$ de un nodo t por:*

$$i(t) = \phi(p(1 \mid t), p(2 \mid t), \ldots, p(J \mid t)).$$

Por ejemplo. la impureza de los nodos t_1 y $t_2 = t_3 \cup t_4$ de la Figura 4 usando el índice de Gini, es:

$$i(t_1) = 1 - \left(\frac{50}{50}\right)^2 = 0 \qquad i(t_2) = 1 - \left(\frac{50}{100}\right)^2 - \left(\frac{50}{100}\right)^2 = 0.5$$

Como se tenia previsto, los cálculos anteriores muestran que el nodo t_1 es más desigual que el nodo t_2. Usualmente, a los nodos con impureza baja se les dice **puros**; en cambio, los nodos con mayor impureza son **impuros**.

Ahora bien, ya que se tiene una forma de observar numéricamente la desigualdad que hay en un nodo, se introducirá la función de disminución de la impureza la cual dará un valor a la utilidad de una pregunta formulada para dividir un nodo.

Definición 5. *Sea t un nodo y s una condición (pregunta) que provoca una división de t en dos nodos t_L,t_R. Se define la disminución de la impureza como:*

$$\Delta i(s, t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R)$$

Donde $p_L = \frac{p(t_L)}{p(t)}$, $p_R = \frac{p(t_R)}{p(t)}$.

Con ayuda de la anterior función se puede comparar el rendimiento de dos o mas preguntas en el mismo nodo t, solo basta con evaluarlas y escoger la que tenga mayor valor bajo la función.

Para mostrar un ejemplo de la anterior definición, se hallará la disminución de la impureza en las siguientes divisiones del mismo nodo $\mathcal{L}$.

Figura 6: Dos divisiones para el nodo $\mathcal{L}$

Usando como función de impureza el Índice de Gini, se tiene:

- Para la condición s_1 en la variable de alto del pétalo que dio como resultado t_1, t_2:

$$\Delta i(s_1, \mathcal{L}) = i(\mathcal{L}) - \left(\frac{50}{150}\right) \cdot i(t_1) - \left(\frac{100}{150}\right) \cdot i(t_2) = 0.6667 - \left(\frac{50}{150}\right) \cdot (0) - \left(\frac{100}{150}\right) \cdot (0.5) = 0.3334$$

- Para la condición s_2 en la variable ancho del pétalo que dio como resultado t_3, t_4:

$$\Delta i(s_2, \mathcal{L}) = i(\mathcal{L}) - \left(\frac{104}{150}\right) \cdot i(t_3) - \left(\frac{46}{150}\right) \cdot i(t_4) = 0.6667 - \left(\frac{104}{150}\right) \cdot (0.545) - \left(\frac{46}{150}\right) \cdot (0.043) = 0.2754$$

Como se puede observar la disminución de la impureza es mas alta con la división que da como resultado t_1, t_2, que con la división que da como resultado t_3, t_4. Pues, una división logra separar toda una clase, mientras que la otra no lo hace muy bien.

Ahora bien, se explicará el origen matemático de la anterior definición. Primeramente, se puede observar que las hojas juegan un papel importante en los árboles, pues esto agrupan los resultados de las divisiones.

Se notará el conjunto de hojas como:

Definición 6. *Sea T un árbol. Se denotará por $\widetilde{T}$ el conjunto de nodos terminales (hojas) de T*

A su vez, para construir la función de disminución de impureza, es necesario observar la impureza de las hojas en el árbol. Para ello se define un promedio de las impurezas en los nodos terminales del árbol:

Definición 7. *La impureza de un árbol esta dada por:*

$$I(T) = \sum_{t \in \widetilde{T}} I(t)$$

Donde $I(t) = i(t) \cdot p(t)$.

Entonces, si se tienen dos árboles T, T' donde sus hojas están marcadas como t y t' tal como se ve en la siguiente figura:

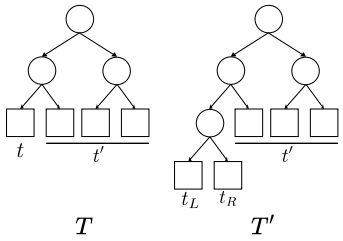


Figura 7: A la izquierda un árbol T. A derecha una extensión del árbol T notada como T'

Se puede esperar que si la división que separó el nodo t es buena, entonces la impureza del árbol $I(T')$ sea menor que $I(T)$, entonces usando la Definición 7:

$$\begin{cases} I(T) = I(t) + \sum_{t' \in \widetilde{T} - \{t\}} I(t') \\ I(T') = I(t_L) + I(t_R) + \sum_{t' \in \widetilde{T} - \{t\}} I(t') \end{cases}$$
$$\implies I(T) - I(T') = I(t) - I(t_L) - I(t_R)$$

Con la idea anterior, las divisiones óptimas deben entonces maximizar $I(T) - I(T')$. Lo cual da cabida a la siguiente definición:

Definición 8. *Sea* t *un nodo y sea* $\mathtt{s}$ *una división de* t *en dos nodos* t_L, t_R. *Entonces, el cambio total de impureza debido a* s *está dado por:*

$$\Delta I(\mathtt{s}, t) = I(t) - I(t_L) - I(t_r)$$

De lo anterior se puede notar que:

$$\Delta I(\mathtt{s}, t) = i(t) \cdot p(t) - i(t_L) \cdot p(t_L) - i(t_R) \cdot p(t_R) = p(t) \cdot \Delta i(\mathtt{s}, t)$$

Por lo tanto la pregunta $\mathtt{s}$ que maximiza $\Delta I(\mathtt{s}, t)$ también maximiza $\Delta i(\mathtt{s}, t)$.

