TTIC 31020: Introduction to Machine Learning
Autumn 2022

(collaborated with Haichuan Wang)

Problem Set #4

1 Boosting

In stepwise fit-forward (least squares) regression, in each iteration a simple regressor is fit to the residuals obtained by the ensemble model up to that iteration. As a result, it is easy to see that *after* this regressor is added, the new residuals are uncorrelated with its predictions, due to a general property of least squares regression. *Advice:Make sure you can actually show it (no need to turn in the proof).*

We will now investigate a similar phenomenon that occurs with weak classifiers in AdaBoost. Here, we assume that the weights are normalized after each update, so that

$$\sum_i W_i^{(t)} = 1$$

for each boosting round t.

Problem 1 **[15 points]**
Consider an ensemble classifier $H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$ constructed by T rounds of AdaBoost on N training examples. Now we add next classifier h_{T+1} to the ensemble, by minimizing the training error weighted by $W_1^{(T)}, \ldots, W_N^{(T)}$, compute α_{T+1}, and update the weights. Show that the training error of the just added h_{T+1} (note: not the error of H_{T+1}) weighted by the *updated* weights $W_1^{(T+1)}, \ldots, W_N^{(T+1)}$, is exactly 1/2.

Now, with that fact in mind, is it possible that AdaBoost would select the same classifier again in the immediately following round, i.e., can we have $h_t = h_{t+1}$ for some t? Can we have $h_{t+k} = h_t$ for some $k > 1$? Explain why or why not.

End of problem 1

1)

- Let $A^{(T+1)} = \{ i \mid h_{T+1}(x_i) \neq y_i \}$,

training error of h_{T+1}

$$= \sum_{i \in A^{(T+1)}} W_i^{(T+1)}$$

$$= \sum_{i \in A^{(T+1)}} \frac{W_i^{(T)} \cdot e^{\overbrace{-y_i \alpha_{T+1} h_{T+1}(x_i)}^{= -1,\ \text{misclassified}}}}{\sum_{i=1}^{N} W_i^{(T)} \cdot e^{-y_i \alpha_{T+1} \cdot h_{T+1}(x_i)}}$$

$$= \frac{\sum_{i \in A^{(T+1)}} W_i^{(T)} \cdot e^{\alpha_{T+1}}}{\underbrace{\sum_{i \in A^{(T+1)}} W_i^{(T)} \cdot e^{\alpha_{T+1}}}_{\text{incorrect classification}} + \underbrace{\sum_{i \notin A^{(T+1)}} W_i^{(T)} \cdot e^{-\alpha_{T+1}}}_{\text{correct classification}}}$$

divide α^{T+1}

$$\frac{\sum_{i \in A^{(T+1)}} W_i^{(T)}}{\sum_{i \in A^{(T+1)}} \cdot W_i^{(T)} + \sum_{i \notin A^{(T+1)}} W_i^{(T)} \cdot e^{-2\alpha_{T+1}}}$$

- $$\alpha_{T+1} = \frac{1}{2} \log \frac{1 - \varepsilon_{T+1}}{\varepsilon_{T+1}}$$

$$= \frac{1}{2} \log \frac{1 - \sum_{i \in A^{(T+1)}} W_i^{(T)}}{\sum_{i \in A^{(T+1)}} W_i^{(T)}}$$

$$= \frac{1}{2} \log \frac{\sum_{i \notin A^{(T+1)}} W_i^{(T)}}{\sum_{i \in A^{(T+1)}} W_i^{(T)}}$$

- plug into training error:

$$= \frac{\sum_{i \in A^{(T+1)}} W_i^{(T)}}{\sum_{i \in A^{(T+1)}} W_i^{(T)} + \sum_{i \notin A^{(T+1)}} W_i^{(T)} \cdot \frac{\sum_{i \in A^{(T+1)}} W_i^{(T)}}{\sum_{i \notin A^{(T+1)}} W_i^{(T)}}}$$

$$= \frac{1}{2}$$

2)

- We can't have $h_{t+1} = h_t$
  - If $h_{t+1} = h_t$,
  - $A^{(t+1)} = \{i \mid h_{t+1}(x_i) \neq y_i\}$

    $= \{i \mid h_t(x_i) \neq y_i\}$

    $= A^{(t)}$
  - Then, $\varepsilon_{t+1} = \sum_{i \in A^{(t+1)}} W_i^{(t)}$

    $= \sum_{i \in A^{(t)}} W_i^{(t)}$

    $= \frac{1}{2}$ (by the previous result)
  - Thus, $\alpha_{t+1} = \frac{1}{2} \log \frac{1-\varepsilon_{t+1}}{\varepsilon_{t+1}} = \frac{1}{2} \log 1 = 0$.

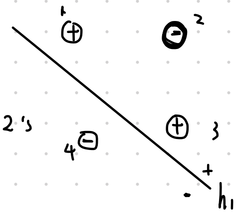
    So, we are not adding h_{t+1} to our combined classifier H.

[With help from Haichuan Wang]

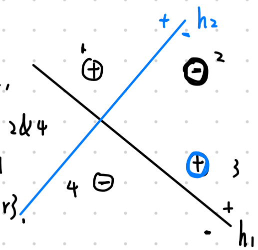
- We can have $h_{t+k} = h_t$.

  For example, when the data are not linearly separable:

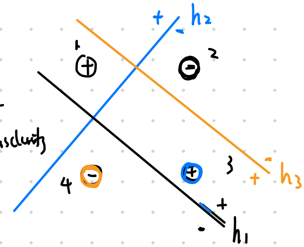
  - First, choose h_1 that misclassifies point 2. This will increase point 2's weight.



