DETRA NOTE

DETRA NOTE 2025-1

BOOSTING ON THE RESPONSES WITH TWEEDIE AND BINOMIAL LOSS FUNCTIONS

Julien Trufin

Detralytics

DISCLAIMER

The content of the Detra Notes for a pedagogical use only. Each business case is so specific that a careful analysis of the situation is needed before implementing a possible solution. Therefore, Detralytics does not accept any liability for any commercial use of the present document. Of course, the entire team remain available if the techniques presented in this Detra Note required your attention.

Detralytics
Avenue du Boulevard 21 Box 5 - 1210 Bruxelles (Belgium)
1-7 Cour Valmy, Le Belvédère - 92800 Puteaux (France)
www.detralytics.com - info@detralytics.eu

ABSTRACT

Boosting emerged from the field of machine learning and became rapidly popular among insurance analysts. The Tweedie and Binomial distributions are the most commonly used in insurance for regression analysis. Hainaut et al. (2022) showed that boosting can be conducted directly on the response under Tweedie loss function and log-link, by adapting the weights at each iteration step. In this note, we recall the results of Hainaut et al. (2022) and we supplement them with an easy probabilistic interpretation to the boosting procedure. Next, we draw a parallel between these results and those established by Hastie et al. (2009) for the Bernoulli loss function and logit-link: Hastie et al. (2009) highlighted that, as an approximation, boosting can also be performed directly with responses under Bernoulli loss function and logit-link. Interestingly, we show that this observation can actually been extended to the Binomial case.

Keywords: Boosting; Tweedie; Binomial loss function.

1 Introduction

Boosting emerged from the field of machine learning and became rapidly popular among insurance analysts. Broadly speaking, boosting is an iterative fitting procedure using weak, or base learners. In a regression context, weak learners have rigid parametric forms and cannot accurately adapt to the data under consideration. In each iteration, boosting fits a weak learner that improves the fit of the overall model such that the ensemble arrives at an accurate prediction. We refer the readers to Denuit et al. (2019b, 2020) for an extensive treatment of boosting in the context of insurance, with applications to tree-based methods and neural networks.

Boosting requires that the analyst first decides which metric should be optimized for estimating the score. In the boosting terminology, this metric is usually referred to as the “loss function”. To ease numerical aspects, boosting is often applied on gradients of the loss function, that is, on the gradients of the deviance function in insurance applications. However, Hainaut et al. (2022) showed that there is often no need to boost gradients in insurance applications. As proved in their Proposition 3.1, boosting can easily be performed directly with responses under Tweedie loss function and log-link.

In this note, we recall the results of Hainaut et al. (2022) and we provide an easy probabilistic interpretation to the boosting procedure under Tweedie loss function and log-link. Then, we also recall that boosting can also be performed directly with responses under Bernoulli loss function and logit-link and we extend this observation to the Binomial loss function.

2 Insurance pricing and boosting

2.1 Insurance pricing

In actuarial pricing, the aim is to evaluate the pure premium as accurately as possible. The target is the conditional expectation $\mu(\boldsymbol{X}) = \mathrm{E}[Y|\boldsymbol{X}]$ of the response Y (claim number or claim amount for instance) given the available information summarized in a vector $\boldsymbol{X}$ of features $X_1, X_2, \ldots, X_p$. The function $\boldsymbol{x} \mapsto \mu(\boldsymbol{x}) = \mathrm{E}[Y|\boldsymbol{X} = \boldsymbol{x}]$ is unknown to the actuary and is approximated by a working predictor $\boldsymbol{x} \mapsto \widehat{\mu}(\boldsymbol{x})$ entering premium calculation.

Lack of accuracy for $\widehat{\mu}(\boldsymbol{x})$ is defined by the generalization error

$$Err(\widehat{\mu}) = \mathrm{E}\left[L(Y, \widehat{\mu}(\boldsymbol{X}))\right],$$

where $L(.,.)$ is the loss function measuring the discrepancy between its two arguments and the expected value is over the joint distribution of $(Y, \boldsymbol{X})$. The goal is to find a function $\boldsymbol{x} \mapsto \widehat{\mu}(\boldsymbol{x})$ of the features minimizing the generalization error.