Cabe agregar que la función de reducción de impureza es siempre mayor o igual que 0, en (Breiman y col., 1984, p. 126) hay una breve prueba de ello bajo la suposición de que la función de impureza (Definición 3) es cóncava en el espacio de vectores de probabilidad.

Usando la función de reducción de impureza repetidas veces en la base de datos se puede crear un primer árbol, que tendrá como resultado hojas muy puras. No obstante, aún hace falta decidir que clase representante tendrán dichas hojas.

1.3. Decidiendo la clase representante en las hojas

Aunque usualmente se tiene como regla tomar como clase representante de un nodo la que tenga mayor proporción en este, pueden existir clases que en su objeto de estudio no se quieren tratar como totalmente distintas. Por lo tanto, se introducirá el costo de clasificación errónea, el cual ayudará a tratar ese tipo de situaciones.

Definición 9. *Sea i, j clases que pertenecen a el conjunto C. Sea $C(i \mid j)$ el costo de clasificar erróneamente la clase j como si fuese la clase i. Dicho costo, debe cumplir que $C(j \mid j) = 0$*

Por ejemplo, $C(i \mid j) = 1 - \delta_{ij}$ es una función de costo, donde:

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Supongamos que se declara que la etiqueta de una hoja t es la clase i, entonces el costo esperado de clasificar erróneamente el nodo t con dicha clase es:

$$\sum_j C(i \mid j) \cdot p(j \mid t)$$

Como el caso particular, si $C(i \mid j) = 1 - \delta_{ij}$ entonces:

$$\sum_j C(i \mid j) \cdot p(j \mid t) = \sum_j p(j \mid t) - \sum_j \delta_{ij} p(j \mid t) = 1 - (0 + \cdots + p(i \mid t) + \cdots + 0) = 1 - p(i \mid t)$$

Ahora bien, se espera que la clase representante de un nodo t tenga el menor costo esperado de clasificación errónea. Entonces, la clase representante del nodo debe definirse como:

Definición 10. *Sea t un nodo. La etiqueta de la clase $j^*(t)$ de t esta definida como:*

$$j^*(t) = arg\min_i \left(\sum_j C(i \mid j) \cdot p(j \mid t) \right)$$

Por ejemplo, si $C(i \mid j) = 1 - \delta_{ij}$ entonces:

$$j^*(t) = arg\min_i (1 - p(i \mid t)) = arg\max_i p(i \mid t) = arg\max_i \left(\frac{N_i(t)}{N(t)} \right) = arg\max_i N_i(t)$$

De el anterior ejemplo, se obtiene la regla usual para determinar el representante de un nodo; tomar la clase que tiene mayor frecuencia.

Ahora bien, se tienen todos los elementos para construir un primer árbol de clasificación. En primer lugar, se necesita buscar la variable que sea parte de la primera pregunta para dividir la raíz $\mathcal{L}$. Para ello, se formularán diferentes preguntas con cada una de las variables, luego se tomará la pregunta que tenga mayor puntaje en la función de disminución de impureza visto en la (Definición 5). Con la primera división obtenida, se realiza de nuevo el procedimiento en los nodos resultantes impuros. Al árbol resultante, se le notará como $T_{\text{máx}}$

Para comenzar en nuestro caso, en cada variable se pueden formular infinitas preguntas; no obstante, dado que cada una de las variables es cuantitativa, la estrategia es formular preguntas de la forma:

¿Se cumple qué $Variable \leq a$?

Donde a es un número cualquiera (conocido como *Umbral* o *Threshold*). Dado que cada umbral produce una partición al nodo, se necesita generar un conjunto finito de umbrales para probar cual divide mejor el nodo. Una de las formas de generar dichos umbrales es de la siguiente manera: Supongamos que una variable tiene los siguientes datos omitiendo los duplicados y organizándolos de menor a mayor $\{1, 3, 4, 5\}$. Entonces el conjunto de umbrales, son los puntos medios que se pueden generar de la anterior lista, dando como resultado $\{2, 3.5, 4.5\}$. Entonces, generando umbrales para todas las variables y observando la disminución de la impureza en cada uno de los casos, se encontró que la variable "Alto del Petalo" con el umbral 2.45 tiene el mayor puntaje. Entonces, el árbol comenzará de la siguiente forma:

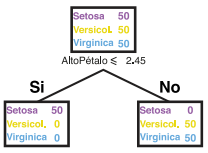


Figura 8: Árbol de decisión para los datos mostrados en la Figura 3

Usando el mismo proceso anterior hasta que las hojas queden lo mas puras posibles, se obtiene:

