[S1] Hello, today I will introduce Gradient Boosting method

[S3] Before introducing the gradient boosting, I will introduce or review the gradient descent at first since those two methods share similar ideas.

[S4] Gradient Descent is an iterative algorithm for finding a local minimum of a differentiable function. The general idea of it is to tweak parameters iteratively in order to minimize a cost function.

You can see that, on the graph, we start by filling theta with random values. Then we improve it gradually, taking one baby step at a time, attempting to decrease the cost function, util the algorithm converges to a minimum.

[S5] And here is the basic steps for the gradient descent method, in short, gradient descent method is to repeat the iterative function theta until convergence.

[S7] Here is an example of applying Gradient Boosting to a regression problem with n samples.

To find an accurate F, we firstly find a weak function F0 to fit. So, the difference between the prediction and the actual value is $y - F_0(x)$. Next, we find h_0 to fit r_0, such that $F_1(x) = F_0(x) + h_0(x)$, and the residual error still exists, which is $r_1(x) = y - F_1(x)$.

Gradient Boosting would keep these steps over and over again util it met the predefined error boundary.

[S8] So, for general problems. During 0 to T steps of Gradient Boosting, suppose there are some imperfect model F_{t-1}.

The algorithm does not change the model directly, instead it try to add an estimator to construct a new model to improve the performance.

[S9] The steps above could be performed iteratively to make function F(x) close to the target value y, and we can express this process by the following equations.

[S10] If we choose square error function as the loss function of gradient boosting, to fit function F , let the prediction value $\hat{y} = F(x)$, the loss function $L = (y - \hat{y})^2/2$, and total error of the whole dataset $J = \sum L$.

So, if we perform gradient operation for the total error J, we will have $y - \hat{y} = -\nabla J$.

Therefore, for square error loss function, the residual error is the negative gradient.

[S11] We can see that for absolute error and Huber error, negative gradient does not equal to residual error, but the negative gradient error is also calculated according the loss function L. And the result of the negative gradient has some relationships between the residual error – They are variants of the residuals error or they are residual error with some adjustment.

Besides, compare to the original residual error $y - F(x)$, the negative gradient of absolute error and Huber error weaken the influence of outliers, or they are less sensitive to errors

[S11] So, here is the steps of applying gradient boosting. First we select a differentiable error function and then construct an initial model, and finally perform iterations until it meet the requirement.

Like other Boosting methods, Gradient Boosting is also an iterative process of building weak models that are progressively enhanced (Boosting) and combined into strong models. However, while other Boosting methods such as AdaBoost iteratively adjust the weights of the samples, Gradient Boosting iteratively fits negative gradients and adjust the weight coefficient of the negative gradient.