2.2 Boosting

Ensemble techniques assume structural models of the form

$$\mu(\boldsymbol{x}) = g^{-1}\left(\mathrm{score}(\boldsymbol{x})\right) = g^{-1}\left(\sum_{m=1}^{M} T(\boldsymbol{x};\mathbf{a}_m)\right), \tag{2.1}$$

where g is the link function (assumed to be monotone and differentiable) and $T(\boldsymbol{x};\mathbf{a}_m)$, $m = 1, 2, \ldots, M$, are usually simple functions of the features $\boldsymbol{x}$, characterized by parameters $\mathbf{a}_m$. In (2.1), the score is the function of features $\boldsymbol{x}$ mapped to $\mu(\boldsymbol{x})$ by the inverse of the link function g.

Let

$$\mathcal{D} = \left\{(\nu_1, y_1, \boldsymbol{x}_1), (\nu_2, y_2, \boldsymbol{x}_2), \ldots, (\nu_n, y_n, \boldsymbol{x}_n)\right\},$$

be the set of observations used to fit the model $\widehat{\mu}$, called training set, where ν_i denotes the weight of observation i. Estimating a score of the form

$$\mathrm{score}(\boldsymbol{x}) = \sum_{m=1}^{M} T(\boldsymbol{x};\mathbf{a}_m),$$

by minimizing the corresponding training sample estimate of the generalized error

$$\min_{\{\mathbf{a}_m\}_1^M} \sum_{i=1}^{n} \nu_i L\left(y_i, g^{-1}\left(\sum_{m=1}^{M} T(\boldsymbol{x}_i;\mathbf{a}_m)\right)\right) \tag{2.2}$$

is in general infeasible. It requires computationally intensive numerical optimization techniques. One way to overcome this problem is to approximate the solution to (2.2) by using a greedy forward stagewise approach, also called boosting.

Forward stagewise additive modeling consists in sequentially fitting a single function and adding it to the expansion of prior fitted terms. Each fitted term is not readjusted as new terms are added into the expansion, contrarily to a stepwise approach where previous terms are each time readjusted when a new one is added. Specifically, we start by computing

$$\widehat{\mathbf{a}}_1 = \underset{\mathbf{a}_1}{\mathrm{argmin}} \sum_{i=1}^{n} \nu_i L\left(y_i, g^{-1}\left(\widehat{\mathrm{score}}_0(\boldsymbol{x}_i) + T(\boldsymbol{x}_i;\mathbf{a}_1)\right)\right),$$

where $\widehat{\mathrm{score}}_0(\boldsymbol{x})$ is an initial guess (for instance, just an intercept). Then, at each iteration $m \geq 2$, we solve the subproblem

$$\widehat{\mathbf{a}}_m = \underset{\mathbf{a}_m}{\mathrm{argmin}} \sum_{i=1}^{n} \nu_i L\left(y_i, g^{-1}\left(\widehat{\mathrm{score}}_{m-1}(\boldsymbol{x}_i) + T(\boldsymbol{x}_i;\mathbf{a}_m)\right)\right), \tag{2.3}$$

with

$$\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i) = \widehat{\text{score}}_{m-2}(\boldsymbol{x}_i) + T(\boldsymbol{x}_i; \widehat{\mathbf{a}}_{m-1}).$$

Boosting is thus an iterative method based on the idea that combining many simple functions should result in a powerful one. In a boosting context, the simple functions $T(\boldsymbol{x}; \mathbf{a}_m)$ are called weak learners or base learners.

3 Tweedie loss functions and log-link

3.1 Tweedie losses

The Tweedie family of distributions regroups the members of the Exponential Dispersion family having power variance functions $V(\mu) = \mu^\xi$ for some ξ. From e.g. Denuit et al. (2019a, Table 4.7), the Tweedie deviance loss function is given by

$$L(Y, \widehat{\mu}(\boldsymbol{X})) = \begin{cases} (Y - \widehat{\mu}(\boldsymbol{X}))^2 & \text{if } \xi = 0, \\ 2\left(Y \ln \frac{Y}{\widehat{\mu}(\boldsymbol{X})} - (Y - \widehat{\mu}(\boldsymbol{X}))\right) & \text{if } \xi = 1, \\ 2\left(-\ln \frac{Y}{\widehat{\mu}(\boldsymbol{X})} + \frac{Y}{\widehat{\mu}(\boldsymbol{X})} - 1\right) & \text{if } \xi = 2, \\ 2\left(\frac{\max\{Y,0\}^{2-\xi}}{(1-\xi)(2-\xi)} - \frac{Y\widehat{\mu}(\boldsymbol{X})^{1-\xi}}{1-\xi} + \frac{\widehat{\mu}(\boldsymbol{X})^{2-\xi}}{2-\xi}\right) & \text{otherwise } (\xi > 0). \end{cases} \tag{3.1}$$

For $\xi = 0$, we recover the L^2 loss function whereas $\xi = 1$ and 2 correspond to the Poisson and Gamma deviance functions, respectively.