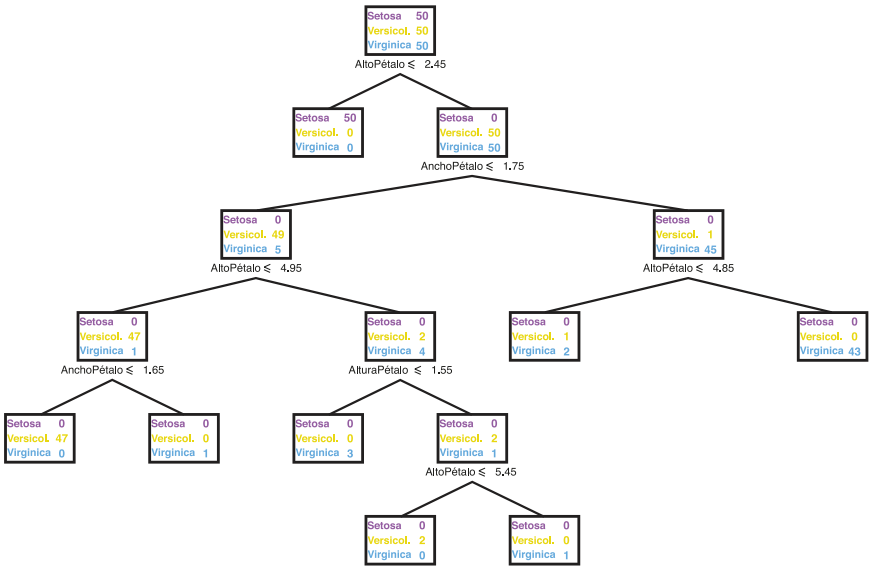


Figura 9: Árbol de clasificación extendido al máximo de la base de datos Figura 3.

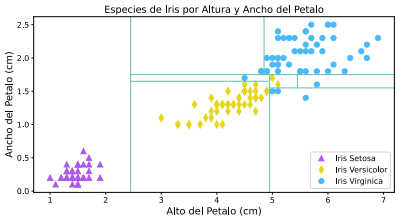


Figura 10: Visualización de la partición de la base de datos causada por el árbol en la Figura 9

Se ha creado el primer árbol. No obstante, dicho árbol debe recortarse (podar), pues la anterior gráfica muestra un sobre ajuste en los datos. Antes de introducir el podado, se necesitan algunas definiciones y además, se probará que al extender un árbol, los errores de clasificación de este decrecen.

1.4. El Costo en el Error de Clasificación

Los árboles de clasificación pueden estar sujetos a errores al clasificar los datos. Por lo tanto, en esta sección de definirá dicho costo, este ultimo sirve para evaluar el rendimiento de los árboles.

Definición 11. *Se define el costo de clasificación errónea $r(t)$ de un nodo t por:*

$$r(t) = \min_i \left(\sum_j C(i \mid j) \cdot p(j \mid t) \right)$$

Sea $R(t) = r(t) \cdot p(t)$, entonces se define el costo de clasificación errónea $R(T)$ de un árbol T como:

$$R(T) = \sum_{t \in \widetilde{T}} R(t) = \sum_{t \in \widetilde{T}} r(t) \cdot p(t)$$

Con las anteriores definiciones, mostrará que cualquier división en un nodo hace decrecer el costo de clasificación errónea. En otras palabras, entre más se expanda el árbol, menor error de clasificación tendrá.

Teorema 3. *Sea t un nodo y t_L, t_R una división de t. Entonces:*

$$R(t) \geq R(t_l) + R(t_R)$$

La igualdad se obtiene si $j^*(t) = j^*(t_L) = j^*(t_R)$

A continuación, se describirá el proceso de podado. Este consiste en extender un árbol hasta el máximo como en la Figura 9, y luego quitar nodos de abajo hacia arriba con un procedimiento especial.

2. El Tamaño Adecuado del Árbol

Dado que un árbol de clasificación totalmente extendido puede causar problemas de sobre ajuste, su tamaño se debe "regularizar" penalizándolo por la cantidad de hojas que el árbol tenga. Para ello, se definirá una generalización del costo de clasificación errónea en un árbol visto en la sección anterior:

Definición 12. *Sea α un número no negativo. Para cualquier nodo t, se define:*

$$R_\alpha(t) = R(t) + \alpha$$

Y a su vez se define para un árbol T:

$$R_\alpha(T) = R(T) + \alpha \left| \widetilde{T} \right|$$

Donde es el número de nodos terminales (hojas) del árbol.

Es la medida $R_\alpha(T)$ la que define hasta que punto se debe extender el árbol. En comparación de $R(T)$, la medida $R_\alpha(T)$ no necesariamente disminuye cuando se extiende el árbol pues el término $\left|\widetilde{T}\right|$ crecerá.

Como se mencionaba anteriormente, el proceso del podado empieza quitando algunos nodos del árbol $T_{\text{máx}}$, no obstante este proceso tiene algunas notaciones. Si se tiene un árbol T como en la Figura 2.a) y t_2 es un nodo de T entonces el sub-árbol T_{t_2} es el árbol formado por todos los descendientes del nodo t_2 incluido el mismo, tal como se ve en la Figura 2.b). Ahora bien, si se quiere cortar el sub-árbol T_{t_2} de T entonces se removerán todos los descendientes de t_2 en el árbol T dando como resultado el árbol notado como $T - T_{t_2}$, ilustrado en Figura 2.c).

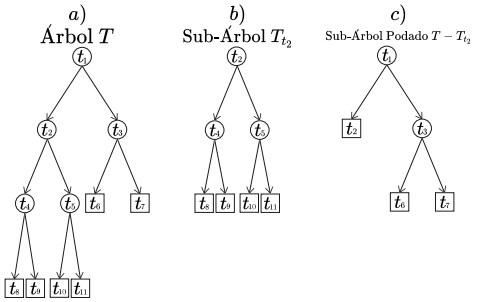


Figura 1l: Ejemplo del proceso de podado para el modelo de árbol

Formalizando el podado, se puede definir:

Definición 13. *Un sub-árbol podado T_1 de un árbol T es un sub-árbol obtenido por el siguiente proceso:*

1. *Tomar un nodo t*
2. *Remover todos los descendientes de t en el árbol T, a excepción de t mismo.*

De hecho, si T_1 es un sub-árbol podado de T entonces se notará como $T_1 \preceq T$.

