Complementi di Algoritmi e Strutture Dati

Boosting

Instructor: *Emmanuel Esposito* version of May 22, 2025

1 Learning a binary function

Instance. Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be a finite instance space containing n data points, and $\mathcal{Y} = \{-1, +1\}$ be the label space containing a negative label (-1) and a positive label $(+1)$. Assume that there exists a binary function $f\colon \mathcal{X} \to \mathcal{Y}$ that provides a labeling to the data points, and suppose you are given a finite set $\mathcal{H} = \{h_1, \dots, h_m\}$ of m binary functions $h_i\colon \mathcal{X} \to \mathcal{Y}$; the class $\mathcal{H}$ is typically known as the *hypothesis class* and f as the *ground truth*. Finally, let $\mathbf{q}$ be any fixed but unknown distribution over $\mathcal{X}$; equivalently, we actually consider $\mathbf{q}$ to be a distribution over the indices $[n] = \{1, \dots, n\}$ of points in $\mathcal{X}$.

Cost function. We measure the performance of the predictions given by any function $h\colon \mathcal{X} \to \mathcal{Y}$ against the ground truth f, with respect to the data distribution $\mathbf{q}$, using a *cost function* $c\colon \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ that assign a cost $c(\widehat{y}, y) \in [0, 1]$ to any pair of predicted label $\widehat{y} \in \mathcal{Y}$ and true label $y \in \mathcal{Y}$. In particular, we assume for ease of presentation that $c(\widehat{y}, y) = 0$ if and only if $\widehat{y} = y$. Then, we can define the *loss* of h over distribution $\mathbf{q}$ as

$$\ell_{\mathbf{q}}(h) = \mathop{\mathbb{E}}_{j \sim \mathbf{q}} \big[c\big(h(x_j), f(x_j)\big)\big] = \sum_{j \in [n]} q_j \cdot c\big(h(x_j), f(x_j)\big) \ .$$

Note that one can think of c as a 2-by-2 matrix given by $C = \big[c(\widehat{y}, y)\big]_{\widehat{y}, y \in \mathcal{Y}} = \left(\begin{smallmatrix} 0 & c^+ \\ c^- & 0 \end{smallmatrix}\right)$.

Notation	Meaning
$\mathcal{X} = \{x_1, \dots, x_n\}$	finite instance space
$\mathcal{Y} = \{-1, +1\}$	binary label space
$f\colon \mathcal{X} \to \mathcal{Y}$	ground-truth labeling
$\mathcal{H} = \{h_1, \dots, h_m\}$	hypothesis class
$\mathbf{q} = (q_1, \dots, q_n)^\top$	distribution over $[n]$
$c\colon \mathcal{Y} \times \mathcal{Y} \to [0, 1]$	cost function

Table 1: Summary table for notation.

Goal. The goal we want to achieve is the following: given access to $\mathcal{H}$ and f, design an aggregation $h^\star = h^\star(h_1, \dots, h_m)$ of the hypothesis functions in $\mathcal{H}$ such that $h^\star = f$.

2 Boosting as a game

The idea is that each hypothesis function h_i provides some information about the ground-truth labeling f, depending on how small its loss is over the given distribution $\mathbf{q}$.

Binary prediction game. If we lacked the information provided by $\mathcal{H}$, the best we could hope for is to select the best distribution over the two possible labels $\mathcal{Y}$ such as to minimize

the expected cost in the worst case. This translates into a two-player zero-sum game with a 2-by-2 cost (instead of payoff) matrix C, which here we call the *binary prediction game*. Given the game C, von Neumann's minimax theorem tells us that the minimax strategy achieves loss equal to the value of the game V_C when facing the worst possible true label. However, this way of predicting the ground truth f is neither deterministic nor perfect in general.

We will show in what follows that it is possible to resolve both downsides provided a sufficiently weak and reasonable assumption on how $\mathcal{H}$ relates to f. The boosting framework considers what is commonly known as the *weak-learning assumption*.

Assumption 1 (Weak learning)**.** *For any distribution* $\mathbf{q}$ *over* $[n]$*, there exists* $i \in [m]$ *such that the hypothesis* h_i *guarantees* $\ell_{\mathbf{q}}(h_i) \leq V_C - \gamma$ *for some constant* $\gamma > 0$*.*

The weak-learning assumption essentially states that, given any distribution $\mathbf{q}$, we can always find a function in $\mathcal{H}$ that guarantees some *advantage* (or edge) γ over the loss compared to the value V_C obtained by predicting at best while ignoring $\mathcal{X}$ and $\mathcal{H}$.