3.2 Log-link function

3.2.1 Result of Hainaut et al. (2022)

Hainaut et al. (2022) showed in their Proposition 3.1 that under Tweedie loss function and log-link, the subproblem (2.3) can be rewritten as

$$\widehat{\mathbf{a}}_m = \underset{\mathbf{a}_m}{\operatorname{argmin}} \sum_{i=1}^{n} \nu_{i,m} L\left(\tilde{r}_{i,m}, \exp\left(T(\boldsymbol{x}_i; \mathbf{a}_m)\right)\right),$$

where $\nu_{i,m} = \nu_i \exp(\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i))^{2-\xi}$ and $\tilde{r}_{i,m} = \frac{y_i}{\exp(\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i))}$. In words, the mth iteration of the boosting procedure reduces to build a single weak learner on the working training set

$$\mathcal{D}^{(m)} = \{(\nu_{i,m}, \tilde{r}_{i,m}, \boldsymbol{x}_i), i = 1, \ldots, n\}$$

using the Tweedie deviance loss and the log-link function. The weights are each time updated together with the responses.

3.2.2 Probabilistic interpretation

There is an easy probabilistic interpretation to this boosting algorithm. Recall that if Y obeys the Tweedie distribution with mean μ, power parameter ξ and weight ν then for any positive constant c, cY is Tweedie with mean $c\mu$, the same power parameter and modified weight $\nu(c)^{\xi-2}$. One says that the Tweedie distributions are closed under this type of scale transformation. See e.g. Denuit et al. (2019a) for a proof. Since the essence of boosting consists in treating $\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i)$ as a constant to estimate $\mathbf{a}_m$, this means that we can equivalently work with response $\tilde{r}_{i,m}$ obeying the Tweedie distribution with adapted weight $\nu_{i,m}$ to perform that estimation. This is a direct application of the result recalled earlier with $c = 1/\exp(\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i))$. This process can even be performed in an iterative way, by dividing the response $\tilde{r}_{i,m}$ with $\exp(T(\boldsymbol{x}_i; \mathbf{a}_m))$ and multiplying $\nu_{i,m}$ with $\exp((2-\xi)T(\boldsymbol{x}_i; \mathbf{a}_m))$ at each step.

Notice that in the Gaussian case ($\xi = 0$), the boosting procedure described here differs from the classical gradient boosting algorithm with L^2 loss, which uses identity link and raw residuals (current estimate subtracted from the response) whereas here, we work with ratios under log-link.

At each step of the boosting algorithm, the response is thus Tweedie distributed so that the loss function selected for the original responses y_i (and weights ν_i) is still the right choice at iteration m for new responses $\tilde{r}_{i,m}$ (and weights $\nu_{i,m}$). This is a direct consequence of the closure property of Tweedie distributed responses Y under scale transformation of the type cY.

4 Binomial loss function and logit-link

4.1 Binomial loss

Boosting can thus be conducted directly on the response under Tweedie loss function and log-link, by adapting the weights at each iteration step.

Let us now consider that the response Y obeys the Binomial distribution with $Y \in \{0, 1, \ldots, k\}$, i.e.

$$\mathrm{P}[Y = y|\boldsymbol{X} = \boldsymbol{x}] = \binom{k}{y} q(\boldsymbol{x})^y (1 - q(\boldsymbol{x}))^{k-y}, \;\; y = 0, 1, \ldots, k,$$

where the Binomial coefficient is defined as

$$\binom{k}{y} = \frac{k!}{y!(k-y)!}.$$

From Denuit et al. (2019a, Table 4.7), the Binomial deviance loss function is given by

$$L(Y, \widehat{\mu}(\boldsymbol{X})) = 2\left(Y \ln \frac{Y}{\widehat{\mu}(\boldsymbol{X})} + (k - Y) \ln \frac{k - Y}{k - \widehat{\mu}(\boldsymbol{X})}\right) \tag{4.1}$$

where $\widehat{\mu}(\boldsymbol{X}) = k\widehat{q}(\boldsymbol{X})$. For model estimation, we can work with

$$\begin{aligned} L(Y, \widehat{\mu}(\boldsymbol{X})) &= -Y \ln \widehat{\mu}(\boldsymbol{X}) - (k - Y) \ln(k - \widehat{\mu}(\boldsymbol{X})) \\ &= -Y \ln \widehat{q}(\boldsymbol{X}) - (k - Y) \ln(1 - \widehat{q}(\boldsymbol{X})) \qquad (4.2) \end{aligned}$$

since the terms in (4.1) that not depend on $\widehat{\mu}$ are irrelevant for model estimation.

