

Foundations of Machine Learning — Homework Assignment 3

Anirudhan J Rajagopalan
N18824115
ajr619

December 09, 2015

A. Boosting-type Algorithm

1. Bound of $1_{u \leq 0}$ Proof of convexity and differentiability

Given: $\phi_p(u) = \max((1+u)^p, 0)$

To prove: 1. Function $\phi_p(u)$ is convex and differentiable

And 2. $\forall u \in \mathbb{R}$ and $p > 1, 1_{u \leq 0} \leq \phi_p(-u)$

Proof of $\forall u \in \mathbb{R}$ and $p > 1, 1_{u \leq 0} \leq \phi_p(-u)$ There are three cases here:

When $u = 0$

$$\begin{aligned} 1_{u \leq 0} &= 1 \\ \phi_p(-u) &= \max((1-0)^p, 0) = 1 = 1_{u \leq 0} \end{aligned}$$

When $u < 0$

$$\begin{aligned} 1_{u \leq 0} &= 1 \\ \phi_p(-u) &= \max((1-u)^p, 0) \\ &= \max((1+u)^p, 0) \text{ Since } u \text{ is negative, } -u \text{ is positive} \\ &> 1 \\ &> 1_{u \leq 0} \end{aligned}$$

When $u > 0$

$$\begin{aligned} 1_{u \leq 0} &= 0 \\ \phi_p(-u) &= \max((1-u)^p, 0) \\ &= \begin{cases} 0, & \text{if } p \text{ is odd} \\ (1-u)^p, & \text{if } p \text{ is positive} \end{cases} \\ &\geq 0 \quad \forall \quad \mathbb{R} \geq 1_{u \leq 0} \end{aligned}$$

Hence proved

Proof of Convexity and Differentiability: The function can be written as a piecewise function based on the value of p :

When p is even (2 and higher value even numbers)

$$\phi_p(u) = (1+u)^p$$

Since $p > 1$, the function is differentiable.

$$\begin{aligned} \phi_p'(u) &= p(1+u)^{p-1} \\ \phi_p''(u) &= (p-1)p(1+u)^{p-2} \\ \therefore p > 1, \phi_p''(u) &> 0 \end{aligned}$$

Since the double derivative is greater than zero, the function is convex and differentiable.

When p is odd (1 and higher value odd numbers) the function can be defined using the piecewise function

$$\phi_p(x) = \begin{cases} 0, & \text{if } u \leq -1 \\ (1+u)^p, & \text{otherwise} \end{cases}$$

The function is piecewise continuous and differentiable. We have to check that the function is differentiable at $u = -1$ to show that the function is differentiable and continuous. We should also check the double derivative to show that the function is convex.

The first derivative can be found using limits.

$$\begin{aligned} \lim_{h \rightarrow 0+} \frac{\phi_p(-1+h) - \phi_p(-1)}{h} &= \lim_{h \rightarrow 0+} \frac{(1+(-1+h))^p - (1-1)^p}{h} \\ &= \lim_{h \rightarrow 0+} \frac{h^p - 0}{h} \\ &= \lim_{h \rightarrow 0+} h^{p-1} \\ &= 0 \end{aligned}$$

Also

$$\begin{aligned} \lim_{h \rightarrow 0-} \frac{\phi_p(-1+h) - \phi_p(-1)}{h} &= \lim_{h \rightarrow 0+} \frac{(1+(-1+h))^p - (1-1)^p}{h} \\ &= \lim_{h \rightarrow 0-} \frac{h^p - 0}{h} \\ &= \lim_{h \rightarrow 0-} h^{p-1} \\ &= 0 \end{aligned}$$

Since both the limit values are equal, the function is differentiable. Therefore:

$$\phi'_p(x) = \begin{cases} 0, & \text{if } u \leq -1 \\ p(1+u)^{p-1}, & \text{otherwise} \end{cases}$$

For showing the function is convex, we find the double differentiable using limits for ϕ'_p and show that it is non negative

$$\begin{aligned}
\lim_{h \rightarrow 0+} \frac{\phi'_p(-1+h) - \phi'_p(-1)}{h} &= \lim_{h \rightarrow 0+} \frac{p(1 + (-1+h))^{p-1} - 0}{h} \\
&= \lim_{h \rightarrow 0+} \frac{p(h)^{p-1}}{h} \\
&= \lim_{h \rightarrow 0+} p(h)^{p-2} \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\lim_{h \rightarrow 0-} \frac{\phi'_p(-1+h) - \phi'_p(-1)}{h} &= \lim_{h \rightarrow 0-} \frac{p(1 + (-1+h))^{p-1} - 0}{h} \\
&= \lim_{h \rightarrow 0-} \frac{p(h)^{p-1}}{h} \\
&= \lim_{h \rightarrow 0-} p(h)^{p-2} \\
&= 0
\end{aligned}$$

Since the left and right derivatives are equal, the function is ϕ'_p is differentiable. Which implies that the original function ϕ_p is double differentiable.

$$\phi''_p(x) = \begin{cases} 0, & \text{if } u \leq -1 \\ (p-1)p(1+u)^{p-2}, & \text{otherwise} \end{cases}$$

Since in both cases the double differentiable is non negative, the function is convex and differentiable.

