XGBoost - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs

1.1 Introduction

XGBoost is a scalable gradient boosting framework that learns an *additive* ensemble of regression trees. At each boosting round it minimizes a *regularized* objective obtained by a second-order (Newton) Taylor approximation of the loss, yielding closed-form leaf updates and a principled split (gain) criterion. We derive these elements and state all variables with their dimensions.

1.2 Data and Notation

Given training data

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \qquad \mathbf{x}_i \in \mathbb{R}^d \text{ (column vector)}, \quad y_i \in \mathbb{R} \text{ (scalar)},$$

the stage-t model prediction is

$$\hat{y}_i \;=\; F_t(\mathbf{x}_i) \;=\; \sum_{s=1}^{t} f_s(\mathbf{x}_i), \qquad f_s : \mathbb{R}^d \to \mathbb{R}.$$

Let $l(y, \hat{y})$ be a differentiable loss; define per-sample gradient and Hessian (scalars)

$$g_i = \left.\frac{\partial l(y_i, \hat{y})}{\partial \hat{y}}\right|_{\hat{y}=F_{t-1}(\mathbf{x}_i)}, \qquad h_i = \left.\frac{\partial^2 l(y_i, \hat{y})}{\partial \hat{y}^2}\right|_{\hat{y}=F_{t-1}(\mathbf{x}_i)}.$$

1.3 Model Formulation and Regularized Objective

The global objective over T trees is

$$\mathcal{L}(F) \;=\; \sum_{i=1}^{n} l\big(y_i, F(\mathbf{x}_i)\big) \;+\; \sum_{t=1}^{T} \Omega(f_t),$$

with tree regularizer

$$\Omega(f) \;=\; \gamma\, T_f \;+\; \frac{\lambda}{2}\sum_{j=1}^{T_f} \mathbf{w}_j^2 \quad \text{(optionally add L1: } +\,\alpha \sum_{j=1}^{T_f} |\mathbf{w}_j|\text{)}.$$

Here T_f is the number of leaves of f, $q : \mathbb{R}^d \to \{1, \ldots, T_f\}$ maps inputs to leaf indices, and $\mathbf{w} \in \mathbb{R}^{T_f}$ are the leaf weights with $f(\mathbf{x}) = \mathbf{w}_{q(\mathbf{x})}$.

1.4 Second-Order Taylor Approximation (Per-Round Objective)

Adding f_t to F_{t-1} gives $F_t = F_{t-1} + f_t$ and the stage objective

$$\mathcal{L}_t \;=\; \sum_{i=1}^{n} l\big(y_i, F_{t-1}(\mathbf{x}_i) + f_t(\mathbf{x}_i)\big) \;+\; \Omega(f_t).$$

A second-order expansion around $F_{t-1}(\mathbf{x}_i)$ yields, up to constants,

$$\widetilde{\mathcal{L}}^{(t)} \;=\; \sum_{i=1}^{n} \Big(g_i\, f_t(\mathbf{x}_i) + \tfrac{1}{2} h_i\, f_t(\mathbf{x}_i)^2\Big) \;+\; \Omega(f_t).$$

If $f_t(\mathbf{x}) = \mathbf{w}_{q(\mathbf{x})}$ with leaves $j = 1, \ldots, T_f$ and index sets $I_j = \{i : \, q(\mathbf{x}_i) = j\}$, define aggregated statistics

$$G_j = \sum_{i \in I_j} g_i, \qquad H_j = \sum_{i \in I_j} h_i.$$

Then

$$\widetilde{\mathcal{L}}^{(t)} \;=\; \sum_{j=1}^{T_f} \Big(G_j\, \mathbf{w}_j + \tfrac{1}{2}(H_j + \lambda)\, \mathbf{w}_j^2 \;+\; \alpha|\mathbf{w}_j|\Big) \;+\; \gamma T_f. \tag{1}$$

1.5 Optimal Leaf Weights (Closed Form)

L2-only regularization ($\alpha = 0$). Minimizing Eqn. (1) w.r.t. each $\mathbf{w}_j$ gives

$$\boxed{\mathbf{w}_j^{\star} \;=\; -\frac{G_j}{H_j + \lambda}} \quad \Rightarrow \quad \widetilde{\mathcal{L}}^{(t)\star} \;=\; -\frac{1}{2}\sum_{j=1}^{T_f} \frac{G_j^2}{H_j + \lambda} \;+\; \gamma T_f.$$

With L1 (elastic net, $\alpha > 0$). Each leaf solves a 1D convex problem with soft-thresholding:

$$\boxed{\mathbf{w}_j^{\star} \;=\; -\,\frac{\operatorname{sgn}(G_j)\, \max\{|G_j| - \alpha,\, 0\}}{H_j + \lambda}},$$

$$\widetilde{\mathcal{L}}^{(t)\star} \;=\; -\frac{1}{2}\sum_{j=1}^{T_f} \frac{\big(\max\{|G_j| - \alpha, 0\}\big)^2}{H_j + \lambda} \;+\; \gamma T_f.$$

1.6 Greedy Split Criterion (Gain)

Consider splitting a node with (G, H) into left/right children (G_L, H_L) and (G_R, H_R).