4.2 Logit-link function

4.2.1 Particular case: $k = 1$

Let us first consider the case where $k = 1$, i.e. Y follows a Bernoulli distribution. In actuarial science, such situations arise when Y represents for instance the occurrence of at least one claim for the policyholder or the detection of a fraudulent case over the observation period. Considering the logit link function $g(x) = \ln\left(\frac{x}{1-x}\right)$ in (4.2), we get

$$\begin{aligned} L\left(Y, g^{-1}(\widehat{\text{score}}(\boldsymbol{X}))\right) &= -Y \ln g^{-1}(\widehat{\text{score}}(\boldsymbol{X})) - (1 - Y) \ln(1 - g^{-1}(\widehat{\text{score}}(\boldsymbol{X}))) \\ &= -Y \ln\left(\frac{1}{1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}}\right) - (1 - Y) \ln\left(1 - \frac{1}{1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}}\right) \\ &= Y \ln\left(1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}\right) - (1 - Y) \ln\left(\frac{e^{-\widehat{\text{score}}(\boldsymbol{X})}}{1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}}\right) \\ &= Y \ln\left(1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}\right) + (1 - Y)\widehat{\text{score}}(\boldsymbol{X}) + (1 - Y) \ln\left(1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}\right) \\ &= (1 - Y)\widehat{\text{score}}(\boldsymbol{X}) + \ln\left(1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}\right) \\ &= \ln\left(e^{(1-Y)\widehat{\text{score}}(\boldsymbol{X})}\right) + \ln\left(1 + e^{-\widehat{\text{score}}(\boldsymbol{X})}\right) \\ &= \ln\left(e^{(1-Y)\widehat{\text{score}}(\boldsymbol{X})} + e^{-Y\widehat{\text{score}}(\boldsymbol{X})}\right) \\ &= \ln\left(1 + e^{-\widehat{\text{score}}(\boldsymbol{X})(2Y-1)}\right). \end{aligned}$$

This latter expression is equivalent to formula (10.18) in Hastie et al. (2009) where the authors rather work with the response $Y' = 2Y - 1 \in \{-1, 1\}$.

The loss function $L\left(Y', \text{score}(\boldsymbol{X})\right) = \ln\left(1 + e^{-\text{score}(\boldsymbol{X})Y'}\right)$ contains the exponential loss $e^{-\text{score}(\boldsymbol{X})Y'}$. Hastie et al. (2009) point out that both loss functions $L\left(Y', \text{score}(\boldsymbol{X})\right)$

and $e^{-\text{score}(\boldsymbol{X})Y'}$ lead to the same solution at the population level. This constitutes a motivation to replace the subproblem (2.3) for Bernoulli loss with logit-link with

$$\widehat{\mathbf{a}}_m = \underset{\mathbf{a}_m}{\operatorname{argmin}} \sum_{i=1}^{n} \nu_i \exp\left(-(2y_i - 1)\left(\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i) + T(\boldsymbol{x}_i; \mathbf{a}_m)\right)\right),$$

which in turn, can be rewritten as

$$\widehat{\mathbf{a}}_m = \underset{\mathbf{a}_m}{\operatorname{argmin}} \sum_{i=1}^{n} \nu_{i,m} \exp\left(-(2y_i - 1)T(\boldsymbol{x}_i; \mathbf{a}_m)\right), \tag{4.3}$$

where $\nu_{i,m} = \nu_i \exp(-(2y_i - 1)\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i))$.

As stated in Hastie et al. (2009), it is worth noticing that although both the exponential loss and Binomial deviance yield the same solution when applied to the population joint distribution, the same is not true for finite data sets.