Boosting game. We now define a more structured game that the one given by matrix C. Let $M \in [0,1]^{m \times n}$ be such that each row i corresponds to hypothesis h_i and each column j corresponds to data point x_j. Each entry of M is defined as

$$M_{i,j} = c(h_i(x_j), f(x_j)) \qquad \forall i \in [m], \forall j \in [n] \ .$$

In order words, $M_{i,j}$ is the cost incurred by the prediction $h_i(x_j)$ given by hypothesis h_i on the data point x_j. Observe that $\ell_{\mathbf{q}}(h_i) = (M\mathbf{q})_i$, and so the weak-learning assumption can be equivalently rewritten as

$$\max_{\mathbf{q}} \min_i \ell_{\mathbf{q}}(h_i) = \max_{\mathbf{q}} \min_i (M\mathbf{q})_i \leq V_C - \gamma \ .$$

By von Neumann's minimax theorem, the left-hand side of the inequality is

$$\max_{\mathbf{q}} \min_i (M\mathbf{q})_i = \max_{\mathbf{q}} \min_{\mathbf{p}} \mathbf{p}^\top M\mathbf{q} = \min_{\mathbf{p}} \max_{\mathbf{q}} \mathbf{p}^\top M\mathbf{q} = \min_{\mathbf{p}} \max_j (M^\top \mathbf{p})_j \ ,$$

and thus, together with the previous inequality given by the weak-learning assumption, we equivalently have that

$$\min_{\mathbf{p}} \max_j (M^\top \mathbf{p})_j \leq V_C - \gamma \ .$$

In other words, there exists some distribution $\mathbf{p}$ over the indices $[m]$ of the hypotheses such that the *randomized hypothesis* h_I given by sampling $I \sim \mathbf{p}$ has expected cost at most $V_C - \gamma$ on *any* data point x_j. Define

$$\mathbf{p}^\star = \arg\min_{\mathbf{p}} \max_j (M^\top \mathbf{p})_j \tag{1}$$

to be such a distribution.[1]

Cost-sensitive majority vote. Think of the mixed strategy $\mathbf{p}^\star = (p_1^\star, \ldots, p_m^\star)^\top$ as a sort of weighting of the hypotheses in $\mathcal{H}$, placing more weight to hypotheses that achieve small cost over points in $\mathcal{X}$ (as given by the matrix M). Then, given $\mathbf{p}^\star$, we may devise a deterministic way to predict labels. Given any fixed data point $x \in \mathcal{X}$, the intuition is to test the randomized

[1] Remark: $\mathbf{p}^\star$ can be efficiently computed via a linear program.

hypothesis h_I on each of the two possible labels in $\mathcal{Y}$ and to compute its expected cost. In order words, for each $y \in \mathcal{Y}$ we compute

$$\mathop{\mathbb{E}}_{I \sim \mathbf{p}^\star} \big[c(h_I(x), y) \big] = \sum_i p_i^\star c(h_i(x), y) = c(-y, y) \sum_{i : h_i(x) \neq y} p_i^\star \, .$$

Intuitively, we would select the label y that minimizes such an expected cost. The final predictor would then become

$$h^\star(x) = \arg\min_{y \in \mathcal{Y}} c(-y, y) \sum_{i : h_i(x) \neq y} p_i^\star \qquad \forall x \in \mathcal{X} \, ,$$

that is, $h^\star$ selects the label y that the majority of $\mathcal{H}$ weighted by $\mathbf{p}^\star$ predicts correctly, after factoring in the contribution of the two possible non-negative costs $c(-1, +1)$ and $c(+1, -1)$, respectively.

In order to prove that $h^\star$ is indeed a perfect predictor for the ground-truth labeling f, we need the following fact about the value of the binary prediction game C.

Fact 1. *The binary prediction game C has $V_C \leq \max\{\alpha c^+, (1-\alpha) c^-\}$ for any $\alpha \in [0, 1]$.*

We are now ready to prove the main result.

Theorem 2. *The function $h^\star$ is equal to f.*

Proof. Assume by way of contradiction that $h^\star \neq f$. This means that there exists an index $k \in [n]$ such that $h^\star(x_k) \neq f(x_k)$. Let $y_k = f(x_k)$, and define

$$w^- = \sum_{i : h_i(x_k) \neq y_k} p_i^\star \qquad \text{and} \qquad w^+ = \sum_{i : h_i(x_k) = y_k} p_i^\star = 1 - w^- \, .$$

Hence, we have that $h^\star(x_k) \neq f(x_k)$ corresponds to $h^\star(x_k) = -y_k$. Using the definition of $h^\star$, this means that

$$(1 - w^-) c(y_k, -y_k) = w^+ c(y_k, -y_k) \leq w^- c(-y_k, y_k) \, .$$

Consequently, we obtain that

$$w^- c(-y_k, y_k) = \max\{ w^- c(-y_k, y_k) \, , (1 - w^-) c(y_k, -y_k) \} \geq V_C \, ,$$

where the inequality follows by Fact 1. On the other hand,

$$\begin{aligned} w^- c(y_k, -y_k) = \sum_i p_i^\star c(h_i(x_k), y_k) &= (M^\top \mathbf{p}^\star)_k \\ &\leq \max_j \, (M^\top \mathbf{p}^\star)_j = \min_{\mathbf{p}} \max_j \, (M^\top \mathbf{p})_j \leq V_C - \gamma \, , \end{aligned}$$

where the last inequality follows by the weak-learning assumption. Combining the two inequalities, we obtain $V_C \leq w^- c^+ \leq V_C - \gamma$, which is a contradiction since $\gamma > 0$. □

3 Exercises (optional)

Exercise 1. Prove that $V_C = \min_{\alpha \in [0,1]} \max\{\alpha c^+, (1-\alpha) c^-\}$. (Note that this implies Fact 1.)

Exercise 2. Prove that $V_C = \frac{c^- c^+}{c^- + c^+}$. (Hint: use Exercise 1.)

Exercise 3. Show how to compute $\mathbf{p}^\star$ (Equation (1)) exactly using a linear program.