

Recitation 8

Lecturer: Regev Schweiger

Scribe: Yishay Mansour

8.1 AdaBoost - Review

Reminder: Algorithm A learns Weak-PAC a concept class C with H if:

$\exists \gamma > 0$,

$\forall c^* \in C$, (target function)

$\forall D$, (distribution)

$\forall \delta < \frac{1}{2}$,

With probability $1 - \delta$, algorithm A outputs an hypothesis $h \in H$ such that $error(h) \leq \frac{1}{2} - \gamma$.

Input: A set of m classified examples: $S = \{\langle \mathbf{x}_1, y_1 \rangle, \langle \mathbf{x}_2, y_2 \rangle, \dots, \langle \mathbf{x}_m, y_m \rangle\}$ where $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$ and a class of weak learners H .

Definitions: Let D_t denote the distribution (weights) of the examples at iteration t : $D_t(x_i)$ is the weight of example $\langle \mathbf{x}_i, y_i \rangle$ at iteration t .

Initialization:

$$D_1(\mathbf{x}_i) = \frac{1}{m} \quad \forall i \in \{1, \dots, m\}$$

Iterate: $t = 1, 2, \dots, T$

$$h_t = \arg \min_{h \in H} \Pr_{\mathbf{x} \sim D_t} [h(\mathbf{x}) \neq y]$$

$$\epsilon_t = \Pr_{\mathbf{x} \sim D_t} [h_t(\mathbf{x}) \neq y]$$

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$$

$$D_{t+1}(\mathbf{x}_i) = \frac{D_t(\mathbf{x}_i) e^{-\alpha_t y_i h_t(\mathbf{x}_i)}}{Z_t}$$

Finally:

$$H(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}) \right)$$

Figure 8.1 illustrates the AdaBoost algorithm.

¹Adapted from *Foundations of Machine Learning*

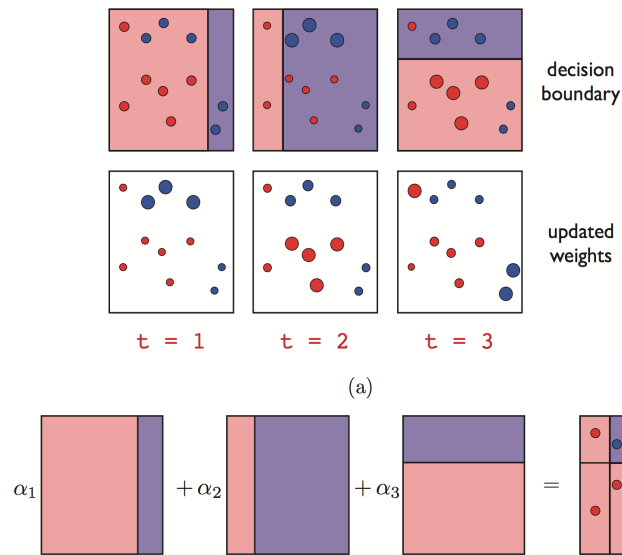


Figure 8.1: Example of AdaBoost with axis-aligned hyperplanes as base learners. (a) The top row shows decision boundaries at each boosting round. The bottom row shows how weights are updated at each round, with incorrectly (resp., correctly) points given increased (resp., decreased) weights. The size of the points represents the distribution weight assigned to them at each boosting round. (b) Visualization of final classifier, constructed as a linear combination of base learners.¹

8.2 Feature decision stumps as weak learners

We show how to efficiently compute the optimal weak learner in the first step of each AdaBoost iteration for a class H of feature decision stumps, given the distribution/weights $D(x_i)$ over the sample set:

Assume each example \mathbf{x} has a set of features $f_1(\mathbf{x}), \dots, f_k(\mathbf{x}), \dots$. Such features can be one of a few types:

1. Binary features: $f_k(\mathbf{x}) \in \{0, 1\}$.
2. Discrete features: for example $f_k(\mathbf{x}) \in \{a, b, c\}$.
3. Continuous features: $f_k(\mathbf{x}) \in \mathbb{R}$.