As for Tweedie losses and log-link, the mth iteration (4.3) of the boosting procedure with the exponential loss can be seen as building a single weak learner on a working training set, here given by

$$\mathcal{D}^{(m)} = \{(\nu_{i,m}, y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}.$$

The weights are each time updated while the responses remain unchanged. The main difference with the Tweedie case is that the weights $\nu_{i,m}$ depend on the response y_i. Observations with $y_i = 0$ have their weights ν_i scaled by a factor $\exp(\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i))$, so that the higher the current scores $\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i)$, the greater the weights $\nu_{i,m}$ at the mth iteration. On the contrary, observations with $y_i = 1$ have their weights ν_i scaled by a factor $\exp(-\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i))$, which means that the higher the current scores $\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i)$, the lower the weights $\nu_{i,m}$ at the mth iteration. Thus, at each iteration, the boosting procedure puts more weights to observations that are not well fitted by the estimates obtained so far. As mentioned in Hastie et al. (2009), the forward stagewise additive modeling using the exponential loss function is actually equivalent to AdaBoost.M1 algorithm (see Algorithm 10.1 in Hastie et al., 2009).

4.2.2 General case

Consider now the cases where $k \geq 1$. The loss associated with the observation $(\nu_i, y_i, \boldsymbol{x}_i)$ is given by

$$L(y_i, \widehat{\mu}(\boldsymbol{x}_i)) = -y_i \ln \widehat{q}(\boldsymbol{x}_i) - (k - y_i) \ln(1 - \widehat{q}(\boldsymbol{x}_i)),$$

which can be rewritten as

$$L(y_i, \widehat{\mu}(\boldsymbol{x}_i)) = \sum_{j=1}^{k} -y_{ij} \ln \widehat{q}(\boldsymbol{x}_i) - (1 - y_{ij}) \ln(1 - \widehat{q}(\boldsymbol{x}_i))$$

with

$$y_{ij} = \begin{cases} 0 & \text{for } j = 1, \ldots, k - y_i, \\ 1 & \text{for } j = k - y_i + 1, \ldots, k. \end{cases}$$

The loss $L(y_i, \widehat{\mu}(\boldsymbol{x}_i))$ can thus be decomposed as the sum of k terms where the jth term $(j = 1, \ldots, k)$ is the Bernoulli loss associated with the pseudo-observation $(\nu_i, y_{ij}, \boldsymbol{x}_i)$.

Therefore, one sees that it is equivalent to work with the Binomial loss and the observations $\{(\nu_i, y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$ or to work with the Bernoulli loss and the pseudo-observations $\{(\nu_i, y_{ij}, \boldsymbol{x}_i), i = 1, \ldots, n, j = 1, \ldots, k\}$. The Binomial case can therefore be reduced to the Bernoulli case studied in the previous section.

This motivates the fact that the mth iteration of the boosting procedure (2.3) with the logit-link function can be approximated as building a single weak learner with the exponential loss on the working training set

$$\mathcal{D}^{(m)} = \{(\nu_{ij,m}, y_{ij}, \boldsymbol{x}_i), i = 1, \ldots, n, j = 1, \ldots, k\},$$

where $\nu_{ij,m} = \nu_i \exp(-(2y_{ij} - 1)\widehat{\text{score}}_{m-1}(\boldsymbol{x}_i))$.

5 Conclusion

Boosting is a powerful machine learning technique used to improve the accuracy of predictive models by combining the strengths of multiple weak learners. Tweedie and Binomial loss functions are the most commonly used loss functions for score estimation in insurance applications. While boosting is often applied on gradients of the selected loss function, we recall that there is no need to boost gradient under Tweedie loss function and log-link, as shown in Hainaut et al. (2022). Moreover, we motivate the fact that boosting with Binomial loss function and logit-link can be reduced to the Bernoulli case and hence be approximated by a boosting with the exponential loss which does not require the computation of gradients of the loss function.

References

- Denuit, M., Hainaut, D., Trufin, J. (2019a). Effective Statistical Learning Methods for Actuaries I: GLM and Extensions. Springer Actuarial Lecture Notes Series.

- Denuit, M., Hainaut, D., Trufin, J. (2019b). Effective Statistical Learning Methods for Actuaries III: Neural Networks and Extensions. Springer Actuarial Lecture Notes Series.

- Denuit, M., Hainaut, D., Trufin, J. (2020). Effective Statistical Learning Methods for Actuaries II: Tree-based Methods and Extensions. Springer Actuarial Lecture Notes Series.

- Hainaut, D., Trufin, J., Denuit, M. (2022). Response versus gradient boosting trees, GLMs and neural networks under Tweedie loss and log-link. Scandinavian Actuarial Journal 2022(10), 841-866.

- Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Stanford, CA: Stanford University.

Detralytics

People drive actuarial innovation

www.detralytics.com - info@detralytics.eu