Adaboost의 가중치 이해하기

Adaboost 첫 제안 : https://www.sciencedirect.com/science/article/pii/S002200009791504X

위 paper의 저자인 Yoav Freund와 Robert Schapire는 Adaboost를 개발한 공로로 괴델상을 수상하였다.

가중치 업데이트 과정은 Rojas, R. (2009)의 http://www.inf.fu-berlin.de/inst/ag-ki/adaboost4.pdf 와 머신러닝과 통계학습의 교과서라 불리는 ESL (Elementary Statistical Learning), 그리고 위키백과의 자료를 바탕으로 풀어 설명하도록 하겠다.

Adaboost는 Loss function으로 exponential loss를 사용하며, 다양한 Loss function을 사용할 수 있는 Gradient boosting model에 포함되는 모델로 볼 수 있다.

하지만, Gradient Boosting은 개별 데이터 (각 row)에 가중치를 주지 않는데, Adaboost는 weak model들이 분류에 실패를 많이한 데이터일수록 더 높은 가중치를 준다.

우선 binary target 변수 (1 or -1, True or False) 를 맞추는 adaboost Classifier의 가중치 공식을 유도해보자.

각 input 값 x_i에 대응되는 label $y_i \in \{-1, 1\}$을 가지고 있는 데이터 $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ 에 대해 각각의 weak classifier를 k_j 라 하자. L개의 weak classifier (일반적으로 decision tree)가 존재한다고 할 때, j번째 weak classifier를 다음과 같이 표현할 수 있다.

$$k_j(x_i) \in \{-1, 1\}$$

그리고, $m-1$번째까지 업데이트된 Boosting Model C_{m-1}을 다음과 같이 정의할 수 있다. (GAM : Generative Additive Model)

$$C_{m-1}(x_i) = \alpha_1 k_1(x_i) + \cdots + \alpha_{m-1} k_{m-1}(x_i)$$

어렵게 생각하지 말고, i번째 row(data)에 대해 첫 번째 weak classifier (decision tree라고 하겠다.)의 가중치가 0.1이고 예측값이 True, 두번째 모델의 가중치는 0.4이고 예측값이 False, 세번째 트리는 0.5의 가중치를 가지고 예측값이 True 인 경우,

$$C_3(x_i) = 0.1 \cdot 1 + 0.4 \cdot (-1) + 0.5 \cdot 1 = 0.2$$

라고 할 수 있는 것이다. 이 것을 sign함수에 넣으면 $C_3(x_i)$는 x_i를 True라고 분류하게 된다.

이제 weak classifier $k_m(x_i)$를 추가하여 $C_m(x_i)$으로 기존의 분류기를 업데이트 하는 과정은 아래와 같다.

$$C_m(x_i) = C_{m-1}(x_i) + \alpha_m k_m(x_i)$$

그렇다면, 약한 분류기 k_m은 무엇이 되는 것이 최선인지, 그 분류기의 가중치 α_m은 얼마가 되어야 하는지 결정을 해야 한다. 이 때, Adaboost에서 사용하는 Loss인 Exponential Loss를 이용하는데, 그 Loss의 공식은 다음과 같다.

y : label 값 (1 or -1, True or False) $f(x)$: 예측값

$$L(y, f(x)) = exp(-yf(x)) = e^{-yf(x)}$$

Boosting Model $C_m(x_i)$의 총 오류 (cost function)를 E 라고 정의할 때, E를 loss function을 이용해 다음과 같이 표현할 수 있다.

(http://www.cs.man.ac.uk/~stapenr5/boosting.pdf 참조)

