Árboles de Decisión: Análisis Crítico, CART y Optimización para la Generalización de Modelos.

Julio Y. Cárdenas Carrera

Resumen

En este ensayo crítico se analiza el capítulo *"Decision Trees"* de Lior Rokach y Oded Maimon [1], reconocido por su cobertura detallada de los fundamentos teóricos y algoritmos asociados a los árboles de decisión. No obstante, el artículo presenta ciertas omisiones relevantes que limitan su aplicabilidad práctica. Aunque se enumeran múltiples algoritmos de inducción, muchos de ellos son poco conocidos o escasamente utilizados en la práctica profesional. En contraste, el algoritmo *CART (Classification and Regression Trees)* recibe una atención mínima.

Este ensayo no busca refutar la propuesta original, sino complementarla críticamente al incluir una exposición formal y detallada de *CART*, incluyendo su estructura binaria, criterios de división (como el índice Gini), y su utilidad en tareas de regresión. Además, se incorporan conceptos fundamentales que el artículo no aborda explícitamente, como el overfitting, el underfitting y las técnicas de poda, esenciales para la generalización de modelos. Se concluye que una mayor profundización en estos elementos habría fortalecido significativamente el valor didáctico y aplicado del capítulo.

1. Introducción.

Un árbol de decisión es un modelo supervisado no paramétrico, su objetivo es predecir u ordenar los datos de entrada X a la clase o valor asociado a un dato Y, reduciendo la impureza de la información en cada etapa de división cumpliendo la función (1)[2].

$$f : X \to Y \tag{1}$$

El funcionamiento del algoritmo se basa en la partición recursiva de los conjuntos de datos T en subconjuntos denominados *splits*, los cuales generan ramas que conectan con nodos hijos [3]. Estas divisiones continúan de forma recursiva hasta que alcanzan un criterio de parada, como la pureza total del nodo o una profundidad máxima definida para el árbol, la cual se conoce como su grado de profundidad.
El resultado final es una estructura en forma de árbol, Figura 1, donde cada nodo interno representa una decisión basada en el atributo con mayor información de manera jerárquica, y cada nodo hoja representa una clasificación final. El árbol comienza con el nodo raíz $\{r\}$, que contiene la primera división, y desde ahí se ramifica hacia sub-arboles T_k mediante divisiones recursivas hasta alcanzar los nodos hoja, donde la impureza, o probabilidad de error ha sido minimizada o eliminada [4].

La definición formal de una estructura de árbol es representada por medio de un Grafo Acíclicos Dirigido (o *DAG* por sus siglas en inglés *Directed Acyclic Graph*).
De acuerdo con la definición (2), el conjunto T se construye como la unión de los subcomponentes $r, T_1, T_2, \ldots, T_k$, formando de esa manera la estructura del árbol.

$$T = \{r\} \cup T_1 \cup T_2 \cup \ldots \cup T_k \tag{2}$$

$$T_k = (V, E)$$

Donde:

$$V = \{v_1, v_2, v_3, \ldots, v_k\} \quad : \text{conjunto de nodos (vértices)}$$
$$E = \{(v_1, v_2), (v_1, v_3)\} \quad : \text{conjunto de aristas (enlaces)}$$

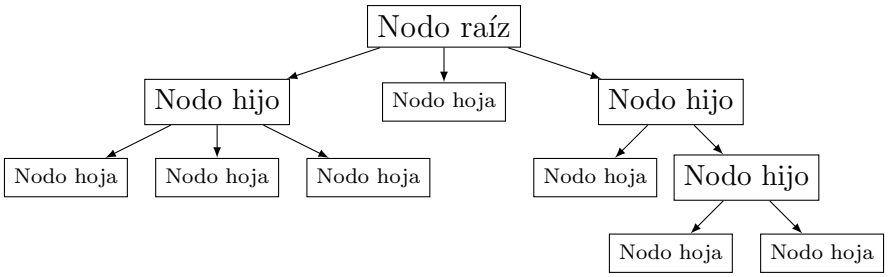

Figura 1: Diagrama de árbol

Uno de los algoritmos mas usados debido a su alta eficiencia es el algoritmo *CART* (*Classification And Regression Trees*, por sus siglas en inglés) siendo este la mejora de los algoritmos de división ID3, C4.5 y C5.0, ya que al no usar logaritmos es, computacionalmente, mas barato, lo que abre camino a entrenamientos con sistemas mas complejos [5]. Con este algoritmo, como su nombre lo menciona, podemos entrenar modelos tanto de regresión como de clasificación.