2. Derive boosting type algorithm

We can build an algorithm similar to adaboost using coordinate descent. The algorithm and the explanation are given below.

The objective function is defined as

$$F(\alpha) = \sum_{i=1}^m \phi_p(-y_i f_T(x_i))$$

Where

$$F_T(x_i) = \sum_{t=1}^T \alpha_t h_t(x_i)$$

and the weighted distribution of the sample is

$$D_t(i) \leftarrow \frac{1}{m} \frac{\phi'_p(-y_i f_t(x_i))}{\sum_{i=1}^m \phi'_p(-y_i f_t(x_i))}$$

Algorithm 1 Boosting type algorithm

```

1:  $S = ((x_1, y_1), (x_2, y_2), \dots, (x_m, y_m))$ 
2: for  $i \leftarrow 1$  to  $m$  do
3:    $D_1(i) \leftarrow \frac{1}{m}$ 
4: for  $i \leftarrow 1$  to  $T$  do
5:    $h_t \leftarrow$  base classifier in  $H$  with small error  $\epsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]$ 
6:    $\alpha_t \leftarrow \eta$  obtained by searching non-linear equation of  $\eta$ 
7:    $f_t \leftarrow f_{t-1} + \alpha_t h_t$ 
8: for  $i \leftarrow 1$  to  $m$  do
9:    $D_t(i) \leftarrow \frac{1}{m} \frac{\phi'_p(-y_i f_t(x_i))}{\sum_{i=1}^m \phi'_p(-y_i f_t(x_i))}$ 
return  $h = \text{sgn}(f_T)$ 

```

The weighted distribution is initialized to $\frac{1}{m}$ initially.

Let e_t denote the unit vector corresponding to the coordinate in \mathbb{R}^n and let α_{t-1} denote the vector based on the $(t-1)$ coefficients.

$$\alpha_{t-1} = \begin{cases} (\alpha_1, \alpha_2, \dots, \alpha_{t-1}, 0, 0, \dots, 0)^T & \text{if } (t-1) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Direction

Coordinate descent selects the direction e_t that minimizes the directional derivative.

$$e_t = \underset{\eta=0}{\operatorname{argmin}_t} \left[\frac{dF(\alpha_{t-1} + \eta e_t)}{d\eta} \right]$$

where

$$\begin{aligned} F(\alpha_{t-1} + \eta e_t) &= \sum_{i=1}^m \phi_p \left(-y_i \sum_{j=1}^{t-1} (\alpha_j + \eta e_t) h_j \right) \\ &= \sum_{i=1}^m \phi_p \left(-y_i \sum_{j=1}^{t-1} \alpha_j h_j(x_i) - \eta y_i h_t(x_i) \right) \end{aligned}$$

Also,

$$\begin{aligned}
\frac{dF(\alpha_{t-1} + \eta e_t)}{d\eta} &= - \sum_{i=1}^m y_i h_t(x_i) \cdot \phi'_p \left(-y_i \sum_{j=1}^{t-1} \alpha_j h_j(x_i) \right) \\
&= \sum_{i=1}^m y_i h_t(x_i) \cdot D(i) \cdot \left(m \sum_{k=1}^m \phi'_p(-y_k f_t(x_k)) \right) \\
&= - \left(\sum_{y_i h_t = -1} D(i) + \sum_{y_i h_t = 1} D(i) \right) \left(m \sum_{k=1}^m \phi'_p(-y_k f_t(x_k)) \right) \\
&= - ((1 - \epsilon_t - \epsilon_t)) \left(m \sum_{i=1}^m \phi'_p(-y_i f_t(x_i)) \right) \\
&= (2\epsilon_t - 1) \left(m \sum_{i=1}^m \phi'_p(-y_i f_t(x_i)) \right) \\
&\propto (2\epsilon_t - 1)
\end{aligned}$$

Because of the above equation, it can be seen that the hypothesis chosen by the algorithm minimizes the mis-classification error.

Step

We can obtain the step by minimizing the derivative of dF for η

$$1 = 2$$

Generalization Bound

A family of functions (H) taking values in $\{+1, -1\}$ with VC-dimension d for any $\delta > 0$ with probability atleast $1 - \delta$, it holds for any $h \in \text{conv}(H)$:

$$R(h) \leq \hat{R}_p(h) + \frac{2}{p} \sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

The above expression is true for all ensemble methods (Refer Corollary 6.2, page 133 of Foundations of Machine Learning, Mehryar Mohri et al.) it will be true for the algorithm described above. Also we cannot have an empirical bound on $\hat{R}_p(h)$ in terms of misclassification error as there is not closed form expression of α_t

