

11.1 Suppose you change the shrinkage (or learning rate) parameter from $\lambda = 0.1$ to $\lambda = 0.01$, holding other parameters constant. What is the likely effect on the number of trees (or iterations) that optimizes your criterion?

11.1 Updating boosting trees: $f_m = f_{m-1} + \lambda B_m b(x; \gamma_m)$

With $\lambda = 0.1$, each tree moves the model 10 times more than with $\lambda = 0.01$.

Smaller learning rate allows for smaller jumps towards the optimum

To reach the same level, you need more trees (Substantially increase)

11.2 The MART algorithm for boosting includes the following step: $f_m = f_{m-1} +$ fitted values. Figure 11.7 displays the fitted values of a logit boost algorithm for $m = 3$ trees. The initial constant was estimated as $f_0 = -0.3$.

- (a) John is aged 23 and has popularity 0.5. Suppose the trees were generated without shrinkage (i.e., $shrinkage = 1$). What is the predicted probability $P(y = 1)$ for John based on the three boosting trees?
- (b) Maria is aged 53 and has popularity 1.3. Suppose the trees were generated without shrinkage (i.e., $shrinkage = 1$). What is the predicted probability $P(y = 1)$ for Maria based on the three boosting trees?
- (c) Suppose the trees were generated with $shrinkage = 0.1$. What is the predicted probability now for John and Maria after the first tree? Why is it not possible to compute the predicted probability after the third tree?

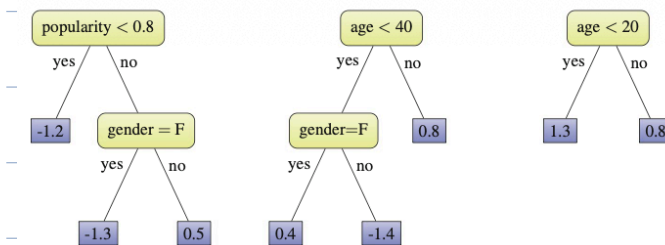


Fig. 11.7 Fitted values of a logit boost algorithm for $m = 3$ trees

11.2

(a) $m = 3$, $f_0 = -0.3$

1st tree $\rightarrow -1.2$

2nd tree $\rightarrow -1.4$

3rd tree $\rightarrow 0.8$

$$f_3 = f_1 + f_2 + f_3$$

$$f(x) = \log\left(\frac{p}{1+p}\right)$$

$$p = \frac{e^{f(x)}}{1 + e^{f(x)}}$$

$$f_3 = 0.8 - 1.2 - 1.4 - 0.3 = -2.1$$

$$p_3 = \frac{e^{-2.1}}{1 + e^{-2.1}} = 0.109$$

(b) 1st tree $\rightarrow -1.3$

$$f_3 = -1.3 + 0.8 + 0.8 - 0.3 = 0$$

2nd tree $\rightarrow 0.8$

$$p_3 = \frac{e^0}{1 + e^0} = \frac{1}{2} = 0.5$$

3rd tree $\rightarrow 0.8$

(c) $shrinkage = 0.1$

$$\text{Recall } f_m = f_{m-1} + \lambda B_m b(x; \gamma_m)$$

$$\text{John} = -0.3 \times (0.1 \times 1.2) = -0.42$$

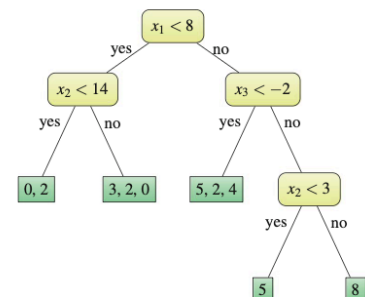
$$p_1 = \frac{e^{-0.42}}{1 + e^{-0.42}} = 0.397$$

$$\text{Maria} = -0.3 \times (0.1 \times -1.3) = -0.47$$

$$p_1 = \frac{e^{-0.47}}{1 + e^{-0.47}} = 0.394$$

11.3 Consider the regression tree in Fig. 11.8. The tree displays which variable is used for splitting. The tree does not show the split value or which branch is the "yes" or "no" branch because it is irrelevant for the question. The terminal nodes display the y-values of all training observations that filter into that leaf. Because there are a total of ten numbers in the leaves, there are a total of ten observations. There are four x-variables: x_1, x_2, x_3, x_4 . One of the x-variables, x_4 , was not used in any split. A variable's influence is based on the total reduction in sums of squares due to that variable.