8.2.1 Binary Features

We have *decision stumps* which have two parameters $c_0, c_1 \in \{+1, -1\}$. Let us fix a binary feature $f_k(\mathbf{x})$ ². Then a stump has the form

$$h_{c_0, c_1}^k(\mathbf{x}) = \begin{cases} c_0 & f_k(\mathbf{x}) = 0 \\ c_1 & f_k(\mathbf{x}) = 1 \end{cases}$$

Given a distribution $D(\mathbf{x}_i)$ we select the optimal c_0 and c_1 for h^k (which is based on feature f_k) as follows. For $j \in \{0, 1\}$ and $b \in \{+1, -1\}$ we compute

$$w_b^j = \sum_{\{i: f_k(\mathbf{x}_i) = j, y_i = b\}} D(\mathbf{x}_i)$$

Note that those values may be computed during a single pass over the m samples. We have that the error of h_{c_0, c_1}^k is

$$\text{error}(h_{c_0, c_1}^k) = w_{-c_0}^0 + w_{-c_1}^1$$

This is because h_{c_0, c_1}^k errs on samples $\langle \mathbf{x}_i, -c_0 \rangle$ with $f_k(\mathbf{x}_i) = 0$ or samples $\langle \mathbf{x}_i, -c_1 \rangle$ with $f_k(\mathbf{x}_i) = 1$. Therefore the optimal stump h_{c_0, c_1}^k has

$$c_j = \begin{cases} +1 & w_{-1}^j \leq w_{+1}^j \\ -1 & w_{-1}^j > w_{+1}^j \end{cases}$$

8.2.2 Discrete Features

The feature $f_k(\mathbf{x})$ has ℓ different values, i.e., $f_k(\mathbf{x}) \in \{v_1, \dots, v_\ell\}$. Similarly, we can set the weak learner to be

$$h_{c_1, \dots, c_\ell}^k(\mathbf{x}) = \begin{cases} c_1 & f_k(\mathbf{x}) = v_1 \\ \vdots & \vdots \\ c_\ell & f_k(\mathbf{x}) = v_\ell \end{cases}$$

Again, the parameters $c_j \in \{+1, -1\}$. As before we can set (during a single pass over the sample set), for $j \in [1, \ell]$ and $b \in \{+1, -1\}$ we compute

$$w_b^j = \sum_{\{i: f_k(\mathbf{x}_i) = v_j, y_i = b\}} D(\mathbf{x}_i)$$

The error is

$$\text{error}(h_{c_1, \dots, c_\ell}^k) = \sum_{j=1}^{\ell} w_{-c_j}^j$$

and we select c_j as before.

²Once we find the optimal stump for each k , an outer loop will choose the best across the possible values of k . Therefore, in what follows we assume a fixed k .

8.2.3 Continuous Features

We now have that $f_k(\mathbf{x}) \in \mathbb{R}$. A possible form for a weak learner is

$$h_v^k(\mathbf{x}) = \begin{cases} c_0 & f_k(\mathbf{x}) \leq v \\ c_1 & f_k(\mathbf{x}) > v \end{cases}$$

Technically v can be of an infinite number of values. However, if we have a sample of size m we can sort the values $f_k(\mathbf{x}_1) \leq \dots \leq f_k(\mathbf{x}_m)$. Although there are an infinite number of possible values for v , there are really only $m + 1$ interesting ones. Namely, v_0, \dots, v_m , where $v_0 \leq f_k(\mathbf{x}_1)$, $v_j \in (f_k(\mathbf{x}_j), f_k(\mathbf{x}_{j+1}))$ and $v_m > f_k(\mathbf{x}_m)$ ³.

Now, for any choice of $m + 1$ such values for v we can apply the method used for the case of binary features above (replacing the binary condition over values of b , $f_k(\mathbf{x}) = 0$, $f_k(\mathbf{x}) = 1$ with the two options $f_k(\mathbf{x}) \leq v$, $f_k(\mathbf{x}) > v$) to find the optimal c_0, c_1 .

Observe, however, that while boosting stumps are widely used in combination with Adaboost and can perform well in practice, the algorithm that returns the stump with the minimal empirical error is not a weak learner! Consider, for example, the simple XOR example with four data points lying in \mathbb{R}^2 , where points in the second and fourth quadrants are labeled positively and those in the first and third quadrants negatively. Then, no decision stump can achieve an accuracy better than $1/2$.

³For example, any two values v and u in the range $[f_k(\mathbf{x}_j), f_k(\mathbf{x}_{j+1}))$ are equivalent in terms of the values $h_v(\cdot)$ and $h_u(\cdot)$ over the m samples.