AdaBoost - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs

1.1 Introduction

AdaBoost constructs a *strong* classifier by additive combination of *weak* classifiers. At each round it (i) reweights the training samples to emphasize hard (misclassified) points, (ii) fits a weak learner to minimize the *weighted* error, (iii) chooses a step size by minimizing an exponential surrogate loss, and (iv) updates the sample weights multiplicatively. We derive the optimal weak-learner weight, the weight-update rule, an error bound, and present the functional-gradient view that motivates the procedure.

1.2 Data and Notation

Let the training set be

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

At boosting round $t = 1, \dots, T$:

- $\mathbf{w}^{(t)} = (w_1^{(t)}, \dots, w_n^{(t)})^\top \in \mathbb{R}^n$ are nonnegative sample weights with $\sum_i w_i^{(t)} = 1$; initialize $w_i^{(1)} = \frac{1}{n}$.
- $h_t : \mathbb{R}^d \to \{-1, +1\}$ is the selected weak classifier.
- $\alpha_t \in \mathbb{R}$ is its coefficient in the additive model.

The *additive* score function and the final classifier are

$$F_T(\mathbf{x}) \;=\; \sum_{t=1}^{T} \alpha_t \, h_t(\mathbf{x}), \qquad H(\mathbf{x}) \;=\; \operatorname{sign}\big(F_T(\mathbf{x})\big). \tag{1}$$

1.3 Model Formulation: Exponential Risk and Stagewise Minimization

AdaBoost can be obtained by forward stagewise minimization of the empirical *exponential* loss

$$\widehat{R}_{\exp}(F) \;=\; \frac{1}{n}\sum_{i=1}^{n}\exp\bigl(-\,y_i F(\mathbf{x}_i)\bigr). \tag{2}$$

Given F_{t-1}, we add a new term αh by solving

$$(\alpha_t, h_t) \in \arg\min_{\alpha\in\mathbb{R},\, h}\; \frac{1}{n}\sum_{i=1}^{n}\exp\bigl(-\,y_i(F_{t-1}(\mathbf{x}_i) + \alpha\, h(\mathbf{x}_i))\bigr). \tag{3}$$

Introduce the normalized weights

$$w_i^{(t)} \;\triangleq\; \frac{\exp\bigl(-\,y_i F_{t-1}(\mathbf{x}_i)\bigr)}{\sum_{j=1}^{n}\exp\bigl(-\,y_j F_{t-1}(\mathbf{x}_j)\bigr)}, \qquad \sum_i w_i^{(t)} = 1, \tag{4}$$

under which Eqn. (3) is proportional to the *partition function*

$$Z_t(\alpha, h) \;=\; \sum_{i=1}^{n} w_i^{(t)}\,\exp\bigl(-\,\alpha\, y_i h(\mathbf{x}_i)\bigr). \tag{5}$$

1.4 Optimal Step Size and Choice of Weak Learner

For a discrete weak learner $h : \mathbb{R}^d \to \{-1, +1\}$, define its *weighted error*

$$\epsilon_t(h) \;=\; \sum_{i=1}^{n} w_i^{(t)}\,\mathbf{1}\{y_i \neq h(\mathbf{x}_i)\} \;=\; \frac{1 - \sum_i w_i^{(t)}\, y_i h(\mathbf{x}_i)}{2}. \tag{6}$$

Because $y_i h(\mathbf{x}_i) \in \{\pm 1\}$,

$$Z_t(\alpha, h) \;=\; (1 - \epsilon_t(h))\, e^{-\alpha} + \epsilon_t(h)\, e^{\alpha}.$$

Line search (optimal α_t). Differentiating and setting $\frac{\partial Z_t}{\partial \alpha} = 0$ gives

$$\boxed{\alpha_t^\star \;=\; \frac{1}{2}\ln\!\left(\frac{1-\epsilon_t}{\epsilon_t}\right)}, \qquad \epsilon_t \equiv \epsilon_t(h_t). \tag{7}$$

Plugging Eqn. (7) back into Eqn. (6) yields the minimized partition value

$$Z_t^\star(h) \;=\; 2\sqrt{\epsilon_t(h)\,(1 - \epsilon_t(h))}. \tag{8}$$

Hence minimizing $Z_t^\star(h)$ is equivalent to minimizing the weighted error $\epsilon_t(h)$; choose

$$h_t \in \arg\min_h\; \epsilon_t(h) \quad \text{under weights } \mathbf{w}^{(t)}. \tag{9}$$

1.5 Sample-Weight Update (Emphasizing Hard Examples)

With (α_t, h_t) chosen, update the weights multiplicatively and renormalize:

$$w_i^{(t+1)} \;=\; \frac{w_i^{(t)}\,\exp\bigl(-\,\alpha_t\, y_i h_t(\mathbf{x}_i)\bigr)}{Z_t}, \qquad Z_t \;=\; \sum_{j=1}^{n} w_j^{(t)}\,\exp\bigl(-\,\alpha_t\, y_j h_t(\mathbf{x}_j)\bigr). \tag{10}$$

Thus correctly classified points ($y_i = h_t(\mathbf{x}_i)$) get down-weighted by $e^{-\alpha_t}$, while misclassified points get up-weighted by $e^{+\alpha_t}$.