2. Tipos de árboles.

Según sea la problematica a resolver, la estructura de los arboles de decisión asi como sus criterios cambian. Los árboles de decisión pueden clasificarse en dos tipos: árboles de regresión y árboles de clasificación. Los árboles de regresión son modelos diseñados para predecir valores continuos de la variable de salida a partir de los atributos de entrada. Por otro lado, los árboles de clasificación están orientados a asignar una clase o categoría a la variable de salida realizando tareas de clasificación. De esta manera, los árboles de regresión abordan interrogantes sobre valores cuantitativos, mientras que los árboles de clasificación resuelven problemas de asignación categórica, Figura 2 [6] que construyen al árbol.

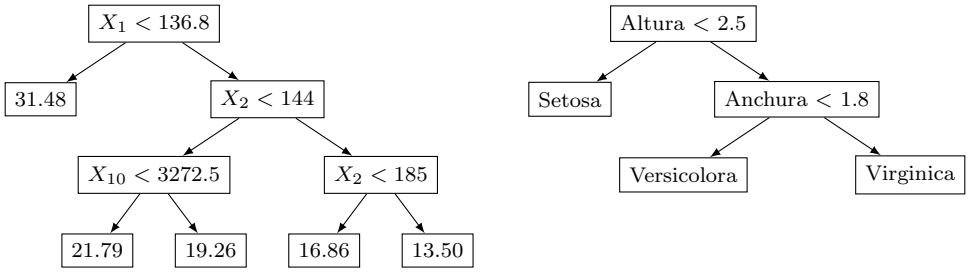

Figura 2: Comparación entre un árbol de regresión y un árbol de clasificación.

Los arboles generados con CART solamente pueden ser binarios, en otras palabras, el nodo $\{r\}$ padre solo puede tener dos nodos hijos, dividiendo el conjunto D en: nodo izquierdo T_{izq}, para valores X menores o iguales al umbral $t \in \mathbb{R}$, y nodo derecho T_{der}, para valores X mayores a t. De esta manera, la representación (3), muestra al nodo hijo izquierdo y nodo hijo derecho.

$$T = \{r\} \cup T_{izq} \cup T_{der} \tag{3}$$

Tal que:

$$T_{izq} = \{(x, y) \in D : x_j \leq t\}$$
$$T_{der} = \{(x, y) \in D : x_j > t\}$$

3. Criterios de división.

Para que un árbol pueda tomar decisiones, deberá comparar todas las posibles respuestas con respecto a los atributos como se describió en la función (1).

3.1. Criterios de clasificación.

Si bien la entropía ϕ es una métrica robusta teóricamente, su uso intensivo de logaritmos puede volverse costoso computacionalmente en grandes volúmenes de datos. Por eso, algunos especialistas argumentan que su relevancia es más conceptual que práctica, siendo reemplazada por el índice Gini en la mayoría de los algoritmos modernos debido a su simplicidad.

3.1.1. Índice Gini.