Fig. 11.8 A regression tree



Compute the influence of the four x-variables based on this tree such that the sum of the influences is 100%. You can compute this entirely by hand or use software for intermediate steps and continue by hand.

11.3 Total 10 numbers in the leaf (10 observations)

$$x_1, x_2, x_3, x_4$$

Influence of the variables.

$$\rightarrow \text{Full of all} = \frac{31}{10}$$

$$\text{All SS} = 54.9$$

$$\boxed{X=1} \quad \hat{c}_m = \frac{7}{5} \quad \text{RSS} = 2\left(0 - \frac{7}{5}\right)^2 + (2 - \frac{7}{5})^2 + (3 - \frac{7}{5})^2 + (2 - \frac{7}{5})^2 = 7.2$$

$$\hat{c}_m = \frac{29}{5} \quad \text{Hence } (5 - \frac{29}{5})^2 + (2 - \frac{29}{5})^2 \dots = 18.8$$

$$\text{Total reduction} = 54.9 - 7.2 - 18.8 = 28.9$$

$$\boxed{X=2} \quad \text{at } x_2 < 1.9 \quad \text{Total RSS} \Rightarrow \bar{y} = \frac{7}{5} \quad \text{SS} = 7.2$$

$$\boxed{LS} \quad \hat{c}_m = 1, \quad \text{RSS} = (0 - 1)^2 + (2 - 1)^2 = 2$$

$$\boxed{RS} \quad \hat{c}_m = \frac{5}{3}, \quad \text{RSS} = (3 - \frac{5}{3})^2 + (2 - \frac{5}{3})^2 + (0 - \frac{5}{3})^2 = 9.667$$

$$7.2 - 2 - 9.667 = 0.533$$

Repeat and x_1 is the most influential one

11.4 Multi-class problems often require the softmax function to convert the scores for each class $k = 1, \dots, K$ to probabilities for each class:

$$\pi_k(x) = e^{f_k(x)} / \sum_{l=1}^K e^{f_l(x)}$$

11.4

$$\textcircled{a} \quad \sum_{k=1}^K \pi_k(x) = \frac{\sum_{k=1}^K e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}} = \frac{\sum_{k=1}^K e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}}$$

Multinomial regression, multi-class boosting, multi-class SVM (Platt's method), and multi-class neural networks all use the softmax function.

(a) Show that the probabilities are all positive and sum to 1.

(b) For a sentiment analysis, reviews of a business are classified into positive, negative, or neutral. Using multinomial boosting, the final score f_{km} after fitting m trees for text x_0 is

$$f_{\text{positive},m}(x_0) = -1, \quad f_{\text{negative},m}(x_0) = +1, \quad f_{\text{neutral},m}(x_0) = +2.$$

Compute the probabilities $\pi_1(x_0)$, $\pi_2(x_0)$, $\pi_3(x_0)$. Is text x_0 classified as positive, negative, or neutral?

① Multinomial boosting

$$f_{k0} = 0.$$

$$\pi_1(x) = \frac{e^{-1}}{e^{-1} + e^1 + e^2} = \frac{e^{-1}}{10.475} = 0.035$$

$$\pi_2(x) = \frac{e^1}{10.475} = 0.259$$

$$\pi_3(x) = \frac{e^2}{10.475} = 0.705$$

$$\text{Compute residuals} \rightarrow p_1 = -1 - 0.035 = -1.035$$

$$p_2 = 1 - 0.259 = \dots$$

Text is likely to be classified as neutral

11.5 XGBoost has a regularization parameter, ν . Like regular boosting, XGBoost also has a learning rate (or shrinkage parameter), λ . How does the effect of these two parameters on the update to f_{m-1} differ?

