Insights, resources, and application of : ‘Gradient Boosting and Ensemble Learners’

BY YALIN LIAO

Boosting approximates the target model $F^*(x)$ by an aditive expansion of the form

$$F_M(x) = w_0 h(x|\theta_0) + w_2 h(x|\theta_1) + \cdots + w_M h(x|\theta_M)$$

where each base model $h_m(x) \in \mathbb{H}$ is learned from a prespecified model space $\mathbb{H}$ and $w_m \in \mathbb{R}$ is its ensemble weights. Boosting method learns all base models in a sequential way: at each step, we aim to learn a new base model $h(x|\theta_M)$ and an ensemble weight w_M such that it can further improve the ensemble model $F_{M-1}(x)$ after being added to the ensemble

$$F_M(x) = F_{M-1}(x) + w_M h(x|\theta_M)$$

Let $\mathrm{lin}(\mathbb{H})$ denote the set containing all finite linear combinations of any functions in base model space $\mathbb{H}$. Then $F_m(x) \in \mathrm{lin}(\mathbb{H})$ for any $m \in \mathbb{N}$. We assume that the loss function $L(f(x), y)$ is a functional in the functional space $\mathrm{lin}(\mathbb{H})$ and so we need to solve the following optimization problem

$$F_M(x) = \arg \min_{f \in \mathrm{lin}(\mathbb{H})} \frac{1}{n} \sum_{i=1}^{N} L(f(x_n), y_n)$$

or

$$F_M(x) = \arg \min_{w_M, \theta_M} \frac{1}{n} \sum_{i=1}^{N} L(F_{M-1}(x) + w_M h(x|\theta_M), y_n)$$

Now the question is how to optimize w_M and θ_M. The strategy is to optimize θ_M first, and then optimize w_M. Learning base model $h(x|\theta_M)$ by gradient descent is exactly the so-called Gradient Boosting. Here we should view the loss $L(f(x), y)$ as a functional in the function space $f(x) \in \mathrm{lin}(\mathbb{H})$. At $M-1$ iteration $f(x) = F_{M-1}(x)$, by gradient descent[1]

$$F_M(x) \quad = \quad F_{M-1}(x) - w \frac{\partial L(f(x), y)}{\partial f(x)} |_{f(x) = F_{M-1}(x)}$$

Here we can regard w as the learning rate but is needed to be optimized. Note that $F_M(x)$ is not ensured to be in $\mathrm{lin}(\mathbb{H})$ since $-\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)}$ may not belong to the base model space $\mathbb{H}$. Similar to Frank-Wolfe algorithm, we could choose a model $h(x|\theta_M)$ in base model space $\mathbb{H}$, which is mostly similar to $-\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)}$. That is,

$$h(x|\theta_M) = \max_{h(x|\theta) \in \mathbb{H}} \left\langle -\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)}, h(x|\theta) \right\rangle$$

or

$$\theta_M = \arg \max_{\theta : h(x|\theta) \in \mathbb{H}} \left\langle -\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)}, h(x|\theta) \right\rangle$$

1. Note $\frac{\partial L(f(x), y)}{\partial f(x)}$ is a scale function in term of x.

Replacing the negative gradient $-\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)}$ by $h(x|\theta_M)$, we obtain

$$F_M(x) \;=\; F_{M-1}(x) + w h(x|\theta_M)$$

Next, we optimize the learning rate w by

$$w_M = \arg\min_w L(F_{M-1}(x) + w h(x|\theta_M), y)$$

In practice[2], assume we have the dataset $\{(x_i, y_i)\}_{i=1}^n$, we obtain θ_M and w_M by solving

$$\theta_M = \arg\max_{\theta: h(x|\theta) \in \mathbb{H}} \sum_{i=1}^n \left\langle -\frac{\partial L(F_{M-1}(x_i), y_i)}{\partial F_{M-1}(x_i)}, h(x_i|\theta) \right\rangle$$

and

$$w_M = \arg\min_w \sum_{i=1}^n L(F_{M-1}(x_i) + w h(x_i|\theta_M), y_i)$$

which are exactly what we have discussed in class[3].

Apparently the learning rule introduced in the paper "Stochastic Gradient Boosting", is different from the above derived rule. In fact, there the similarity between $-\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)}$ and and $h(x|\theta)$ is measured by the square l_2 norm up to a factor ρ instead of the inner product[4]. That is,

$$\theta_M = \arg\min_{\theta: h(x|\theta) \in \mathbb{H}, \rho} \left\| -\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)} - \rho h(x|\theta) \right\|_2^2$$

Evaluated on datasets $\{(x_i, y_i)\}_{i=1}^n$ yields

$$\theta_M = \arg\min_{\theta: h(x|\theta) \in \mathbb{H}, \rho} \sum_{i=1}^n \left[-\frac{\partial L(F_{M-1}(x_i), y_i)}{\partial F_{M-1}(x_i)} - \rho h(x_i|\theta) \right]^2$$

Similarly, w_M is given by

$$w_M = \arg\min_w \sum_{i=1}^n L(F_{M-1}(x_i) + w h(x_i|\theta_M), y_i)$$

2. $\langle f(x), g(x) \rangle$ evaluated on finite dataset $\{x_1, \cdots, x_n\}$ is $\sum_{i=1}^n f(x_i) g(x_i)$ where $f = (f(x_1), \cdots, f(x_n))$ and $g = (g(x_1), \cdots, g(x_n))$.

3. The lecture adopts its equivalent form $\theta_M = \arg\min_{\theta: h(x|\theta) \in \mathbb{H}} \sum_{i=1}^n \left\langle \frac{\partial L(F_{M-1}(x_i), y_i)}{\partial F_{M-1}(x_i)}, h(x_i|\theta) \right\rangle$.

4. Why the factor ρ is necessary? On the one hand, when measuring functions' similarly it's necessary to consider the scalor. For example, $\sin x$ and $2\sin x$ have same curve shape though they are not identical. On the other hand, $\theta_M = \arg\min_{\theta: h(x|\theta) \in \mathbb{H}, \rho} \left\| -\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)} - h(x|\theta) \right\|_2^2$ may not be solvable. Imagine the case functions $h(x|\theta)$ in $\mathbb{H}$ are bounded in $[0, 1]$ but $-\frac{\partial L(F_{M-1}(x), y)}{\partial F_{M-1}(x)}$ exceeds the interval on many points. Introducing an additional optimizing variable ρ makes the optimization problem more solvable.