

EST-25134: Aprendizaje Estadístico

Profesor: Alfredo Garbuno Iñigo — Primavera, 2023 — Ensamblados y *Boosting*.

Objetivo: En esta sección estudiaremos otra estrategia para combinar modelos en uno sólo. La estrategia que estudiaremos en esta sección es un mecanismo altamente flexible para regresión y clasificación y actualmente representa el estado del arte para resolver problemas de predicción con datos tabulares.

Lectura recomendada: Capítulo 10 del libro Hastie et al. [1].

1. INTRODUCCIÓN

Boosting es otra estrategia de agregación de modelos. Es una herramienta bastante general que puede ser utilizada para problemas de regresión o clasificación [2]. La estrategia de **boosting** es una **estrategia secuencial** que busca mejorar la capacidad predictiva lograda anteriormente.

Usualmente con *boosting* utilizamos árboles pequeños (sesgo alto), al contrario de bosques aleatorios donde se prefieren árboles profundos (sesgo bajo).

En *boosting* el sesgo se disminuye con modelos predictivos que se encargan de distintos grupos en el conjunto de entrenamiento. La varianza se puede controlar con el parámetro de tasa de aprendizaje.

2. MODELOS ADITIVOS SECUENCIALES

En el marco de **modelos aditivos secuenciales** aplicados a regresión, consideremos el contexto de modelado por etapas. En este marco, consideremos un predictor de la forma

$$f(x) = \sum_{k=1}^M \beta_k b_k(x) = \sum_{k=1}^M T_k(x), \quad (1)$$

donde cada término es el resultado de ajustar un árbol de regresión.

Recuerda que un árbol T_k de decisión está definido por sus parámetros θ_k los cuales incluyen las variables que se utilizan para los cortes, el punto de corte y la predicción en cada una de las regiones terminales.

Resolver este problema es difícil pues se tienen que ajustar los coeficientes y los árboles al mismo tiempo. Así que lo resolveremos de manera secuencial. Es decir, consideremos que estamos en la iteración m y que un modelo pre-entrenado

$$f_{m-1}(x) = \sum_{k=1}^{m-1} T_k(x), \quad (2)$$

donde T_k con $k = 1, \dots, m-1$ son los árboles individualmente entrenados.

Para la iteración actual, ajustaremos un modelo adicional al resolver

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n \ell(y_i, f_{m-1}(x_i) + T(x_i)) , \quad (3)$$

donde $\ell(y, \hat{y})$ es una función de pérdida adecuada.

En el contexto de regresión, si utilizamos pérdida cuadrática podemos agrupar y reescribir

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n \left(r_i^{(m)} - T(x_i) \right)^2 , \quad (4)$$

donde $r_i^{(m)}$ es el residual que enfrentamos al momento de ajustar el m -ésimo modelo.

2.1. Clasificación binaria

Para problemas de clasificación necesitamos resolver en escala logarítmica para después transformar a probabilidades por medio de

$$\mathbb{P}(Y = 1|x) = p(x) = h(f(x)) , \quad (5)$$

donde h es la función logística.

En este caso resolvemos el problema de optimización

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n \ell(y_i, f_{m-1}(x_i) + T(x_i)) , \quad (6)$$

donde ℓ es la **devianza** ($-2 \times \log$ -verosimilitud). Además, usaremos la siguiente codificación de las clases $y \in \{-1, 1\}$. Por lo que la devianza la podemos escribir como

$$\ell(y, \hat{z}) = -[(y+1) \log h(\hat{z}) - (y-1) \log(1-h(\hat{z}))] , \quad (7)$$

la cual se puede escribir (utilizando la expresión de la función h) como

$$\ell(y, \hat{z}) = 2 \log \left(1 + e^{-y\hat{z}} \right) . \quad (8)$$

Nota que en el contexto de regresión el producto $y\hat{z}$ tiene un efecto similar al de un residual en el contexto de regresión. ¿Qué pasa con los signos de la categoría observada y y del *score* latente \hat{z} cuando coinciden y cuando no?

Esto nos permite formular el problema de optimización como

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n 2 \log \left(1 + e^{-y_i(f_{m-1}(x_i) + T(x_i))} \right) , \quad (9)$$

el cual puede ser re-escrito como

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n 2 \log \left(1 + d_i^{(m)} e^{-y_i T(x_i)} \right) . \quad (10)$$

2.1.1. *Para pensar:* ¿Cómo cambia la función objetivo si la función de pérdida es una pérdida exponencial? Es decir, bajo

$$\ell(y, f_m(x)) = \exp(-yf_m(x)) . \quad (11)$$

En este caso (clasificación binaria) vemos que la formulación no es tan fácil que con pérdida cuadrática en el contexto de regresión. Sin embargo, parece ser que los datos reciben una ponderación $d_i^{(m)}$.

3. GRADIENT BOOSTING

Utilizaremos *gradient boosting* para resolver la formulación anterior. Esta técnica se puede aplicar de manera general para cualquier función de pérdida. Lo que necesitamos es resolver

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n \ell(y_i, f_{m-1}(x_i) + T(x_i)) , \quad (12)$$

donde podemos interpretar como *qué tanto tenemos que movernos a partir de f_{m-1} para disminuir la función de pérdida.*

Si nos fijamos en uno de los términos y escribimos

$$\ell(y_i, f_{m-1}(x_i) + z_i) , \quad (13)$$

entonces sabemos que tenemos que modificar de acuerdo a

$$z_i = -\lambda \cdot \frac{\partial \ell}{\partial z_i}(y_i, f_{m-1}(x_i) + z) \Big|_{z=0} , \quad (14)$$

donde tomamos una **longitud de paso** (λ) para estabilizar el proceso de optimización.