$$E = \sum_{i=1}^{N} e^{-y_i C_m(x_i)}$$

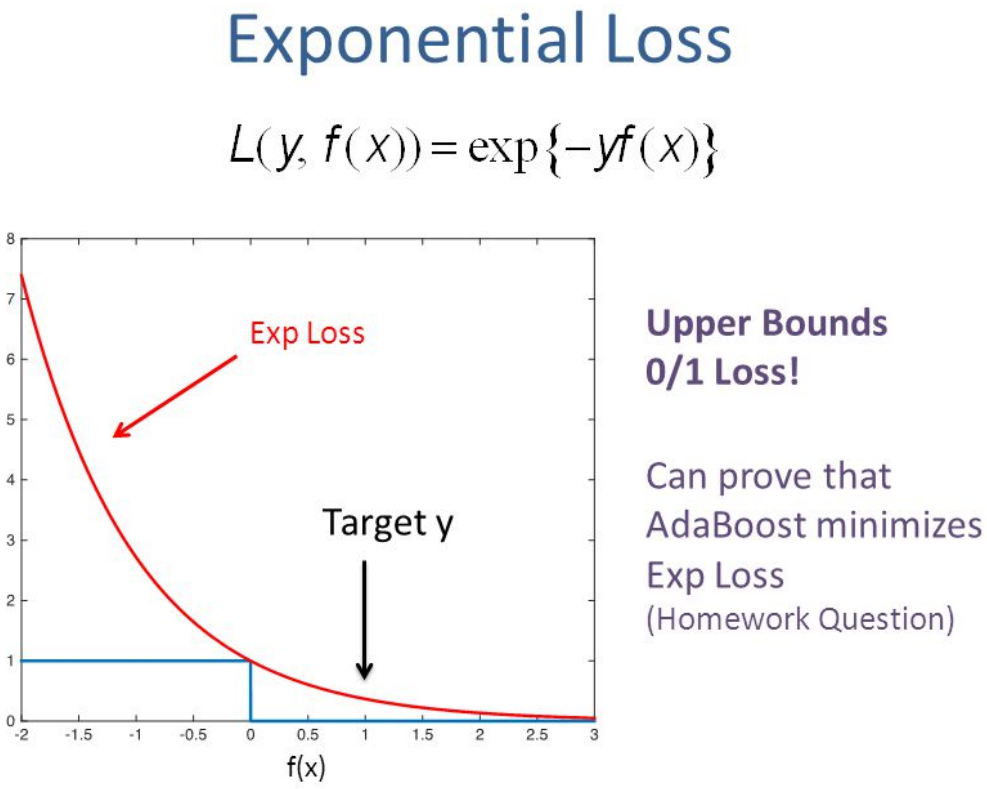
식은 복잡해 보이지만, 각 데이터 포인트마다 exponential loss를 구한 뒤, 이를 모두 더하는 것이다. 식은 복잡하지만 예시로 이해하면 쉽다. $N = 3$으로 데이터가 3개이고, 부스팅 모델이 예측한 결과 $C_m(x_i)$ 와 label 값 y_i이 다음 표와 같다고 해보자.

index	y_true	y_predict
1	1	1
2	1	-1
3	-1	-1

모델이 첫 번째와 세 번째 데이터는 올바르게 예측했지만 두 번째 데이터는 틀렸다. 이를 식으로 표현하면 다음과 같다.

$$E = e^{-1 \cdot 1} + e^{-1 \cdot (-1)} + e^{-(-1) \cdot (-1)} = e^1 + 2 \cdot e^{-1}$$

맞은 데이터에는 e^{-1}의 error (loss)가, 틀린 데이터에는 e^1의 error가 계산되었다. 분명히 맞은 데이터인데 왜 error가 존재하는지 직관적으로 이해가 되지 않을 수도 있다. 다음 그림을 보자.



우리가 분류 모델에서 직관적으로 사용하고 있는 loss는 0/1 loss 이다. 맞으면 0, 틀리면 1의 손해를 가지는 것이다. 하지만, exponential loss는 맞아도 e^{-1}, 틀리면 e^{1}의 손해를 가지도록 한다.

이러한 Loss를 이용하여, 데이터의 가중치 개념을 포함한 총 오류 E 를 정의해보자. 위의 예제에서는 각 데이터의 가중치가 모두 같은, 즉 모두 1 (1/3 이라고 할 수도 있습니다)인 경우의 E 값이 $e^{1} + 2 \cdot e^{-1}$ 이었다. 이번에는, 각각의 데이터 가중치가 [1,2,3]이라고 해보자. 이 때의 총 오류 E 를 계산하는 방법은 다음과 같다.

$$E = 1 \cdot e^{-1 \cdot 1} + 2 \cdot e^{-1 \cdot (-1)} + 3 \cdot e^{-(-1) \cdot (-1)} = 2 \cdot e^{1} + 4 \cdot e^{-1}$$

이를 일반화하여 표현하면 다음과 같다.

$$E = \sum_{i=1}^{N} w_i^{(m)} e^{-y_i C_m(x_i)}$$

$w_i^{(m)}$에서 i는 데이터의 index를 나타내고 m는 m번째 가중치를 나타낸다. **Adaboost에서는 각 데이터 포인트의 Loss를 그 데이터 포인트의 가중치에 곱하여 계속 업데이트한다.** (밑의 문제 증명으로 해결해야 하는데 증명해주실 분~)

- Show that if we assign cost a to misses and cost b to hits, where $a > b > 0$, we can rewrite such costs as $a = c^{d}$ and $b = c^{-d}$ for constants c and d. That is, exponential loss costs of the type e^{α_m} and $e^{-\alpha_m}$ do not compromise generality.