Ahora bien, se construirá una serie de sub árboles podados del árbol extendido $T_{\text{máx}}$; no obstante, dichos sub árboles serán cuidadosamente escogidos. Pues deben cumplir lo siguiente:

Definición 14. *Dado un árbol T y un número α no negativo, el* ***sub-árbol podado óptimo*** *de T notado como $T(\alpha)$ debe cumplir las siguientes condiciones:*

1. $T(\alpha) \preceq T$
2. *Para cualquier sub-árbol podado $T' \preceq T$ se cumple que $R_\alpha(T(\alpha)) \leq R_\alpha(T')$*
3. *Si un sub-árbol podado $T' \preceq T$ satisface $R_\alpha(T(\alpha)) = R_\alpha(T')$ entonces $T(\alpha) \preceq T'$*

En consecuencia a la anterior definición se demostrará el siguiente teorema que muestra la existencia y unicidad de un sub-árbol podado óptimo del árbol T para un $\alpha \geq 0$ fijo:

Teorema 4. *Sea T un árbol, entonces el sub-árbol podado óptimo $T(\alpha) \preceq T$ existe y es único.*

Demostración: La demostración se hará con una inducción completa sobre el número de hojas del árbol T. Si $\left|\widetilde{T}\right| = 1$ entonces el árbol solo consiste en el nodo raíz. Por lo tanto es trivial que $T(\alpha)$ es el nodo raíz también. Ahora bien, supongamos que la hipótesis se cumple para todo número natural i tal que $i \leq n$ donde n es un natural cualquiera, entonces se probará la hipótesis para $n+1$.

Sea $\left|\widetilde{T}\right| = n+1$ y t_1 el nodo raíz del árbol. Puesto que el número de nodos terminales es mayor que 1 deben existir nodos subyacentes t_L, t_R de t_1. Sea T_{t_L} el sub-árbol formado por todos los nodos descendientes a t_L incluido el mismo y, sea T_{t_R} el sub-árbol formado por todos los nodos descendientes a t_R incluido el mismo. A los anteriores sub-árboles T_{t_L}, T_{t_R} se les llamará *ramas principales de T*.

Dado que $\left|\widetilde{T}\right| = \left|\widetilde{T_{t_L}}\right| + \left|\widetilde{T_{t_R}}\right|$, se puede afirmar:

$$\left|\widetilde{T_{t_L}}\right| < \left|\widetilde{T}\right| \qquad \left|\widetilde{T_{t_R}}\right| < \left|\widetilde{T}\right|$$

Entonces, usando la hipótesis de inducción, existen los sub-árboles podados óptimos $T_{t_L}(\alpha), T_{t_R}(\alpha)$ en T_{t_L}, T_{t_R} respectivamente. Ahora bien, se mostrará que el árbol T_1 formado por la raíz t_1 y con ramas principales T_{t_L}, T_{t_R} es el sub-árbol podado óptimo $T(\alpha)$. Para probar lo anterior se necesita considerar dos casos:

Supongamos que $R_\alpha(\{t_1\}) \leq R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha))$. Sea $T' \preceq T$ tal que $\left|\widetilde{T'}\right| > 1$, y T'_{t_L}, T'_{t_R} son sus ramas principales. Por lo tanto, $T'_{t_L} \preceq T_{t_L}$ y $T'_{t_R} \preceq T_{t_R}$. Entonces, usando la Definición 14 se tiene que $R_\alpha(T_{t_L}(\alpha)) \leq R_\alpha(T'_{t_L})$ y $R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T'_{t_R})$, entonces:

$$\begin{aligned}
R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha)) &\leq R_\alpha(T'_{t_L}) + R_\alpha(T'_{t_R}) \\
&\leq \left(R(T'_{t_L}) + R(T'_{t_R})\right) + \alpha\left(\left|\widetilde{T'_{t_L}}\right| + \left|\widetilde{T'_{t_R}}\right|\right) \\
&\leq R(T') + \alpha\left(\left|\widetilde{T'_{t_L}}\right| + \left|\widetilde{T'_{t_R}}\right|\right) && \left(R(T'_{t_L}) + R(T'_{t_R}) = R(T')\right) \qquad (1) \\
&\leq R(T') + \alpha\left(\left|\widetilde{T'}\right|\right) && \left(\left|\widetilde{T'_{t_L}}\right| + \left|\widetilde{T'_{t_R}}\right| = \left|\widetilde{T'}\right|\right) \\
R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha)) &\leq R_\alpha(T')
\end{aligned}$$

Dado que $R_\alpha(\{t_1\}) \leq R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha))$ entonces:

$$R_\alpha(\{t_1\}) \leq R_\alpha(T')$$

Puesto que no puede haber un sub-árbol podado de $\{t_1\}$ y este ultimo es un sub-árbol podado de T entonces se puede afirmar que $T(\alpha) = \{t_1\}$

Por otro lado, si $R_\alpha(\{t_1\}) > R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha))$, entonces $T(\alpha)$ no puede ser el nodo raíz t_1 del árbol T. Por lo tanto, sea T' un sub-árbol podado de T no trivial donde T'_{t_L}, T'_{t_R} son sus ramas principales. Entonces, usando la Definición 14 se tiene que $R_\alpha(T_{t_L}(\alpha)) \leq R_\alpha(T'_{t_L})$ y $R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T'_{t_R})$, por lo tanto:

$$R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T'_{t_L}) + R_\alpha(T'_{t_R})$$

