

Notes on Adaboost

Raphael Sznitman

Given a training set: $\mathcal{L} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, where $x_i \in \mathcal{R}^k$ and $y_i \in \{+1, -1\}$, our goal is to build a classifier $F : \mathcal{X} \rightarrow \{+1, -1\}$, such that

$$F(x) = \text{sign}\left(\sum_{m=1}^M \alpha_m h_m(x)\right)$$

where $h_m(x)$ is a “weak classifier” - does slightly better than chance when classifying x - and M is number of classifier we will use. With each datapoint (x_i, y_i) we associate a weight, $w_i \in \mathcal{R}$, such that $\mathcal{L} = \{(x_1, y_1, w_1), \dots, (x_n, y_n, w_n)\}$. Each weight will indicate how hard that point is to classify with the current classifier. Note that these weights are normalized such that $\sum_{i=1}^n w_i = 1$.

Will choose to construct this classifier by iteratively picking a new h_m and α_m .

$$m = 1 : f_0(x) = \alpha_1 h_1(x)$$

$$m = 2 : f_1(x) = f_0(x) + \alpha_1 h_1(x)$$

\vdots

$$m = M : f_M(x) = f_{M-1}(x) + \alpha_m h_m(x)$$

such that each time step, we we simply need to pick a h_m and α_m . Note that as the classifier is constructed, the weight of each point will be adjusted in order to reflect the difficulty of the point and the current function. In order to pick a h_m and α_m , we define the expected loss of this classifier (the amount of error the classifier does when classifying) over all the training examples as

$$\begin{aligned} L(f_m) &= \sum_{i=1}^n e^{-y_i f_m(x_i)} \\ &= \sum_{i=1}^n e^{-y_i (f_{m-1}(x_i) + \alpha_m h_m(x))} \end{aligned} \tag{1}$$

At time step m , we assume that all α 's and h 's from previous steps have already been chosen and are fixed. Hence, we simply need to choose α_m and h_m and do this by minimizing the expected loss with respect to α_m and h_m .

$$L(f_m) = \sum_{i=1}^n e^{-y_i (f_{m-1}(x_i) + \alpha_m h_m(x))}$$

$$\begin{aligned}
&= \sum_{i=1}^n e^{-y_i f_{m-1}(x_i) - y_i \alpha_m h_m(x)} \\
&= \sum_{i=1}^n e^{-y_i f_{m-1}(x_i)} e^{-y_i \alpha_m h_m(x)} \\
&= \sum_{i=1}^n w_i^{m-1} e^{-y_i \alpha_m h_m(x)} \tag{2}
\end{aligned}$$

where $w_i^{m-1} = e^{-y_i f_{m-1}(x_i)}$, since this is equivalent to the previous weight of a given point x_i . We further break the loss function by

$$L(f_m) = e^{-\alpha_m} \sum_{i \in \mathcal{N}} w_i^{m-1} + e^{\alpha_m} \sum_{i \notin \mathcal{N}} w_i^{m-1}$$

where $\mathcal{N} = \{i | y_i = h_m(x_i)\}$. That is, \mathcal{N} is the set of data points which are correctly classified by $h_m(\cdot)$. Then

$$L(f_m) = e^{-\alpha_m} \sum_{i=1}^n w_i^{m-1} - e^{-\alpha_m} \sum_{i=1}^n w_i^{m-1} I(y_i \neq h_m(x_i)) + e^{\alpha_m} \sum_{i \notin \mathcal{N}} w_i^{m-1}$$

where $I(\cdot)$ is the indicator function. We can re-write this as

$$\begin{aligned}
L(f_m) &= e^{-\alpha_m} \sum_{i=1}^n w_i^{m-1} + (e^{\alpha_m} - e^{-\alpha_m}) \sum_{i=1}^n w_i^{m-1} I(y_i \neq h_m(x_i)) \\
&= e^{-\alpha_m} + (e^{\alpha_m} - e^{-\alpha_m}) \sum_{i=1}^n w_i^{m-1} I(y_i \neq h_m(x_i)) \tag{3}
\end{aligned}$$

since $\sum_{i=1}^n w_i = 1$.

From this last equation, we can see that picking h_m consists in computing

$$h_m = \arg \min_{h \in H} \sum_{i=1}^n w_i^{m-1} I(y_i \neq h_m(x_i))$$

which is picking the weak classifier which has the smallest weighted error. Computing α_m can be done differentiating $L(f_m)$ with respect to α_m , setting it to zero and solving for α_m . We now show this:

$$\frac{dL(f_m)}{d\alpha_m} = -e^{-\alpha_m} + e^{\alpha_m} \sum_{i=1}^n w_i^{m-1} I(y_i \neq h_m(x_i)) + e^{-\alpha_m} \sum_{i=1}^n w_i^{m-1} I(y_i \neq h_m(x_i)) = 0$$

Let $\epsilon_m = \sum_{i=1}^n w_i^{m-1} I(y_i \neq h_m(x_i))$, then

$$\begin{aligned}
-e^{-\alpha_m} + \epsilon_m e^{\alpha_m} + \epsilon_m e^{-\alpha_m} &= 0 \\
\epsilon_m e^{\alpha_m} + \epsilon_m e^{-\alpha_m} &= e^{-\alpha_m} \\
\epsilon_m e^{2\alpha_m} + \epsilon_m &= 1 \\
\alpha_m &= \frac{1}{2} \log\left(\frac{1 - \epsilon_m}{\epsilon_m}\right)
\end{aligned}$$

Once α_m and h_m have been computed, then we must re-weight all the w 's in order to adjust how the new classifier performs with the data. To do this we recall that

$$\begin{aligned} w_i^m &= e^{-y_i F_m(x_i)} \\ &= e^{-y_i f_{m-1}(x_i) - y_i \alpha_m h_m(x_i)} \\ &= w_i^{m-1} e^{-y_i \alpha_m h_m(x_i)} \end{aligned}$$

Note that $-y_i h(x_i) = 2I(y_i \neq h(x_i))$. Then,

$$\begin{aligned} w_i^m &= w_i^{m-1} e^{-y_i \alpha_m h_m(x_i)} \\ &= w_i^{m-1} e^{\alpha_m 2I(y_i \neq h(x_i)) - \alpha_m} \\ &= w_i^{m-1} e^{\alpha_m 2I(y_i \neq h(x_i))} e^{-\alpha_m} \end{aligned}$$

Once this has been calculated for each point, all w 's must be normalized, such that $\sum_{i=1}^n w_i = 1$.