$-y_i C_{m-1}(x_i)$ 라는 표현은 얼핏 보면 복잡해보이지만, 분류기가 잘 분류했으면 -1의 값을 가지고, 잘못 분류했으면 1의 값을 가지게 된다. 계속 맞는 데이터는 가중치에 e^{-1}이 계속 곱해지면서 낮은 가중치를 가질 것이고, 계속 틀리는 데이터일 수록 가중치에 e^{1} 이 계속 곱해지면서 높은 가중치를 가지게 될 것이다.

N개의 데이터에 대해, 초기에 동일한 가중치를 준 다음, weak classifier가 분류에 성공한 데이터는 다음 classifier가 중요하게 고려하지 않도록 e^{-1}의 가중치를 업데이트 해주고, 틀린 데이터는 중요하게 고려하도록 e^{1}의 가중치를 주려고 한다. 이를 다음과 같이 표현할 수 있다.

$m > 1$ 에 대하여 $w_i^{(1)} = 1$, $w_i^{(m)} = e^{-y_i C_{m-1}(x_i)}$

다시 위의 문제로 돌아가서, 어떤 α_m 과 k_m 을 사용 해야할지 구해보도록 하자.

$$C_m(x_i) = \alpha_1 k_1(x_i) + \cdots + \alpha_m k_m(x_i)$$

임을 이용해서 총 오류 E를 다시 표현해보면,

$$E = \sum_{i=1}^{N} w_i^{(m)} e^{-y_i C_m(x_i)} = \sum_{i=1}^{N} w_i^{(m)} e^{-y_i \alpha_m k_m(x_i)}$$

이제, 이 식을 이용하여 총 오류를 최소화하는 모델별 가중치 α_m 을 찾아보자.

$y_i k_m(x_i) = 1$ 인 잘 분류된 데이터와 $y_i k_m(x_i) = -1$ 인 잘못 분류된 데이터로 위 식을 쪼개보자.

$$E = \sum_{y_i = k_m(x_i)} w_i^{(m)} e^{-y_i \alpha_m k_m(x_i)} + \sum_{y_i \neq k_m(x_i)} w_i^{(m)} e^{-y_i \alpha_m k_m(x_i)}$$

$$= \sum_{y_i = k_m(x_i)} w_i^{(m)} e^{-\alpha_m} + \sum_{y_i \neq k_m(x_i)} w_i^{(m)} e^{\alpha_m}$$

$$= e^{-\alpha_m} \times \sum_{y_i = k_m(x_i)} w_i^{(m)} + e^{\alpha_m} \times \sum_{y_i \neq k_m(x_i)} w_i^{(m)}$$

이제 위 식을 기준으로 best α_m 을 찾을 수 있다.

총 오류 E 가 최소가 되도록 하는 α_m을 편미분을 통해 구해보자.

$$E = \sum_{y_i = k_m(x_i)} w_i^{(m)} e^{-\alpha_m} + \sum_{y_i \neq k_m(x_i)} w_i^{(m)} e^{\alpha_m}$$

$$\frac{\partial E}{\partial \alpha_m} = -e^{-\alpha_m} \sum_{y_i = k_m(x_i)} w_i^{(m)} + e^{\alpha_m} \sum_{y_i \neq k_m(x_i)} w_i^{(m)} = 0$$

위 방정식을 만족시키는 α_m을 구하기 앞서, 가중치를 고려한 모델의 오분류율을 다음과 같이 정의하자.

$$\epsilon_m = \frac{\sum_{y_i \neq k_m(x_i)} w_i^{(m)}}{\sum_{y_i \neq k_m(x_i)} w_i^{(m)} + \sum_{y_i = k_m(x_i)} w_i^{(m)}}$$

이 때, $\sum_{y_i \neq k_m(x_i)} w_i^{(m)} = W_e$, $\sum_{y_i = k_m(x_i)} w_i^{(m)} = W_c$라 하자.

$$\frac{\partial E}{\partial \alpha_m} = -W_c e^{-\alpha_m} + W_e e^{\alpha_m} = 0$$

으로 위 방정식을 간략하게 표현할 수 있고, 양변에 e^{α_m}을 곱해 식을 다음과 같이 간략화할 수 있다.

$$\frac{\partial E}{\partial \alpha_m} = -W_c + W_e e^{2\alpha_m} = 0$$

따라서, 최적의 α_m은 다음과 같다.

$$\alpha_m = \frac{1}{2} ln \left(\frac{W_c}{W_e} \right)$$

이 때, $\epsilon_m = \frac{W_e}{W_c + W_e}$, $W = W_c + W_e$ 이라 하면

$$\alpha_m = \frac{1}{2} ln \left(\frac{W - W_e}{W_e} \right) = \frac{1}{2} ln \left(\frac{1 - \epsilon_m}{\epsilon_m} \right)$$

으로 정리할 수 있다.