# Mathematical Terms Used in Gradient Boosting

Mangukiya Raj

## Mathematical Terms in Gradient Boosting

1. $F(x)$**: Prediction Function** This is the model's predicted value (e.g., predicted stock price) for the input features $x$ at any stage.

2. $L(y, \hat{y})$**: Loss Function** It measures the difference between the actual value $y$ and the predicted value $\hat{y} = F(x)$. A common choice is Mean Squared Error:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

3. $\eta$**: Learning Rate** A small positive number (e.g., 0.1) that controls how much each weak learner contributes to the final model. It helps avoid overfitting.

4. $\gamma$**: Step Size** The optimal amount by which the new tree's prediction should be scaled when added to the current model. It is computed to minimize the loss:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

5. $h_m(x)$**: Weak Learner** A simple model (usually a decision tree) trained at the $m$-th iteration to correct the errors of the previous model.

6. $r_{im}$**: Residual or Pseudo-Residual** The difference (or gradient) between the actual output and the current prediction. It shows how wrong the current model is. Computed as:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

7. $\arg\min$ **(Argument of Minimum)** This notation means: "the value of the parameter

that minimizes the function." For example:

$$\arg\min_{\gamma} L(y, F + \gamma h)$$

means find $\gamma$ that makes $L$ smallest.

8. $\partial$**: Partial Derivative** Used to compute the rate of change of a multivariable function with respect to one variable. In boosting, it's used to compute how the error changes with prediction.

9. $F_0(x)$**: Initial Model** The starting point of the boosting algorithm. Usually the constant prediction that minimizes the loss over the dataset:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

10. $F_m(x)$**: Final Boosted Model** The updated model at iteration $m$, obtained by adding the new weak learner:

$$F_m(x) = F_{m-1}(x) + \eta\gamma_m h_m(x)$$

This process is repeated for $M$ rounds.

## Real-Time Example: Gradient Boosting for Apple Inc. Stock Prediction

Let us consider a real-time application of gradient boosting for predicting the stock price of Apple Inc. ($AAPL$).

We begin by collecting historical data for the last 3 years, including daily closing prices, trading volume, and technical indicators like moving average (MA), exponential moving average (EMA), and Relative Strength Index (RSI).

This data is prepared into features $X = [x_1, x_2, ..., x_n]$ and the target variable $y$ as the stock closing price for the next day. Gradient boosting is applied as follows:

First, we define the initial prediction $F_0(x)$ as the mean of all target prices in the training set. Then, we compute residuals $r_{im}$ for each training example, which represent the errors made by the model at that stage.

At each iteration $m$, a regression tree $h_m(x)$ is trained to predict the residuals. These predictions are scaled by the learning rate $\eta$ and added to the current model:

$$F_m(x) = F_{m-1}(x) + \eta\gamma_m h_m(x)$$

This process continues for $M$ iterations. The final model $F_M(x)$ provides the predicted stock price for Apple on a given day, based on historical trends.

## Numerical Example

Suppose we use the last three closing prices of Apple stock to predict the next day's price. Consider:

- Day 1: $150
- Day 2: $152
- Day 3: $151

Let our target (actual price on Day 4) be $153.

**Step 1: Initial prediction**:

$$F_0(x) = \text{mean of targets} = 151.5$$

**Step 2: Compute residual**:

$$r = y - F_0(x) = 153 - 151.5 = 1.5$$

**Step 3: Train a weak learner (tree)** to predict the residual $r = 1.5$ based on $x = [150, 152, 151]$. Assume it gives $h_1(x) = 1.2$

**Step 4: Update prediction**:

$$F_1(x) = F_0(x) + \eta \cdot h_1(x) = 151.5 + 0.1 \cdot 1.2 = 151.62$$

Repeat the process with new residuals until convergence. Over time, the model approaches the true stock price prediction by minimizing the error iteratively.

## How Gradient Boosting is Used in Stock Price Prediction

Gradient boosting is widely used in financial modeling because it is able to handle the non-linear and noisy nature of stock market data. In stock price prediction, it uses multiple decision trees to learn patterns from historical data such as past prices, volumes, moving averages, RSI, and other technical indicators. By focusing on the mistakes of previous predictions through the use of gradients, the model can fine-tune its performance over time. This iterative correction process helps in capturing short-term price movements and trends more accurately.

Moreover, the boosting framework can rank the importance of features, helping analysts understand which factors most influence price changes. The ability to model complex relationships and improve accuracy with each round makes gradient boosting highly effective for forecasting future prices and making data-driven investment decisions.