Por lo tanto:

$$\begin{aligned} R_\alpha(T_1) &= R(T_1) + \alpha\left|\widetilde{T_1}\right| \\ &= \left(R(T_{t_L}(\alpha)) + \alpha\left|\widetilde{T_{t_L}(\alpha)}\right|\right) + \left(R(T_{t_R}(\alpha)) + \alpha\left|\widetilde{T_{t_R}(\alpha)}\right|\right) \\ &= R_\alpha\left(T_{t_L}(\alpha)\right) + R_\alpha\left(T_{t_R}(\alpha)\right) \\ R_\alpha(T_1) &\leq R_\alpha(T'_{t_L}) + R_\alpha(T'_{t_R}) \end{aligned} \tag{2}$$

Entonces, de manera similar a lo mostrado en (1) se tiene que:

$$R_\alpha(T_1) \leq R_\alpha(T')$$

Ahora bien, solo hace falta demostrar que para cualquier sub-árbol podado T' de T_1 tal que $T_1 \preccurlyeq T'$ entonces $R_\alpha(T_1(\alpha)) \neq R_\alpha(T')$

Sea T' un sub-árbol podado de T con ramas principales T'_L, T'_R, tal que $T_1 \not\preceq T'$, sin perdida de generalidad se puede asumir que $T'_L \prec T_{t_L}(\alpha)$ y $T_{t_R}(\alpha) \preceq T'_R$, entonces por la Definición 14 se tiene que:

$$R_\alpha\left(T_{t_L}(\alpha)\right) < R_\alpha(T'_L) \qquad R_\alpha\left(T_{t_R}(\alpha)\right) \leq R_\alpha(T'_R)$$

Por lo tanto:

$$R_\alpha\left(T_{t_L}(\alpha)\right) + R_\alpha\left(T_{t_R}(\alpha)\right) < R_\alpha(T'_L) + R_\alpha(T'_R)$$

Así, con cálculos similares en (1), (2) se tiene que:

$$R_\alpha(T_1) < R_\alpha(T')$$

Entonces, se puede concluir que $T_1 = T(\alpha)$.

Puesto que se demostró la hipótesis para un árbol de $n+1$ hojas, entonces por inducción matemática queda demostrado que $T(\alpha)$ existe y es única. ■

Ahora bien, se mostrará el proceso de podado. Primeramente del árbol $T_{\text{máx}}$ se halla el sub-árbol podado óptimo $T(0)$, la forma de hacerlo es la siguiente. Sean t_L, t_R los nodos descendientes de un nodo t cualquiera en el árbol $T_{\text{máx}}$, se sabe por el Teorema 3 que $R(t) \geq R(t_L) + R(t_R)$, por lo tanto se podará de $T_{\text{máx}}$ los nodos t tal que $R(t) = R(t_L) + R(t_R)$, al árbol resultante se le notará T_1. Lo anterior, se puede observar como un teorema:

Teorema 5. *El sub-árbol T_1 del resultado anterior es igual al sub-árbol podado óptimo $T(0)$ de $T_{\text{máx}}$*

Ahora bien, el propósito ahora es tratar de generalizar el teorema anterior con el fin de podar árboles y que sean equivalentes a un $T(\alpha)$ con $\alpha > 0$. Para ello, se mostrará el siguiente teorema:

Teorema 6. *Para cualquier nodo no terminal* $t \in T(0)$ *se cumple que:*

$$R(T_t) < R(t)$$

A modo de tratar de generalizar el teorema anterior, se desea que se cumpla:

$$R_\alpha(T_t) < R_\alpha(t)$$

Donde t es un nodo no terminal de un árbol T. No obstante, para que la anterior desigualdad se cumpla se necesita observar la siguiente condición sobre el número α.

Sí se cumple que $R_\alpha(T_t) < R_\alpha(t)$ entonces se obtiene la equivalencia:

$$\begin{aligned} R(T_t) + \alpha\left|\widetilde{T}_t\right| &< R(t) + \alpha \\ \alpha &< \frac{R(t) - R(T_t)}{\left|\widetilde{T}_t\right| - 1} \end{aligned} \tag{3}$$

Desde luego $\dfrac{R(t) - R(T_t)}{\left|\widetilde{T}_t\right| - 1} > 0$ pues por el Teorema 6 se tiene que $R(t) - R(T_t) > 0$ y dado que t es no terminal entonces $\left|\widetilde{T}_t\right| - 1 > 0$.

