Consider some data $\{(x_i, y_i)\}_{i=1}^n$ and a differentiable loss function $\mathcal{L}(y, F(x))$ and a multiclass classification problem which should be solved by a gradient boosting algorithm. Let therefore be the cross-entropy loss function defined by the class probabilities gained from the softmax function so that :

$$softmax = \frac{e^{y_i}}{\sum_{k=1}^N e^{y_k}} \tag{1}$$

$$\mathcal{L}(y_i, \hat{y}_i) = -\sum y_i \log \hat{y}_i \tag{2}$$

where y_i defines the the relative frequencies of each class in our target variable y.

In this case we end up with the partial derivatives for the softmax function

$$D_j \, \mathrm{softmax}_i = \frac{\delta softmax_i}{\delta y_j} = \begin{bmatrix} D_1 \, \mathrm{softmax}_1 & \times & D_N \, \mathrm{softmax}_1 \\ \vdots & \ddots & \vdots \\ D_1 \, \mathrm{softmax}_N & \times & D_N \, \mathrm{softmax}_N \end{bmatrix} \tag{3}$$

$$D_j \; \mathrm{softmax} \;_i = \left\{ \begin{array}{ll} \mathrm{softmax}_i - \mathrm{softmax}_j^2 & i = j \\ -\,\mathrm{softmax}_j \times \mathrm{softmax}_i & i \neq j \end{array} \right\} \tag{4}$$

And the derivative of the cross-entropy loss function w.r.t. $F(x)$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = \left(-\sum_i y_i \log \hat{y}_i \right) = -\frac{y_i}{\hat{y}_i} \tag{5}$$

Combining both gradients leads to the gradient of the loss function

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \hat{y}} &= -\frac{y_i}{\hat{y}_i} \hat{y}_i \left(1 - \hat{y}_i\right) + \sum_{t \neq i} -\frac{y_t}{\hat{y}_t} \left(-\hat{y}_t \hat{y}_i\right) \\ &= -y_i + y_i \hat{y}_i + \sum_{t \neq i} y_t \hat{y}_i \\ &= -y_i + \sum_t y_t \hat{y}_i \\ &= \hat{y}_i \underbrace{\sum_t y_t}_{=1} - y_i \\ &= \hat{y}_i - y_i \end{aligned} \tag{6}$$

In this sense our initial model $F_0(x)$ should be :

$$F_0(x) = \frac{e^{y_i}}{\sum_{k=1}^N e^{y_k}} \tag{7}$$

One obtains the intial residuals $r_{i0} = y_i - F_0(x)$ which are then used to fit a classification tree with R_{im} terminal nodes.

The pseudo residuals are obtained through

$$\begin{aligned} r_{im} &= -\left[\frac{\partial \mathcal{L}\left(y_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n \\ &= -\sum_{i=1}^{N}(\hat{y}_i - y_i) \\ &= \sum_{i=1}^{N}(y_i - \hat{y}_i) \end{aligned} \tag{8}$$

The output values for each terminal node can be derived using a second-order Taylor approximation so that,

$$\begin{aligned} \gamma_{lm} &= \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{lm}} L\left(y_i, F_{m-1}\left(x_i\right) + \gamma\right) \\ &\approx -\left[\mathcal{L}\left(y_i, F\left(x_i\right)\right) + \frac{\partial \mathcal{L}\left(y_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)} + \frac{\partial \mathcal{L}\left(y_i, F\left(x_i\right)\right)}{\partial^2 F\left(x_i\right)}\right] \\ &= y_i log(\hat{y}_i) + \sum_{i=1}^{N}(y_i - \hat{y}_i) - 1 \end{aligned} \tag{9}$$

In the end our new predictions should be :

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{m} \gamma_{jm} I\left(x \in R_{jm}\right) \tag{10}$$