EST-25134: Aprendizaje Estadístico

Profesor: Alfredo Garbuno Iñigo — Primavera, 2023 — Extensiones Arboles.
Objetivo: Que veremos.
Lectura recomendada: Referencia.

1. INTRODUCCIÓN

Los árboles de decisión por construcción son insensibles a:

- Variables continuas con *outliers*.
- Variables continuas con diferentes escalas.
- Variables categóricas (no se necesitan transformar a *dummies*).

El algoritmo de ajuste de un árbol de decisión, en regresión por ejemplo, busca minimizar la función de pérdida

$$R_\alpha(T) = \sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T|\,, \tag{1}$$

donde $\hat{y}_{R_m} = \sum_{i:x_i \in R_m} y_i / n_m$ es el promedio de las respuestas en la región m-ésima, a tráves de ir buscando secuencialmente las regiones R_m que logren la máxima disminución de RSS.

La búsqueda se puede lograr por medio de un algoritmo voraz (*greedy*) que escoge variables y puntos de corte.

1.0.1. Para pensar: En nuestra última discusión sobre árboles de decisión es que tienen un sesgo al utilizar variables categóricas con muchas etiquetas. ¿Por qué?

1.1. Motivación

Consideremos el experimento siguiente. Tenemos una colección de 7 atributos y una respuesta. Los datos son generados de manera aleatoria.

```
set.seed(108727)
generate_data(nsamples = 100) ▷
    print(n = 5)
```

De manera artificial consideraremos que la respuesta y está relacionada con los demás atributos (sabemos que no es cierto).

```
fit_tree ← function(engine){
  data_train ← generate_data()

  tree_spec ← decision_tree(tree_depth = 2) ▷
    set_engine(engine) ▷
    set_mode("regression")

  tree_spec ▷
    fit(y ∼ ., data = data_train) ▷
    extract_fit_engine() ▷
    vi() ▷
    filter(Importance > 0) ▷
    mutate(rank = min_rank(desc(Importance)))
}
```

Repetimos este proceso (generar datos aleatorios y ajustar un árbol) un número determinado de veces y registramos cuántas veces cada atributo fue utilizado en el nodo raíz.

En contraste, con un modelo denominado `conditional trees` somos capaces de evitar ese sesgo. Incluso podemos evitar escoger variables de corte que no tienen asociación con la respuesta.

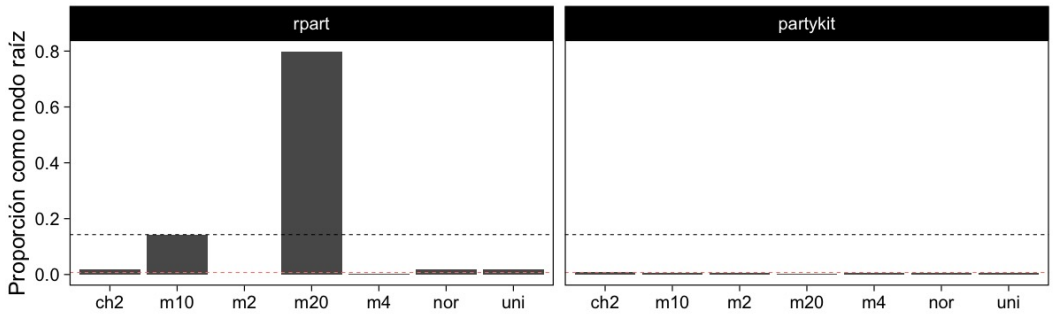

FIGURA 1. *La línea negra representa la probabilidad de seleccionar una variable al azar como nodo raíz ($p = 1/7$) y la cruva salmón representa la tolerancia de error (tasa de falsos positivos) ajustada ($\alpha = 0.05/7$).*

1.2. Ejemplos de pruebas de hipótesis de independencia

2. EJEMPLO DE PARTYKIT

3. CONCLUSIONES

El modelo de `CTree` es un modelo:

1. que utiliza pruebas de hipótesis para determinar variables y puntos de corte;
2. tiene un mecanismo de selección insesgado;
3. no requiere mucho post-procesamiento (poda);

De acuerdo a Greenwell [1] , aún cuando `CTree` tiene mejores propiedades estadística que `CART` hay un uso generalizado por el último debido a herramientas de código abierto.

Además, de acuerdo a Loh [2] mientras un árbol se escoja por cuestiones predictivas y no por inferencia tiene un riesgo bajo al utilizar procedimientos con sesgo. Por otro lado, validación cruzada puede ayudar a eliminar ramas redundantes durante el proceso de poda (mientras tengamos pocos atributos y un número suficiente de datos).

REFERENCIAS

[1] B. M. Greenwell. *Tree-Based Methods for Statistical Learning in R*. Chapman and Hall/CRC, Boca Raton, first edition, may 2022. ISBN 978-1-00-308903-2. . 2

[2] W.-Y. Loh. Fifty Years of Classification and Regression Trees. *International Statistical Review / Revue Internationale de Statistique*, 82(3):329–348, 2014. ISSN 0306-7734. 2