El índice Gini es un criterio de división que mide la probabilidad p de que un elemento seleccionado aleatoriamente sea clasificado incorrectamente.

$$Gini = 1 - \sum_{i=1} p_i^2 \tag{4}$$

$$p_i = \frac{n_i}{|N|}$$

Para realizar el *Split* se busca con la suma ponderada el valor mas pequeño, debido a que corresponde con el atributo X_j, con el cual, después de dividir, tendrá menor impureza.

$$Gini_{Split} = \frac{n_{izq}}{|N|} Gini_{izq} + \frac{n_{der}}{|N|} Gini_{der}$$

3.2. Criterios para regresión.

3.2.1. Error Cuadrático Medio (MSE).

El criterio MSE es una de las métricas más utilizadas en la regresión lineal, utilizada para evaluar la precisión de los modelos de regresión. El MSE realiza una medida promedio entre el valor observado y y el valor esperado $\hat{y}$, realizando el split en búsqueda del mejor atributo y el mejor umbral [7]. Sin embargo, la regresión derivada de un modelo CART no es lineal, si no un regresión escalonada (*piecewise constant regression*), ya que al utilizar esta metrica se obtiene mayor sensibilidad a valores numéricamente distantes del resto de datos, también conocidos como *outliers*

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5}$$

4. *Overfitting*

A pesar de la gran eficiencia de los árboles de decisión, una limitación notable es su propensión al sobre ajuste o *overfitting*. El *overfitting*, surge cuando un modelo es excesivamente complejo y se ajusta demasiado a los datos de entrenamiento. Es crucial encontrar un equilibrio adecuado entre la complejidad del modelo y la cantidad y calidad de datos de entrada, ya que una gestión inadecuada puede generar su contraparte: el *underfitting*.
El *underfitting* ocurre cuando el modelo es demasiado simple o cuando la cantidad de datos de un entrenamiento es insuficiente. Al medir el error entre el valor real y el valor predicho, un modelo sub ajustado mostrará un alto rango de error tanto en los datos de entrenamiento como en los nuevos datos de entrada, lo que puede causar errores graves de clasificación y predicción

de la variable salida. Por otro lado, el sobre ajuste en un modelo minimiza el error en los datos de entrenamiento, apegándose excesivamente a las características de estos datos, impidiendo generalizar de manera efectiva los nuevos datos de entrada, brindando una capacidad limitada en la clasificación y predicción de datos.
En la Figura 3 se representa la diferencia entre los ajustes de modelos: *underfitting, overfitting* e ideal [8] [9].

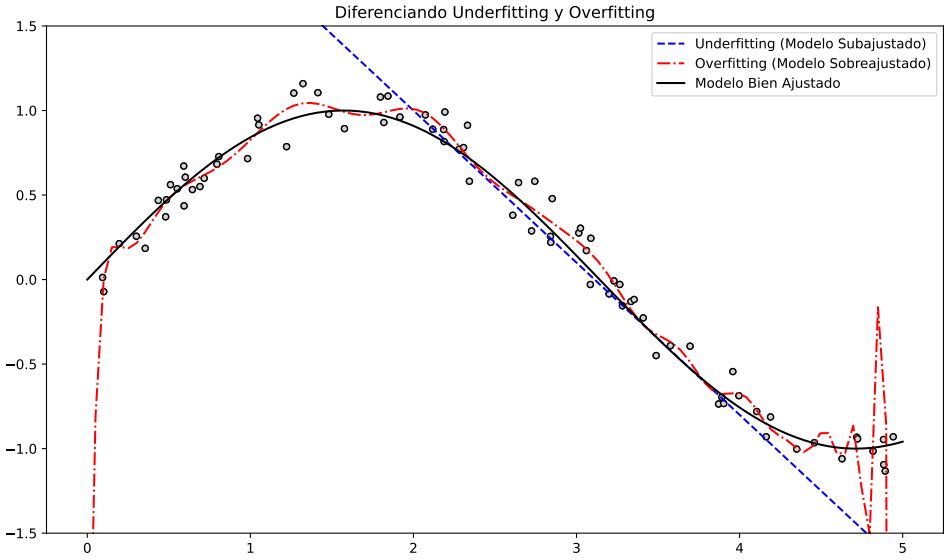

Figura 3: Diferencias entre ajustes de modelos

5. Poda

Los árboles de decisión son muy susceptibles al overfitting como se menciono anteriormente, el libre crecimiento del mismo refleja un resultado ajustado y no generativo. Para evitar estos casos se utilizan técnicas de poda durante y después de su entrenamiento.

