Machine Learning **Fall 2023**

Lecture 9: Boosting, Online Learning

Lecturer: Liwei Wang *Scribe: Yixing Liu, Kaiwen Hu,Haoran Li, Gongle Xue, Bowen Ye*

Boosting: Given a weak learner, boost it to a strong learner, also called ensemble.

9.1 AdaBoost

Algorithm 1 Ada Boost

Input $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, a week learning algorithm $\mathcal{A}$
Initialize $D_1(i) = \frac{1}{n}$
for $t = 1, 2, ..., T$ **do**
  Using weak learner $\mathcal{A}$, train a weak classifier h_t on D_t, $h_t(x_i) \in \{1, -1\}$
  $\epsilon_t := \sum_i D_t(i) I[y_t \neq h_t(i)]$
  $\gamma_t = 1 - 2\epsilon_t$
  $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$
  $Z_t := \sum_{i=1}^{n} D_t(i) \exp(-\alpha_t y_i h_t(x_i))$
  $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t h_i h_t(x_i))}{Z_t}$
end for
Output $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

It should be noted that the algorithm is dealing with a binary classification problem, where ϵ_t and α_t represent the error rate and the accuracy of the current classifier respectively, and Z_t is a normalization factor. Intuitively, the more accurate a weak classifier is, the more proportion it will occupy in the output. Ada boost is trying to minimize the Exponential loss:

$$L_{exp} = \frac{1}{n} \sum_{i=1}^{n} \exp(-y_i f(x_i))$$

which is an upper bound of 0-1 classification loss.

Lemma 9.1 $\alpha_t = \arg\min_\alpha Z_t = \arg\min_\alpha \sum_{i=1}^{n} D_t(i) \exp(-\alpha_t y_i h_t(x_i))$

Proof:

$$\min_\alpha Z_t = \min_\alpha \sum_{i=1}^{n} D_t(i) \exp(-y_i h_t(x_i))^\alpha$$

$$\sum_{i=1}^{n} D_t(i) \exp(-y_i h_t(x_i))^\alpha = (\epsilon_t) \exp \alpha + (1 - \epsilon_t) \exp(-\alpha) \geq 2\sqrt{\epsilon_t(1 - \epsilon_t)}$$

According to the AM-GM Inequalitythe quality holds if and only if:

$$\epsilon_t \exp(\alpha) = (1-\epsilon_t)\exp(-\alpha) \Leftrightarrow \alpha = \frac{1}{2}\ln\frac{1+\gamma_t}{1-\gamma_t}$$

■

Lemma 9.2 $\prod_{t=1}^{T} Z_t = \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i f(x_i)) = \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i\sum_{i=1}^{T}\alpha_t h_t(x_i))$

Proof:

$$\begin{aligned} Z_T &= \sum_{i=1}^{n} D_T(i)\exp(-y_i\alpha_T h_T(x_i)) \\ &= \sum_{i=1}^{n}\frac{D_{T-1}(i)\exp(-y_i\alpha_{T-1}h_{T-1}(x_i))}{Z_{T-1}}\exp(-y_i\alpha_T h_T(x_i)) \end{aligned}$$

Therefore, we can say

$$\begin{aligned} Z_{T-1}Z_T &= \sum_{i=1}^{n} D_{T-1}(i)\exp(-y_i\alpha_{T-1}h_{T-1}(x_i) - y_i\alpha_T h_T(x_i)) \\ &= \sum_{i=1}^{n}\frac{D_{T-2}(i)\exp(-y_i\alpha_{T-2}h_{T-2}(x_i))}{Z_{T-2}}\exp(-y_i\alpha_{T-1}h_{T-1}(x_i) - y_i\alpha_T h_T(x_i)) \\ &\cdots\cdots \end{aligned}$$

Repeat the above process, and we eventually have

$$\begin{aligned} \prod_{t=1}^{T} Z_t &= \sum_{i=1}^{n} D_1(i)\exp(-y_i\sum_{t=1}^{T}\alpha_t h_t(x_i)) \\ &= \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i\sum_{t=1}^{T}\alpha_t h_t(x_i)) \end{aligned}$$

■

Lemma 9.3 *Assume in Ada Boost algorithm, $\gamma_t \geq \gamma > 0, \forall t \in [T]$. Then*

$$P_s(yf(x) \leq 0) = \frac{1}{n}\sum_{i=1}^{n} I[y_i f(x_i) \leq 0] \leq (1-\gamma^2)^{T/2}$$

Proof: Because that the exponential loss is the upper bound of 0-1 classification loss, we only need to prove that the exponential loss converges at this rate. According to lemma 2, the exponential loss equals to $\prod Z_t$

$$\begin{aligned} Z_t &= (2(1-\gamma_t)\epsilon_t)^{\frac{1}{2}} \\ &\leq 2\sqrt{\frac{1-\gamma}{2}(1-\frac{1-\gamma}{2})} \\ &= \sqrt{1-\gamma^2} \end{aligned}$$

Therefore we have

$$\begin{aligned} P_s(yf(x) \leq 0) &\leq \prod_{t=1}^{T} \sqrt{1-\gamma^2} \\ &= (1-\gamma^2)^{T/2} \end{aligned}$$

■

Based on the assumption, which is easy to satisfy, the training error decreases to 0 after finite steps. $T = O(\ln n)$

Now let's consider the performance of h_t on D_{t+1}, i.e. $\sum_{i=1}^{n} D_{t+1}(i) I[y_i \neq h_t(x_i)]$

Lemma 9.4

$$\sum_{i=1}^{n} D_{t+1}(i) I[y_i \neq h_t(x_i)] = \frac{1}{2}$$

Proof:

$$\begin{aligned} \sum_{i=1}^{n} D_{t+1}(i) I[y_i \neq h_t(x_i)] &= \sum_{i=1}^{n} \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t} I[y_i \neq h_t(x_i)] \\ &= \frac{\sum_{i=1}^{n} D_t(i) exp(\alpha_t) I[y_i \neq h_t(x_i)]}{\sum_{i=1}^{n} D_t(i) exp(\alpha_t) I[y_i \neq h_t(x_i)] + \sum_{i=1}^{n} D_t(i) exp(-\alpha_t) I[y_i = h_t(x_i)]} \\ &= \frac{\epsilon_t exp(\alpha_t)}{\epsilon_t exp(\alpha_t) + (1-\epsilon_t) exp(-\alpha_t)} \end{aligned}$$

The selection of α_t is to minimize z_t and so we have $\epsilon_t exp(\alpha_t) = (1-\epsilon_t) exp(-\alpha_t)$, thus

$$\sum_{i=1}^{n} D_{t+1}(i) I[y_i \neq h_t(x_i)] = \frac{1}{2}$$

■

9.2 Online learning

Online machine learning is a method of machine learning in which data becomes available in a sequential order and is used to update the best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once. [1]

9.2.1 Follow the leader

Assume that there are T rounds and N experts. At the tth round, every expert $i \in$[N] makes a prediction, denoted as $y_{t,i}^{\sim} \in \{0,1\}$, then the learner makes prediction $\tilde{y_t}$. After that, the adversary reveals $y_t \in \{0,1\}$. The objective is to minimize the accumulate loss (here it is 0-1loss) $\sum_{t=1}^{T} I[\tilde{y_t} \neq y_t]$.

Our simple intuition might lead us to the "follow the leader" strategy, which means at the tth round, we make the same prediction as best expert, that is, the expert who makes the smallest number of mistakes in the former t - 1 rounds. However, this strategy gives poor performance both theoretically and practically.

References

[1] https://en.wikipedia.org/wiki/Online_machine_learning