Introduction to Machine Learning **Fall Semester, 2013**

Recitation 8: December 8

Lecturer: Mariano Schain *Scribe: ym*

8.1 Adaboost

8.1.1 Algorithm

Input: A set of m classified examples: $S = \{< x_1, y_1 >, < x_2, y_2 >, \cdots, < x_m, y_m >\}$ where $y_i \in \{-1, 1\}$ 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 $< x_i, y_i >$ at iteration t.
Initialization:

$$D_1(i) = \frac{1}{m} \quad \forall i \in \{1, \cdots, m\}$$

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

$$\begin{aligned} h_t &= \arg\min_{h \in H} \Pr_{x_i \sim D_t}[h(x_i) \neq y_i] \\ \epsilon_t &= \Pr_{x_i \sim D_t}[h_t(x_i) \neq y_i] \\ \alpha_t &= \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t} \\ D_{t+1}(x_i) &= \frac{D_t(x_i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \end{aligned}$$

Finally:

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

8.1.2 Error of h_t on D_{t+1}

We will show that the hypothesis h_t has error $1/2$ on D_{t+1} and hence $h_{t+1} \neq h_t$ (since by the weak learning assumption, the weak learner has error bounded away from $\frac{1}{2}$).

Claim 8.1 $\Pr_{x_i \sim D_{t+1}}[h_t(x_i) \neq y_i] = \frac{1}{2}$

We first show a few quantities that we will use latter.

$$e^{-\alpha_t} = \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} \qquad e^{\alpha_t} = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$$

Therefore,

$$\epsilon_t e^{\alpha_t} = \sqrt{\epsilon_t(1-\epsilon_t)} = (1-\epsilon_t)e^{-\alpha_t}$$

Recall that,

$$\sum_{i:y_i \neq h_t(x_i)} D_t(x_i) = \epsilon_t.$$

Now the normalization is,

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t(x_i) e^{-y_i \alpha_t h_t(x_i)} \\ &= \sum_{i:y_i = h_t(x_i)} D_t(x_i) e^{-\alpha_t} + \sum_{i:y_i \neq h_t(x_i)} D_t(x_i) e^{\alpha_t} \\ &= (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t} \\ &= 2\sqrt{\epsilon_t(1-\epsilon_t)} \end{aligned}$$

Now we can prove the claim.

$$\begin{aligned} \Pr_{x_i \sim D_{t+1}}[h_t(x_i) \neq y_i] &= \sum_{i:h_t(x_i)\neq y_i} D_{t+1}(x_i) \\ &= \sum_{i:h_t(x_i)\neq y_i} \frac{D_t(x_i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \\ &= \sum_{i:h_t(x_i)\neq y_i} D_t(x_i) \frac{e^{\alpha_t}}{Z_t} \\ &= \frac{e^{\alpha_t}}{Z_t}\epsilon_t = \frac{\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\epsilon_t}{2\sqrt{\epsilon_t(1-\epsilon_t)}} = \frac{1}{2} \end{aligned}$$

8.1.3 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 x has a set of features $f_1(x), \ldots, f_k(x), \ldots$. Such features can be one of a few types:

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

Binary Features

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

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

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

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

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} is

$$error(h_{c_0,c_1}) = w^0_{-c_0} + w^1_{-c_1}$$

This is because h_{c_0,c_1} errs on samples $< x_i, -c_0 >$ with $f_k(x_i) = 0$ or samples $< x_i, -c_1 >$ with $f_k(x_i) = 1$. Therefore the optimal stump h_{c_0,c_1} has

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

Discrete Features

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

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

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

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

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

The error is

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

and we select c_j as before.

Continuous Features

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

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

Technically v can be of an infinite number of values. However, if we have a sample of size m we can sort the values $f_k(x_1) \leq \cdots \leq f_k(x_m)$. Although there are an infinite number of possible values for v, there are really only $m+1$ interesting ones. Namely, $v_0, \ldots, v_m$, where $v_0 \leq f_k(x_1)$, $v_j \in (f_k(x_j), f_k(x_{j+1}]$ and $v_m > f_k(x_m)$ [2].

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, $f_k(x) = 0$, $f_k(x) = 1$ with the two options $f_k(x) \leq v$, $f_k(x) > v$) to find the optimal c_0, c_1.

[2]For example, any two values v and u in the range $[f_k(x_j), f_k(x_{j+1}]$ are equivalent in terms of the values $h_v(\cdot)$ and $h_u(\cdot)$ over the m samples.