C. Randomized Halving

1. Psuedo code

Algorithm 2 Randomized Halving

```

1:  $H_1 \leftarrow H$ 
2: for  $t \leftarrow 1$  to  $T$  do
3:    $RECEIVE(x_t)$ 
4:    $r_t \leftarrow \frac{\sum_{i: y_{t,i}=1} 1}{|H_t|}$ 
5:    $p_t \leftarrow 1$ 
6:   if  $r_t \leq \frac{3}{4}$  then
7:      $p_t \leftarrow \lceil \frac{1}{2} \log_2 \frac{1}{1-r_t} \rceil$ 
8:    $\hat{y}_t \leftarrow GetRandomNumberWithProbability([1, 0], [p_t, 1 - p_t])$ 
9:    $RECEIVE(y_t)$ 
10:  if  $\hat{y}_t \neq y_t$  then
11:     $H_{t+1} \leftarrow \{c \in H_t : c(x_t) = y_t\}$ 
return  $H_{T+1}$ 

```

2. Prove $\forall t \geq 1, E[\mu] \leq \frac{\phi_t - \phi_{t+1}}{2}$

Given: Potential function: $\phi_t = \log_2 |H_t|$ and $\mu_t = 1_{y_t \neq \hat{y}_t}$

Proof:

We are only considering the case when the predicted value \hat{y}_t is not equal to the received value y_t . The value of expectation can be written as

$$\begin{aligned} E[\mu_t] &= p_t * 1 + (1 - p_t) * 0 \\ &= p_t * 1 \end{aligned}$$

The probability of predicting 1 by the randomized algorithm is the probability of making a mistake since we are only considering the cases in which we make mistakes ($\mu_t = 1_{y_t \neq \hat{y}_t}$)

Therefore,

$$\begin{aligned} E[\mu_t] &= p_t \\ &= \lceil \frac{1}{2} \log_2 \frac{1}{1-r_t} \rceil 1_{r_t \leq \frac{3}{4}} + 1_{r_t > \frac{3}{4}} \\ &\leq \lceil \frac{1}{2} \log_2 \frac{1}{1-r_t} \rceil \end{aligned}$$

Since, the Expectation will be 1 when $r_t > \frac{3}{4}$ which corresponds to the maximum expectation here, we can upper bound the expectation by using $r_t \leq \frac{3}{4}$ as $\lceil \frac{1}{2} \log_2 \frac{1}{1-r_t} \rceil$ equals 1 when $r_t = \frac{3}{4}$

Let E_1, E_0, E_t denote the number of experts predicting 1, 0 and the total number of experts in a round t .

$$\begin{aligned}
E[\mu_t] &\leq \left\lfloor \frac{1}{2} \log_2 \frac{1}{1-r_t} \right\rfloor \\
&= \left\lfloor \frac{\log_2 \frac{1}{1-r_t}}{2} \right\rfloor \\
&= \left\lfloor \frac{\log_2 \frac{1}{1-r_t}}{2} \right\rfloor \\
&= \left\lfloor \frac{\log_2 \frac{1}{1-\frac{|E_1|}{|H_t|}}}{2} \right\rfloor \\
&= \left\lfloor \frac{\log_2 \frac{|H_t|}{|H_t|-|E_1|}}{2} \right\rfloor \\
&= \left\lfloor \frac{\log_2 \frac{|H_t|}{|E_0|}}{2} \right\rfloor \\
&= \left\lfloor \frac{\log_2 |H_t| - \log_2 |E_0|}{2} \right\rfloor \\
&= \left\lfloor \frac{\log_2 |H_t| - \log_2 |H_{t+1}|}{2} \right\rfloor \\
&= \left\lfloor \frac{\phi_t - \phi_{t+1}}{2} \right\rfloor \\
\therefore E[\mu_t] &\leq \frac{\phi_t - \phi_{t+1}}{2}
\end{aligned}$$

3. Expected number of mistakes.

Given: N be the total number of experts at the beginnning of the iterations (denoted by H_1). Since we are considering a relizable scenario at the end of the algorithm the number of experts should be atleast one. So $H_T = 1$.

To Prove: The expected number of mistakes made by Randomized Halving is at most $\frac{1}{2} \log_2 N$

Proof:

Lets consider the total expectation of mistakes of the Randomized Halving

algorithm over T iterations.

$$\begin{aligned}
E[\mu_T] &\leq \sum_{t=1}^T \frac{\phi_t - \phi_{t+1}}{2} \\
&\leq \frac{(\phi_{H_1} - \phi_{H_2}) + (\phi_{H_2} - \phi_{H_3}) \cdots (\phi_{H_{T-2}} - \phi_{H_{T-1}}) + (\phi_{H_{T-1}} - \phi_{H_T})}{2} \\
&\leq \frac{\phi_{H_1} - \phi_{H_T}}{2} \\
&\leq \frac{\phi_N - \phi_T}{2} \\
&\leq \frac{\phi_N}{2} \\
&\leq \frac{1}{2} \log_2 N
\end{aligned}$$

Hence proved.

Here $\phi_T = 0$ because the number of experts at line T is 1. Therefore, log of T will be zero.

4. [Bonus Question]

As we have seen in the previous answer, the mistakes made by the randomized algorithm is bounded by $\frac{1}{2} \log_2 N$. This upper bound is dependent only on the number of initial experts, N . Therefore any randomized algorithm that is dependent on the opinion of the experts to generate its predictions will have similar upper bound of $\lfloor \frac{1}{2} \log_2 N \rfloor$. The floor function is used as mistakes are natural numbers.