Ahora bien, si α toma valores muy grandes entonces habrá nodos que no satisfagan la desigualdad (3). Para encontrar dichos nodos, se definirá la función $g_1(t)$ para un nodo $t \in T(0)$ por:

$$g_1(t) = \begin{cases} \frac{R(t) - R(T_t)}{\left|\widetilde{T}_t\right| - 1} & si \quad t \notin \widetilde{T(0)} \\ \infty & \text{en otro caso} \end{cases}$$

Entonces, se define:

$$\alpha_2 = \min_{t \in T(0)} g_1(t)$$

Por lo tanto, si para un número α se cumple que $\alpha < \alpha_2$ entonces la desigualdad (3) se satisface para todos los nodos $t \in T(0)$ no terminales. Sin embargo, sí $\alpha = \alpha_2$ entonces existirán algunos nodos que conviertan la inecuación (3) en una igualdad dichos nodos se llamarán *nodos de enlace mas débiles.*

En consecuencia, sí t_1' es un nodo de enlace débil, entonces:

$$g_1(t_1') = \alpha_2 = \min_{t \in T(0)} g_1(t)$$

Dado que cuando se poda el árbol $T(0)$ en el nodo t_1' se obtiene el sub-árbol podado $T(0) - T_{t_1'}$. Entonces:

$$R_{\alpha_2}(T(0)) = R_{\alpha_2}\left(T(0) - T_{t_1'}\right)$$

Pues $R_{\alpha_2}\left(T_{t_1'}\right) = R_{\alpha_2}(t)$. Por otro lado, si hay un nodo de enlace mas débil entonces debe ser podado del árbol $T(0)$. A este nuevo árbol se le notará como T_2 (este ultimo cumple que $T_2 = T(\alpha_2)$).

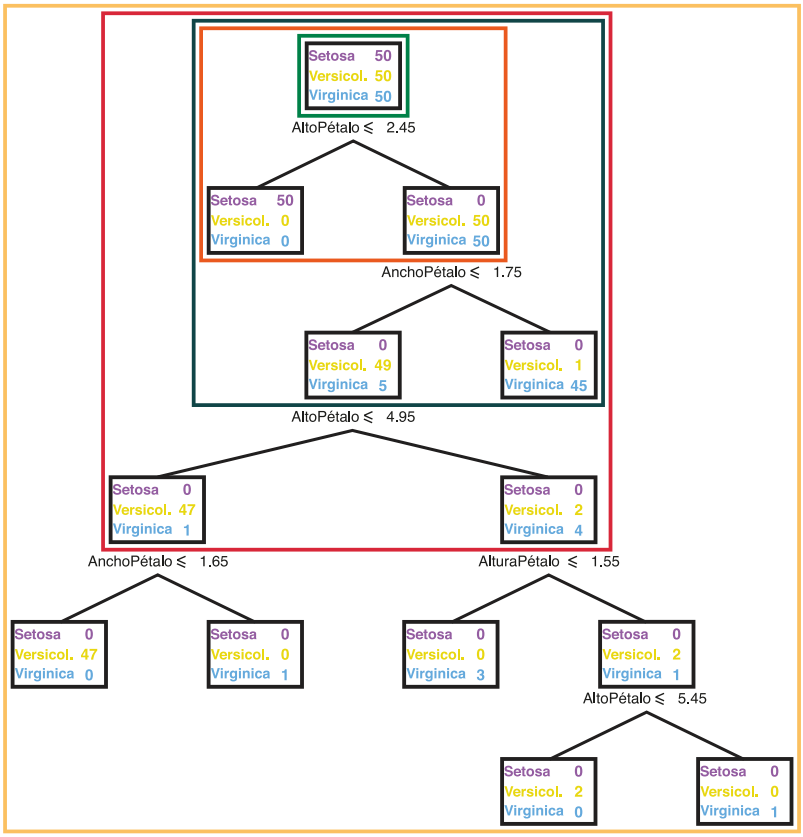
Además, si se repite el anterior proceso con el árbol T_2 se obtendrá como resultado un número α_3 y un nuevo sub-árbol de $T(0)$ notado como $T_3 = T(\alpha_3)$. Consecuentemente, repitiendo el proceso consecutivamente, se tendra como resultado los árboles:

$$T(0) = T_1 \geq T_2 \geq T_3 \geq \cdots \geq \{t_1\}$$

Y la relación entre los números:

$$0 = \alpha_1 < \alpha_2 < \alpha_3 < \ldots$$

Usando el anterior procedimiento, se obtuvo el siguiente resultado:



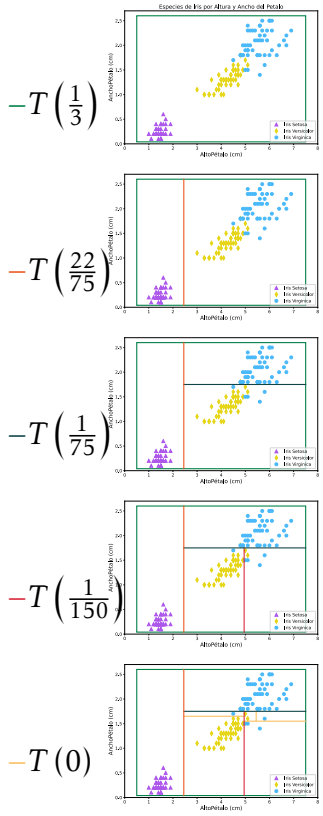


Figura 12: A izquierda, los sub-árboles optimos. A derecha, las divisiones que cada uno de los sub-árboles construyen respectivamente.

Por ende, solo hace falta observar cual de los árboles $T_1, T_2, \ldots$ tiene el mejor rendimiento, para ello se utilizará la validación cruzada; sin embargo el funcionamiento de dicho método depende del siguiente teorema:

Teorema 7. *Sea* $T_k = T(\alpha_k)$

1. *Si* $\alpha_1 \leq \alpha_2$ *entonces* $T(\alpha_2) \leq T(\alpha_1)$
2. *Si existe* α *tal que* $\alpha_k \leq \alpha < \alpha_{k+1}$ *entonces* $T(\alpha_k) = T(\alpha)$

Ahora bien, se procederá a mostrar el método de validación cruzada para observar que árbol ilustrado en Figura 12 tiene el mejor rendimiento.