La restricción que tenemos es que $z_i = T(x_i)$ para toda observación. De hecho, lo mejor que podríamos tener es que

$$T(x_i) \approx \frac{\partial \ell}{\partial z_i}(y_i, f_{m-1}(x_i)) = g_{i,m} . \quad (15)$$

Así que formulamos un nuevo problema de optimización como

$$\min_{T \in \mathcal{T}} \sum_{i=1}^n (g_{i,m} - T(x_i))^2 , \quad (16)$$

donde $g_{i,m}$ es el gradiente de la función objetivo en la iteración m evaluando en la observación i . Por lo que la actualización sería de la forma

$$f_m(x) = f_{m-1}(x) + \lambda T(x) . \quad (17)$$

3.1. Pseudo-código (árboles de regresión)

El caso de regresión con pérdida cuadrática se presenta a continuación como un ejemplo. Nota que en esta formulación el problema de optimización en términos del gradiente de la función de pérdida es equivalente a ajustar los residuales. Esto nos ayuda a conectar la formulación con resolver un problema predictivo. La tasa de aprendizaje en este caso, previene el sobreajuste pues ayuda a reducir la importancia de la solución a los residuales.

1. Definimos $\hat{f}(x) = 0$, y $r_i = y_i$ para toda observación en el conjunto de entrenamiento.
2. Para cada $m = 1, 2, \dots, M$:

a) Ajustamos un árbol sencillo \hat{f}_m con $d + 1$ nodos terminales para el conjunto $\{(x_i, r_i)\}_{i=1}^n$.

b) Actualizamos el predictor \hat{f} al incluir una versión escalada del nuevo árbol

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}_m(x). \quad (18)$$

c) Actualizamos los residuales

$$r_i \leftarrow r_i - \lambda \hat{f}_m(x). \quad (19)$$

3. Regresamos el modelo

$$\hat{f}(x) = \lambda \sum_{m=1}^M \hat{f}_m(x). \quad (20)$$

3.2. Funciones de pérdida

La selección de función de pérdida parte crucial del algoritmo y se escoge de acuerdo al problema y al objetivo que se quiera resolver. Por ejemplo, en regresión tenemos (por nombrar un par):

1. **Pérdida cuadrática:**

$$\ell(y, z) = \frac{1}{2}(y - z)^2, \quad \frac{\partial \ell}{\partial z} = -(y - z). \quad (21)$$

2. **Pérdida absoluta:**

$$\ell(y, z) = |y - z|, \quad \frac{\partial \ell}{\partial z} = \frac{|y - z|}{y - z}. \quad (22)$$

En el contexto de clasificación podemos utilizar:

1. **Devianza binomial:**

$$\ell(y, z) = -\log(1 + e^{-yz}), \quad \frac{\partial \ell}{\partial z} = I(y = 1) - h(z). \quad (23)$$

2. **Pérdida exponencial:**

$$\ell(y, z) = e^{-yz}, \quad \frac{\partial \ell}{\partial z} = -ye^{-yz}. \quad (24)$$

3.3. Parámetros a optimizar

Los parámetros que usualmente se ajustan con validación cruzada son:

- La tasa de aprendizaje o tamaño de paso λ .
- El número de términos del modelo M .

Más los adicionales de la familia de árboles:

- Profundidad del árbol.
- Número de observaciones en los nodos terminales.

Se pueden incorporar adicionales:

- El número de predictores a utilizar (como en RF).
- Alguna cota de reducción de función objetivo para profundizar el árbol.
- Tamaño de submuestreo.

4. IMPORTANCIA DE VARIABLES

Tanto para bosques aleatorios como modelos de *boosting* basados en árboles se puede estimar cuáles fueron las variables (atributos) que mas contribuyeron en la construcción del modelo.

En cada *nodo interno* de los árboles construidos sabemos que la variable y el punto de corte se escogieron de acuerdo a que maximizaban la **mejora** en dicha región.

Para calcular la importancia de la variable j en el árbol T se suman las contribuciones cada vez que esta variable fue utilizada para generar cortes

$$\mathcal{I}_j^2(T) = \sum_{t=1}^{J-1} \hat{\iota}_t^2 I(v(t) = j), \quad (25)$$

donde $\hat{\iota}_t^2$ es el registro de la mejora en RSS, Gini o **entropía cruzada** por el nodo t cuando este nodo toma el corte utilizando la variable $v(t)$.

La métrica de importancia en un ensamble de modelos considera promediar la mejora en todo el ensamble

$$\mathcal{I}_j^2 = \frac{1}{M} \sum_{m=1}^M \mathcal{I}_j^2(T_m). \quad (26)$$

Se acostumbra registrar importancias relativas de manera que la variable con mayor importancia se le asigna un *score* de 100 puntos y las demás se calculan de manera proporcional.

REFERENCIAS

- [1] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY, 2009. ISBN 978-0-387-84857-0 978-0-387-84858-7. . [1](#)
- [2] R. Schapire. *The Boosting Approach to Machine Learning: An Overview*. MIT press, 2001. [1](#)