La técnica de poda previa es utilizada en modelos como precaución, evitando su amplio crecimiento del arbol donde sus hojas tienen valores pequeños o insignificantes, este proceso se logra cortando ramas y convirtiendo nodos de decisión en hojas. Por otro lado, la técnica de poda posterior conduce a un modelo un poco mas ajustado pero necesita realizar cálculos mas rigurosos. En la practica, es mas común observar la técnica de poda posterior, ya que es mas complejo saber en donde hay que cortar la rama antes de que este crezca mas [10].

Algunos de los algoritmos de poda mas comunes son: *Cost-Complexity Pruning* utilizado en técnicas poda posterior y *Max depth* para poda previa.[11]

6. Conclusión

Este ensayo crítico ha demostrado que, si bien el capítulo "Decision Trees"de Lior Rokach y Oded Maimon ofrece una cobertura exhaustiva de los fundamentos teóricos y algoritmos de árboles de decisión, presenta omisiones significativas que limitan su aplicabilidad práctica. Se ha argumentado que una profundización en algoritmos de uso extendido como CART (Classification and Regression Trees), el cual es computacionalmente más eficiente al no usar logaritmos y permite entrenar modelos de regresión y clasificación, es crucial para una comprensión completa del campo.
Adicionalmente, se ha enfatizado la importancia de abordar conceptos esenciales como el overfitting y el underfitting. El overfitting, que se produce cuando el modelo se ajusta excesivamente

a los datos de entrenamiento impidiendo una generalización efectiva a nuevos datos , y el underfitting, que ocurre cuando el modelo es demasiado simple o los datos de entrenamiento son insuficientes, son fenómenos cruciales a comprender para desarrollar modelos de decisión robustos. Para contrarrestar el overfitting, se ha destacado la relevancia de las técnicas de poda, como la poda previa y la poda posterior, las cuales son indispensables para controlar la complejidad del árbol y mejorar su capacidad de generalización.
En definitiva, la inclusión de una explicación formal y detallada de CART, junto con una discusión explícita sobre el overfitting, el underfitting y las técnicas de poda, habría fortalecido significativamente el valor didáctico y la aplicabilidad del capítulo original. Estas adiciones habrían provisto a los lectores de herramientas y conocimientos más completos y pertinentes para la implementación y optimización de árboles de decisión en contextos reales.

Referencias

[1] L. Rokach and O. Maimon, "Decision trees," in *Data Mining and Knowledge Discovery Handbook*. Springer, 2005, pp. 165–192.

[2] IBM. (s.f.) ¿qué es un árbol de decisión? https://www.ibm.com/mx-es/think/topics/decision-trees.

[3] ——. (2025) Fondo de los árboles de decisión. https://www.ibm.com/docs/es/db2/11.5.x?topic=trees-background.

[4] A. Carlos, "Modelos de aprendizaje automoático mediante árboles de decisión," Universidad del Centro de Estudios Macroeconómicos de Argentina, Buenos Aires, Serie Documentos de Trabajo 778, 2021. [Online]. Available: https://hdl.handle.net/10419/238403

[5] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1st ed. New York: Chapman and Hall/CRC, 1984.

[6] M. Ignacio, "Árboles de clasificación y regresión," Universitat de València, Valencia, España, Tech. Rep., s.f. [Online]. Available: https://www.uv.es/mlejarza/actuariales/tam/arbolesdecision.pdf

[7] K. Stewart. (2025) error cuadrático medio. [Online]. Available: https://www.britannica.com/topic/linear-regression

[8] J. I. Bagnato. (2018) Qué es overfitting y underfitting y cómo solucionarlo. [Online]. Available: https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/

[9] MathWorks. (s.f.) Introducción al overfitting. [Online]. Available: https://la.mathworks.com/discovery/overfitting.html

[10] Analytics Vidhya. (2024) All about decision tree from scratch with python implementation. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-python-implementation/

[11] The Pennsylvania State University. (s.f.) Minimal cost-complexity pruning. [Online]. Available: https://online.stat.psu.edu/stat857/node/60/