3. Seleccionando al mejor árbol podado

Para evaluar el rendimiento de un árbol, se utilizan dos subconjuntos disjuntos de los datos: un conjunto de entrenamiento y un conjunto de prueba. Con el conjunto de entrenamiento se construirá el clasificador (en este caso el árbol); y por otro lado, el conjunto de prueba se evalúa en el árbol anterior para observar su precisión. Por lo tanto, se definirán algunas *cantidades teóricas* asociadas al clasificador d formado por el árbol:

Definición 15. *La probabilidad teórica de clasificar la clase* j *como una clase* i *está dada por:*

$$Q^*(i \mid j) = P\left(d(X) = i \mid C = j\right)$$

Definición 16. *El valor esperado teórico para el costo del error de clasificación de la clase* j *es:*

$$R^*(j) = \sum_i C(i \mid j)\, Q^*(i \mid j)$$

Definición 17. *El costo general teórico del clasificador* d*, está dado por:*

$$R^*(d) = \sum_j R^*(j)\, \pi(j)$$

Para esta ultima definición cabe resaltar la siguiente observación. Supongamos que $C(i \mid j) = 1 - \delta_{ij}$ entonces:

$$R^*(d) = \sum_j R^*(j)\,\pi(j) = \sum_j \sum_i (1 - \delta_{ij})\, Q^*(i \mid j)\,\pi(j) = \sum_i (1 - Q^*(i \mid i))\,\pi(i) = P(d(X) \neq C)$$

De lo anterior, se concluye que el error de costo general teórico con la condición anterior no es más que la probabilidad de que el árbol cometa errores al clasificar. Con lo anterior en mente se procederá a mostrar el método de validación cruzada.

3.1. Validación Cruzada

En la practica habitual de la validación cruzada de V-iteraciones, primero se divide los datos $\mathcal{D}$ en V conjuntos disjuntos $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_V$. Ahora bien, sea $\mathcal{D}^{(v)} = \mathcal{D} - \mathcal{D}_v$ donde $v \in \{1, 2, \ldots, V\}$.

Primeramente, usando $\mathcal{D}$ hacemos crecer el árbol $T_{\text{máx}}$ para luego obtener los alfas $0 = \alpha_1 < \alpha_2 < \ldots$ y sus respectivos árboles podados $T_1 \geq T_2 \geq \ldots \{t_1\}$ donde $T_k = T(\alpha_k)$. Desde luego los árboles $T_1, T_2, \ldots$ no se pueden usar para la validación cruzada pues se usó la muestra de aprendizaje para construirlos.

Entonces, se define $\alpha'_k = \sqrt{\alpha_k \cdot \alpha_{k+1}}$. Para cada $v \in \{1, 2, \ldots, V\}$ y $\alpha \in \{\alpha'_1, \alpha'_2, \ldots\}$, se usará $\mathcal{D}^{(v)}$ para hacer crecer $T^{(v)}_{\text{máx}}$ y luego se construirá $T^{(v)}(\alpha)$ aplicando el proceso usando R_α. Dado que $\mathcal{D}_v$ no se usó para formar los árboles $T^{(v)}(\alpha)$ entonces se puede usar como conjunto de prueba para el clasificador.

Entonces usando $\mathcal{D}_v$ en lo árboles $T^{(v)}(\alpha)$, se calculará:

- $N^{(v)}_{ij}$: El número de clases j en $\mathcal{D}_v$ clasificados por $T^{(v)}(\alpha)$ como i
- $N_{ij} = \sum_i N^{(v)}_{ij}$: El número total de clases j que fueron clasificadas como i en el proceso de validación cruzada

El propósito es que para un V grande, $T^{(v)}(\alpha)$ y $T(\alpha)$ tendrán una precisión similar. Entonces:

$$Q^{CV}(i \mid j) = \frac{N_{ij}}{N_j} \approx Q^*(i \mid j) \quad R^{CV}(j) = \sum_i C(i \mid j) Q^{CV}(i \mid j) \approx R^*(j) \quad R^{CV}(T(\alpha)) = \sum_j R^{CV}(j)\pi(j) \approx R^*(T(\alpha))$$

Por lo tanto, usando el Teorema 7 se tiene que $T_{\alpha' k} = T_{\alpha_k} = T_k$. Entonces se tiene que el estimado de $R^{CV}(T_k)$ por:

$$R^{CV}(T_k) = R^{CV}(T_{\alpha_k})$$

Finalmente se toma el $T_{k'}$ entre $T_1, T_2, \ldots, \{t_1\}$ tal que:

$$R^{CV}(T_{k'}) = \min_k R^{CV}(T_k)$$

Retomando el problema de la base de datos que se esta manejando. Se obtuvo en Figura 12 que los α_k son:

$$\alpha_1 = 0, \alpha_2 = \frac{1}{150}, \alpha_3 = \frac{1}{75}, \alpha_4 = \frac{22}{75}, \alpha_5 = \frac{1}{3}$$

Entonces los α'_k serán:

$$\alpha'_1 = 0; \alpha'_2 = 0.00943, \alpha'_3 = 0.06254, \alpha'_4 = 0.31269$$

Haciendo el proceso de validación cruzada como se describió anteriormente se formó la siguiente tabla:

		$R^{CV}(T_k)$			
k	$\lvert\widetilde{T}_k\rvert$	$V=2$	$V=5$	$V=25$	$V=150$
1	7	0.04933 ± 0.0098	0.04667 ± 0.009	0.04667 ± 0.007	0.04666
2	4	0.052 ± 0.00643	0.06267 ± 0.015	0.06267 ± 0.007	0.05333
3	3	0.06533 ± 0.009	0.07333 ± 0.004	0.05467 ± 0.0049	0.04666
4	2	0.4533 ± 0.098	0.4853 ± 0.0316	0.484 ± 0.027	0.6667

Dado que los mejores resultados los tiene $k = 1$, entonces se puede deducir que el árbol $T(0)$ es el que mejor rendimiento tiene.

