Gradient Boosting for Binary Classification

We're going to derive gradient boosting for binary classification. The loss function for binary classification for $y \in \{1, -1\}$ is

$$L(y, f) = \log\big(1 + e^{-y\, f(x)}\big).$$

For reference find the gradient boosting algorithm at the end of these notes.

Formula for f_0

Claim: The best constant for binary classification in this setting

$$f_0 = \log\Big(\frac{1+\bar{y}}{1-\bar{y}}\Big).$$

Proof: The best constant is the α that minimizes the following expression:

$$Q(\alpha) = \sum_{i=1}^{N} \log\big(1 + e^{-y_i\, \alpha}\big). \tag{1}$$

To find the optimal α we take the derivative of (1) with respect to α and set it equal to 0:

$$Q'(\alpha) = -\sum_{i=1}^{N} \frac{y_i}{1 + e^{y_i \alpha}} = 0. \tag{2}$$

Let N^+ be the number of training observations with $y = 1$ and N^- be the number of training observations with $y = -1$. Since y_i is either 1 or -1,

$$\sum_{i=1}^{N} \frac{y_i}{1 + e^{y_i \alpha}} \;=\; \frac{N^+}{1 + e^{\alpha}} \;-\; \frac{N^-}{1 + e^{-\alpha}} \;=\; 0.$$

So

$$\frac{N^+}{1 + e^{\alpha}} \;=\; \frac{N^-}{1 + e^{-\alpha}}.$$

By multiplying the right part of the equation by $\frac{e^{\alpha}}{e^{\alpha}}$ we observe $N^+ = N^-\, e^{\alpha}$. Finally we arrive at

$$\alpha = \log\big(\frac{N^+}{N^-}\big).$$

Note:

$$\log\Big(\frac{1+\bar{y}}{1-\bar{y}}\Big) \;=\; \log\big(\frac{N^+}{N^-}\big),$$

where $\bar{y}$ is the mean of the targets so $\bar{y} = \frac{N^+ - N^-}{N}$, with $N = N^+ + N^-$. Therefore:

$$\frac{1+\bar{y}}{1-\bar{y}} = \frac{N^+}{N^-}.$$

Formula for pseudo-residuals

The general formula for the pseudo-residual is:

$$r_i = -\left[\frac{\partial L(y, f)}{\partial f}\right]_{f=f_{m-1}(x_i),\, y=y_i}$$

First, since

$$L(y, f) = \log\!\left(1 + e^{-y\,f}\right)$$

and

$$\frac{\partial L(y, f)}{\partial f} = -\,\frac{y}{1 + e^{y\,f}}$$

the pseudo-residual r_i is:

$$r_i = \frac{y_i}{1 + e^{\,y_i\,f_{m-1}(x_i)}}.$$

Formula for the best constant per region

The next step is to find the optimal constant per region. That is

$$\beta_j = \arg\min_{\beta} L_{R_j}(\beta)$$

where

$$L_{R_j}(\beta) = \sum_{x_i \in R_j} L\!\left(y_i,\, f_{m-1}(x_i) + \beta\right) = \sum_{x_i \in R_j} \log\!\left(1 + e^{-\,y_i\,\left[f_{m-1}(x_i)+\beta\right]}\right)$$

In the same way as before, we take the derivative of $L_{R_j}(\beta)$ with respect to β and set it equal to 0. Call this function $G(\beta)$:

$$G(\beta) = L'_{R_j}(\beta) = -\sum_{x_i \in R_j} \frac{y_i}{1 + e^{\,y_i\,\left[f_{m-1}(x_i)+\beta\right]}}.$$

The equation $G(\beta) = 0$ does not have a closed form solution, but the optimal β can be approximated by

$$\beta_j \;\approx\; -\,\frac{G(0)}{G'(0)}.$$

Why?

Let's do a Taylor expansion at $\beta = 0$:

$$G(\beta) = G(0) + (\beta - 0)\, G'(0) + O(\beta^2).$$

So if we truncate it into a first-order Taylor expansion, we see that approximately ,

$$G(\beta) \;\approx\; G(0) \;+\; \beta\, G'(0) \;=\; 0.$$

Thus, an approximate solution to $G(\beta) = 0$ and therefore an approximate ideal constant for the region R_j is:

$$\beta_j \;\approx\; -\,\frac{G(0)}{G'(0)}.$$

Calculating β_j:

Note that the numerator:

$$-\,G(0) = \sum_{x_i \in R_j} \frac{y_i}{1 + e^{\,y_i\, f_{m-1}(x_i)}} = \sum_{x_i \in R_j} r_i.$$

Then turning our attention to the denominator we can show that:

$$G'(\beta) = \sum_{x_i \in R_j} \frac{\big(y_i\big)^2\, e^{\,y_i\,[\,f_{m-1}(x_i)+\beta\,]}}{\Big(1 + e^{\,y_i\,[\,f_{m-1}(x_i)+\beta\,]}\Big)^2} = |r_i|(1 - |r_i|).$$

So finally the optimal β for region R_j is:

$$\beta_j \approx -\,\frac{G(0)}{G'(0)} = \frac{\sum_{x_i \in R_j} r_i}{\sum_{x_i \in R_j} |r_i|\,\big(1 - |r_i|\big)}.$$

Gradient Tree Boosting Algorithm

procedure GRADIENT TREE BOOSTING

Initialize $f_0(x) = \arg\min_{\beta} \sum_{i=1}^{N} L\big(y_i, \beta\big)$

for $m = 1$ **to** M **do**

Compute the pointwise negative gradient of the loss function at the current fit:

$$r_i = -\Big[\frac{\partial L(y, f)}{\partial f}\Big]_{f = f_{m-1}(x_i), y = y_i} \qquad \text{for } i = 1, 2, \ldots, n$$

Approximate the negative gradient by fitting a regression tree to the targets r_i, giving terminal regions R_{jm}, $j = 1, \ldots, J_m$.

Compute new predictions for every terminal node. For $j = 1, \ldots, J_m$ compute

$$\beta_{jm} = \arg\min_{\beta} \sum_{x_i \in R_{jm}} L\big(y_i, f_{m-1}(x_i) + \beta\big)$$

Update $f_m(x) = f_{m-1}(x) \ + \ \sum_{j=1}^{J_m} \beta_{jm}\, \mathbf{1}\big(x \in R_{jm}\big)$

end for loop

return $f(x) = f_M(x)$