L2-only ($\alpha = 0$).

$$\boxed{\text{Gain} \;=\; \frac{1}{2}\left(\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}\right) \;-\; \gamma}.$$

With L1.

$$\boxed{\text{Gain} \;=\; \frac{1}{2}\left(\frac{(\max\{|G_L|-\alpha, 0\})^2}{H_L+\lambda} + \frac{(\max\{|G_R|-\alpha, 0\})^2}{H_R+\lambda} - \frac{(\max\{|G|-\alpha, 0\})^2}{H+\lambda}\right) \;-\; \gamma}.$$

Accept a split if Gain > 0; the same score/gain is used for pruning.

1.7 Algorithmic Steps (One Boosting Round)

1. **Grad/Hess:** For each i, compute g_i, h_i at $\hat{y}_i = F_{t-1}(\mathbf{x}_i)$.
2. **Tree growth:** Starting from the root, evaluate candidate splits using the Gain above (including both default directions for missing values); greedily choose the best positive-Gain split subject to constraints (e.g., max depth, min leaf size).
3. **Leaf values:** Set $\mathbf{w}_j^\star$ by the closed forms above (L2 or L1).
4. **Update with shrinkage:** With learning rate $\eta \in (0, 1]$,

$$F_t(\mathbf{x}) \;=\; F_{t-1}(\mathbf{x}) \;+\; \eta\, \mathbf{w}^\star_{q(\mathbf{x})}.$$

Subsampling (optional). Row/column subsampling reduces variance and accelerates search. *Large-scale splits.* Histogram/quantile-sketch approximations accumulate (G, H) per bin to scan thresholds efficiently. *Missing values.* Choose and store a default branch (left/right) per split by maximizing Gain.

1.8 Common Losses: g_i, h_i Examples

Let $p_i = \sigma(\hat{y}_i) = 1/(1 + e^{-\hat{y}_i})$.

$$\begin{array}{llll}
\text{Squared error: } l = \frac{1}{2}(y-\hat{y})^2 & \Rightarrow & g_i = \hat{y}_i - y_i, & h_i = 1. \\
\text{Logistic (binary): } l = -y\log p - (1-y)\log(1-p) & \Rightarrow & g_i = p_i - y_i, & h_i = p_i(1-p_i). \\
\text{Poisson: } l = e^{\hat{y}} - y\hat{y} & \Rightarrow & g_i = e^{\hat{y}_i} - y_i, & h_i = e^{\hat{y}_i}.
\end{array}$$

Sample/class weights $u_i \ge 0$ are absorbed by $g_i \leftarrow u_i g_i$, $h_i \leftarrow u_i h_i$.

1.9 Variables, Dimensions, and Properties

- $\mathbf{x}_i \in \mathbb{R}^d$: feature vector (dimension $d \times 1$); $y_i \in \mathbb{R}$: target.
- $n, d \in \mathbb{N}$: #samples and #features; $T \in \mathbb{N}$: #boosting rounds.
- $f_t : \mathbb{R}^d \to \mathbb{R}$: tree at round t; T_f: its leaf count (scalar).
- $q : \mathbb{R}^d \to \{1, \ldots, T_f\}$: leaf index function; $\mathbf{w} \in \mathbb{R}^{T_f}$: leaf weights; $\mathbf{w}_j$ may be treated as a scalar leaf score under this notation.
- $I_j \subset \{1, \ldots, n\}$: indices routed to leaf j; $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$.
- Regularization: $\gamma, \lambda, \alpha \in \mathbb{R}_{\ge 0}$; learning rate $\eta \in (0, 1]$.

1.10 Summary

Starting from a regularized empirical risk, XGBoost performs a second-order functional descent restricted to tree functions. A Taylor expansion turns each round into the separable convex problem Eqn. (1), whose minimizers provide closed-form leaf weights (L2: $\mathbf{w}_j^\star = -G_j/(H_j + \lambda)$; with L1: soft-thresholded). The split *Gain* compares parent/children scores and drives greedy tree growth and pruning. Shrinkage, subsampling, missing-value defaults, and approximate split finding make the procedure scalable and robust.