4. Construyendo un árbol de clasificación para la base de datos IRIS

El propósito ahora es emplear la base de datos Iris usando todas sus variables para crear un árbol óptimo con la teoría usada en los anteriores capítulos. Con fines prácticos se notará la base de datos anteriormente mencionada como $\mathcal{L}$. Entonces, se obtuvo los siguientes resultados:

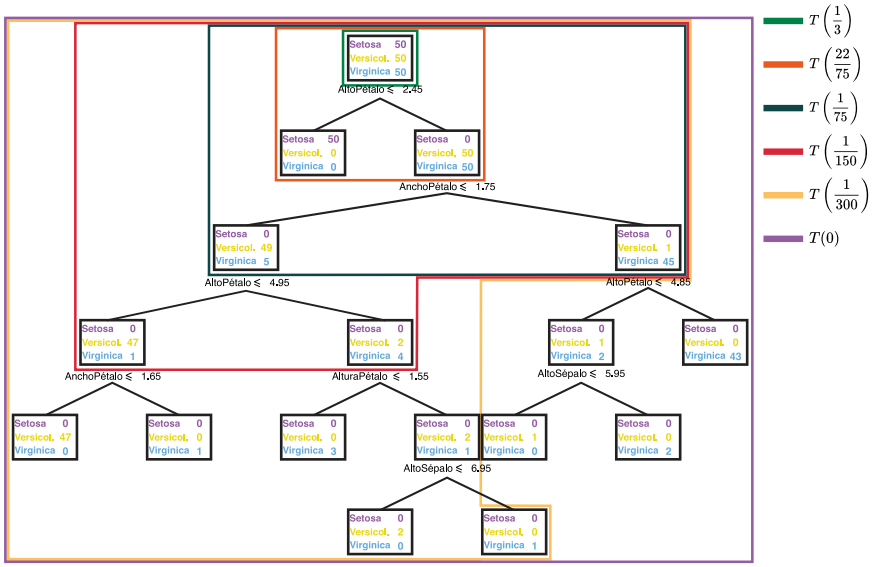


Figura 13: Árboles que son candidatos para ser el mejor

Empleando la validación cruzada para encontrar el sub-árbol que mejor se desempeña, se obtuvo que los α'_k son:

$$\alpha'_1 = 0 \quad \alpha'_2 = 0.004714 \quad \alpha'_3 = 0.009428 \quad \alpha'_4 = 0.062538 \quad \alpha'_5 = 0.31269$$

Dando como resultado la siguiente tabla:

		$R^{CV}(T_k)$				
k	$\lvert\widetilde{T}_k\rvert$	$V = 6$	$V = 15$	$V = 50$	$V = 75$	$V = 150$
1	9	0.06 ± 0.0072	0.06267 ± 0.011	0.06267 ± 0.0032	0.06267 ± 0.0032	0.06
2	7	0.0533 ± 0.004	0.05733 ± 0.0067	0.056 ± 0.0032	0.05467 ± 0.0034	0.0533
3	4	0.06533 ± 0.0096	0.064 ± 0.005	0.06533 ± 0.0076	0.05867 ± 0.0049	0.0625
4	3	0.07333 ± 0.0072	0.064 ± 0.015	0.06133 ± 0.015	0.04933 ± 0.0052	0.0466
5	2	0.468 ± 0.06	0.456 ± 0.03	0.508 ± 0.0196	0.5387 ± 0.022	0.6667

Aunque el árbol asociado a $k = 4$ muestra un buen rendimiento cuando $V = 150$, el árbol asociado a $k = 2$ tiene mejores resultados para cada V. Entonces, se tomará como árbol de clasificación a $T\left(\frac{1}{130}\right)$

5. Conclusión

Como se ha visto en el desarrollo de este trabajo, el modelo de árbol de clasificación tiene un amplio desarrollo matemático, tanto en su construcción como la validación del modelo. Además, son sencillos de entender y aplicar.

No obstante, debido a su funcionamiento simple, en algunas base de datos complejas su rendimiento puede ser bajo. Sin embargo, los árboles de clasificación son los cimientos de otros métodos más potentes (Bosques Aleatorios, Gradient Boosting).

Referencias

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.

Choi, H. I. (2017). Classification and Regression Tree (CART). https://www.math.snu.ac.kr/~hichoi/machinelearning/lecturenotes/CART.pdf

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179-188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Loh, W.-Y. (2014). Fifty years of classification and regression trees: fifty years of classification and regression trees. *International Statistical Review*, *82*(3), 329-348. https://doi.org/10.1111/insr.12016

Messenger, R. & Mandell, L. (1972). A modal search technique for predictive nominal scale multivariate analysis. *Journal of the American Statistical Association*, *67*(340), 768-772. https://doi.org/10.1080/01621459.1972.10481290

Morgan, J. N. & Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, *58*(302), 415-434. https://doi.org/10.1080/01621459.1963.10500855