1.6 Training-Error Bound and Edge

The empirical training error of $H(\mathbf{x}) = \text{sign}(F_T(\mathbf{x}))$ obeys

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{y_i \neq H(\mathbf{x}_i)\} \;\leq\; \prod_{t=1}^{T} Z_t^\star \;=\; \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}. \tag{11}$$

Let the *edge* be $\gamma_t \triangleq \frac{1}{2} - \epsilon_t$. Then $Z_t^\star = \sqrt{1-4\gamma_t^2} \leq e^{-2\gamma_t^2}$ and

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{y_i \neq H(\mathbf{x}_i)\} \;\leq\; \exp\Big(-2\sum_{t=1}^{T}\gamma_t^2\Big), \tag{12}$$

so any sequence with $\epsilon_t < \frac{1}{2}$ reduces the bound exponentially.

1.7 Functional-Gradient View

The gradient of Eqn. (2) at the training points is

$$\frac{\partial \widehat{R}_{\text{exp}}}{\partial F}(\mathbf{x}_i) \;=\; -\frac{1}{n}\, y_i \exp\big(-y_i F(\mathbf{x}_i)\big) \;\propto\; -y_i\, w_i^{(t)}.$$

Thus, choosing h_t to minimize $\sum_i w_i^{(t)} \mathbf{1}\{y_i \neq h(\mathbf{x}_i)\}$ is a projection of the negative gradient onto the class of weak learners, while Eqn. (7) is the exact line search along h_t.

1.8 Algorithm (Discrete AdaBoost)

1. **Initialize:** $w_i^{(1)} = \frac{1}{n}$, $F_0 \equiv 0$.
2. **For** $t = 1, \ldots, T$**:**
   (a) Fit h_t to minimize $\epsilon_t(h)$ under weights $\mathbf{w}^{(t)}$; compute $\epsilon_t = \epsilon_t(h_t)$.
   (b) Set $\alpha_t = \dfrac{1}{2}\ln\Big(\dfrac{1-\epsilon_t}{\epsilon_t}\Big)$.
   (c) Update weights via Eqn. (10).
   (d) Update $F_t(\mathbf{x}) \leftarrow F_{t-1}(\mathbf{x}) + \alpha_t h_t(\mathbf{x})$.
3. **Output:** $H(\mathbf{x}) = \text{sign}\big(F_T(\mathbf{x})\big)$.

1.9 Variants (Brief)

Real AdaBoost (confidence-rated). Allow $h_t : \mathbb{R}^d \to \mathbb{R}$ (e.g., leaf-wise real scores). Minimizing $\sum_i w_i^{(t)} e^{-y_i h_t(\mathbf{x}_i)}$ yields, for a region R (e.g., a leaf),

$$h_t^\star|_R \;=\; \tfrac{1}{2}\log \frac{\sum_{i\in R: y_i=+1} w_i^{(t)}}{\sum_{i\in R: y_i=-1} w_i^{(t)}},$$

and one can take $\alpha_t = 1$ (absorbed into h_t).

Multiclass (SAMME). For K classes and discrete h_t, the coefficient becomes

$$\alpha_t \;=\; \log\frac{1-\epsilon_t}{\epsilon_t} + \log(K-1), \quad \widehat{y}(\mathbf{x}) = \arg\max_k \sum_{t=1}^{T} \alpha_t\, \mathbf{1}\{h_t(\mathbf{x}) = k\}.$$

1.10 Summary of Variables and Dimensions

- $\mathbf{x}_i \in \mathbb{R}^d$: feature vector (dimension $d \times 1$); $y_i \in \{-1, +1\}$: label.
- $n, d \in \mathbb{N}$: #samples and #features; $T \in \mathbb{N}$: #boosting rounds.
- $\mathbf{w}^{(t)} \in \mathbb{R}^n$: sample-weight vector at round t; elements $w_i^{(t)} \geq 0$ and $\sum_i w_i^{(t)} = 1$.
- $h_t : \mathbb{R}^d \to \{-1, +1\}$: weak learner at round t; $\alpha_t \in \mathbb{R}$: its coefficient.
- $\epsilon_t \in [0, 1]$: weighted error of h_t; $Z_t > 0$: normalization in Eqn. (10).
- $F_T(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$: additive score; $H(\mathbf{x}) = \mathrm{sign}(F_T(\mathbf{x}))$: final classifier.

1.11 Summary

From first principles: AdaBoost is the forward stagewise minimizer of the empirical exponential risk: (i) define weights from the current additive model Eqn. (4); (ii) pick h_t by minimizing weighted error Eqn. (9); (iii) take the exact line-search step Eqn. (7); (iv) update weights multiplicatively Eqn. (10); (v) aggregate predictions as in Eqn. (1). The product-of-partition-functions bound Eqn. (11) shows exponential decay of training error when each weak learner has edge $\gamma_t > 0$.