11.5

Regularization parameter, ν

XGBoost learning rate

$$\mathcal{E}(l(y, f_{(m-1)}) + \mathcal{E}(\hat{\gamma}_{imgi} + 0.5 \hat{\epsilon}_{hi}^2 + 0.5 \nu \hat{\epsilon}_{ij}^2 + \lambda \tau$$

- ν : Affects the size of each leaf's fitted value, larger ν shrinks γ to near 0, and

each tree's prediction is smaller. Penalizing large weights directly. Does not affect the tree structure

$$- \lambda: \text{From } f_m = f_{m-1} + \lambda \cdot r_m$$

Scales down the tree's contribution before adding it to the model

Residuals from update step. Contribution to the entire tree

Scales the whole tree down

11.6 Consider a regression XGBoost model for the data in Table 11.2. Use a learning rate of 0.5, and $\nu = 0.5$. Assume the maximal tree depth is 2.

(a) Initialize $f_0 = 0.5$. Build the first tree for XGBoost. You need to decide on the splits; fitted values are computed below.

11.6

$$\lambda = 0.5, \nu = 0.5 \text{ Max. depth} = 2$$

$$\textcircled{a} f_m = f_{m-1} - \lambda \cdot r_i$$

$$f_0 = 0.5$$

Similarity_{root} = measures how homogeneous the residuals are

in a node. Low similarity = impure node = bad

$$r_{im} = \frac{\sum_{x \in R} (y_i - f_{m-1}(x))}{n_{im} + \nu}$$

$$= \frac{(7.5 + 2.5 + 5.5 + 6.5)^2}{9.5} = 107.56$$

234

11 Boosting

Table 11.2 Toy data for XGBoost exercise

y	8	3	6	7
x	1	2	6	6

(b) Prune the first tree using $\alpha = 0.5$.

(c) Compute fitted values $\hat{y}_{im}$ for the pruned first tree.

(d) Compute f_1 .

(e) Why would $f_0 = \bar{y}$ in this case have been a better starting value than $f_0 = 0.5$?

Split ① 1.5

$$S_R = \frac{7.5^2}{1.5} = 37.5$$

$$S_L = \frac{(2.5 + 5.5 + 6.5)^2}{3.5} = \frac{14.5^2}{3.5} = 60.07$$

$$\text{Gain} = 60.07 + 37.5 - 107.56 = -10.0$$

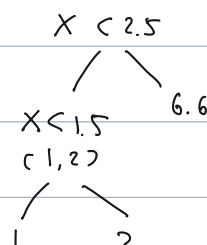
1st tree

Split ② 2.5

$$S_R = \frac{(7.5 + 2.5)^2}{2.5} = \frac{100}{2.5} = 40$$

$$S_L = \frac{(5.5 + 7.5)^2}{2.5} = \frac{144}{2.5} = 57.6$$

$$\text{Gain} = 57.6 + 40 - 107.56 = -9.96$$



2.5 is better

1.2 split ① $S_{root} = \frac{(7.5 + 2.5)^2}{2.5} = \frac{100}{2.5} = 40$

$$S_L = \frac{7.5^2}{1.5} = 37.5$$

$$S_R = \frac{2.5^2}{1.5} = 4.17$$

$$\text{Gain} = 4.17 + 37.5 - 40 = 1.67$$

6.6 split ①

= No need since identical

$$\gamma_{T_1} = \frac{\text{sum of resids}}{\# \text{ of resids} + 1} = \frac{7.5}{1.5} = 5$$

$$\gamma_{T_2} = \frac{2.5}{1.5} = 1.667$$

$$\gamma_{T_3} = \frac{(5.5 + 6.5)}{2.5} = 4.8$$

⑥ Prune the first tree using $\lambda = 0.5$

Remove a split if gain $< \lambda$

Prune the root split as it had negative values

$$\textcircled{c} \quad r = \frac{7.5 + 2.5 + 5.5 + 6.5}{4.5} = 4.89$$

$$\textcircled{d} \quad f_1 = f_{m-1} - \lambda \cdot r_{\text{im}}$$

$$= f_0 - (0.5)(4.89) = 2.94$$

⑦ $\bar{y}$ would be better because residuals would be smaller, needs less boosting rounds