Mathematical Foundations of XGBoost

Quant Finance Portfolio

December 25, 2025

Abstract

This document details the mathematics behind XGBoost (eXtreme Gradient Boosting), the algorithm used in Project 13 for price prediction. We cover Gradient Boosted Decision Trees (GBDT), the Regularized Learning Objective, and the second-order approximation used for optimization.

1 Gradient Boosting Framework

Gradient boosting fits an additive ensemble of models to approximate a function $F(x)$. The prediction at step t, $\hat{y}_i^{(t)}$, is given by:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{1}$$

where f_t is a new decision tree added to minimize the residual error.

2 Regularized Learning Objective

XGBoost minimizes the following objective function at step t:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{2}$$

where l is a differentiable convex loss function, and $\Omega(f)$ penalizes the complexity of the tree:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2 \tag{3}$$

Here, T is the number of leaves and w are the leaf weights.

3 Second-Order Approximation

To optimize $\mathcal{L}^{(t)}$ quickly, XGBoost uses a second-order Taylor expansion of the loss function:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{4}$$

where $g_i = \partial_{\hat{y}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial_{\hat{y}}^2 l(y_i, \hat{y}^{(t-1)})$ are the gradient and hessian statistics.

4 Optimal Leaf Weights

For a fixed tree structure with leaf sets $I_j = \{i|q(x_i) = j\}$, the optimal weight w_j^* for leaf j is found by setting the derivative to zero:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{5}$$

This formula allows XGBoost to efficiently calculate the quality of a tree structure and perform effective split finding.

5 Weighted Quantile Sketch

The approximate algorithm requires finding candidate split points among weighted data points (h_i acts as the weight). XGBoost introduces a distributed weighted quantile sketch algorithm. Let $D_k = \{(x_{1k}, h_1), \ldots, (x_{nk}, h_n)\}$ be the data for feature k. The rank function is defined as:

$$r_k(z) = \frac{1}{\sum_{i=1}^{n} h_i} \sum_{i \in D_k, x_{ik} < z} h_i \tag{6}$$

The goal is to find candidates $S_k = \{s_{k1}, \ldots, s_{kl}\}$ such that:

$$|r_k(s_{k,j}) - r_k(s_{k,j+1})| < \epsilon \tag{7}$$

The sketch algorithm merges summaries from different workers with a theoretical guarantee on the accuracy ϵ.

6 System Design for Scalability

To achieve high performance, XGBoost employs several system optimizations:

6.1 Column Block for Out-of-Core Computing

Data is stored in Compressed Column Storage (CSC) format. Each column (feature) is sorted and stored separately. This allows the algorithm to scan columns in parallel when searching for splits. For datasets that exceed RAM, XGBoost uses a block-based out-of-core system where data is sharded onto disk and prefetched asynchronously.

6.2 Cache-Aware Access

Non-continuous memory access causes cache misses. XGBoost buffers gradient statistics (g_i, h_i) in local cache to optimize the accumulation process, achieving significant speedups on modern CPUs.

7 Regularization and Structure Score Proof

The structure score for a leaf j is derived analytically. Starting from the Taylor expansion:

$$\mathcal{L} \approx \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T \tag{8}$$

Taking the derivative w.r.t w_j and setting to 0 yields the optimal weight $w_j^* = -\frac{G_j}{H_j + \lambda}$. Substituting w_j^* back gives the optimal objective value (structure score):

$$\mathcal{L}^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{9}$$

This score is used to evaluate the quality of a tree structure during split finding.

8 Conclusion

XGBoost is not just a gradient boosting algorithm but a highly optimized system. Its mathematical formulation (second-order approximation, regularization) combined with systems engineering (weighted sketch, column blocks) makes it the state-of-the-art for tabular data prediction.