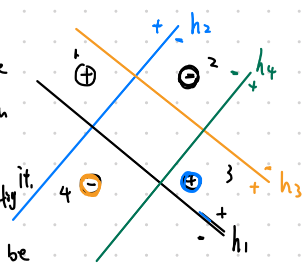
  - Second, to not misclassify point 2, we need to classify 2 & 4 together, which will misclassify either 1 or 3. Say we choose h_2.



  - Third, to correctly classify 3 & 2, we can only choose h_3, which will misclassify 4.



  - Fourth, point 1 is the only point whose weight has not been increased. So we'll choose h_4 to misclassify it.



  - Fifth, h_5 can only be one of h_1, h_2, h_3, h_2. $\Rightarrow$ a repeat.

Problem 2 [15 points]
Recal the expression of the vote strength α_t for a weak classifier h_t in AdaBoost,

$$\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}, \tag{1}$$

where ϵ_t is the weighted training error of that weak classifier under the current weights at the beginning of iteration t in which it is chosen.

Show that (1) minimizes the empirical exponential loss (i.e., exponential loss on the training data) assuming the selection of the given h_t.

End of problem 2

- Let $A^{(t)} = \{ i \mid h_t(x_i) \neq y_i \}$
- $L(H_t, X, y) = \sum_{i=1}^{N} e^{-y_i \cdot H_t(x_i)}$

$$= \sum_{i=1}^{N} e^{-y_i \cdot H_{t-1}(x_i)} \cdot e^{-y_i \cdot \alpha_t h_t(x_i)}$$

$$= \sum_{i=1}^{N} W_i^{(t-1)} \cdot e^{-y_i \cdot \alpha_t h_t(x_i)}$$

$$= \sum_{i \in A^t} W_i^{(t-1)} \cdot e^{\alpha_t} + \sum_{i \notin A^t} W_i^{(t-1)} \cdot e^{-\alpha_t}$$

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

- take derivative and set to zero:

$$\frac{\partial L}{\partial \alpha_t} = \epsilon_t e^{\alpha_t} - (1-\epsilon_t) \cdot e^{-\alpha_t} = 0$$

take log:

$$\log \epsilon_t + \alpha_t = \log(1-\epsilon_t) - \alpha_t$$

$$\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$$

[With help from Haichuan Wang]

3 Optimal classification

We have seen in class that the minimal risk for a particular joint distribution $p(\mathbf{x}, y)$ under 0/1 loss

$$L_{0/1}(\hat{y}, y) = \begin{cases} 0 & \text{if } \hat{y} = y, \\ 1 & \text{if } \hat{y} \neq y \end{cases}$$

is attained by the Bayes classifier $h^*(\mathbf{x}) = \operatorname{argmax}_c p(c \mid \mathbf{x})$. One may suspect that this bound is limited to *deterministic* classifiers. An attempt to "beat" this bound, then, could be based on the following, *randomized* classifier. Define, for any data point $\mathbf{x}$, a probability distribution $q(c \mid \mathbf{x})$ over class labels c conditioned on the input $\mathbf{x}$. The resulting randomized classifier (for which q serves as a parameter), given a data point $\mathbf{x}$, draws a random class label from q:

$$h_r(\mathbf{x}; q) = c_r, \qquad c_r \sim q(c \mid \mathbf{x}).$$

To express the risk of this classifier we need to take the expectation over all possible outcomes of the random decision:

$$R(h_r; q) = \int_{\mathbf{x}} \sum_{c=1}^{C} \sum_{c'=1}^{C} L_{0/1}(c', c)\, q(c_r = c' \mid \mathbf{x})\, p(\mathbf{x}, y = c)\, d\mathbf{x}. \tag{2}$$

Problem 7 **[15 points]**
Show that for any q,

$$R(h_r; q) \geq R(h^*),$$

that is, that the risk of the randomized classifier h_r defined above is at least as high as the Bayes risk.

End of problem 7

Advice: As we saw in class, it is enough to show that the inequality holds for the conditional risk, i.e., that $R(h_r|\mathbf{x}) \geq R(h^*|\mathbf{x})$ for any $\mathbf{x}$. To attack this problem, write out the expectation in (2), conditional on an $\mathbf{x}$, and think of the best possible distribution q (in hindsight) that you could use to minimize it.

- Optimal Bayes
  - $R(h^* \mid x) = 1 - \max_c \{ p(y = c \mid x) \}$

- Random:
  - $R(h_r \mid x) = \sum_{c=1}^{C} \sum_{c'=1}^{C} L_{0/1}(c', c) \cdot q(c_r = c' \mid x) \cdot p(y = c \mid x)$

$$= \sum_{c=1}^{C} \sum_{c' \neq c}^{C} \underbrace{L(c', c)}_{=1} \cdot q(c_r = c') \cdot p(y = c \mid x)$$

$$+ \sum_{c=1}^{C} \sum_{c' = c}^{C} \underbrace{L(c', c)}_{=0} \cdot q(c_r = c') \cdot p(y = c \mid x)$$

$$= \sum_{c=1}^{C} \sum_{c' \neq c}^{C} q(c_r = c') \cdot p(y = c \mid x)$$

$$= 1 - \sum_{c=1}^{C} q(c_r = c) \cdot p(y = c \mid x)$$

- To minimize random risk,

let $c^* = \operatorname{argmax}_c \{ p(y = c \mid x) \}$,

then the best random distribution is

$$q(c_r \mid x) = \begin{cases} q(c_r = c^* \mid x) = 1 \\ q(c_r \neq c^* \mid x) = 0 \end{cases}$$

- Then,

$$R(h_r \mid x) = 1 - \sum_{c=1}^{C} q(c_r = c \mid x) \cdot p(y = c \mid x)$$

$$\geq 1 - \max_c \{ p(y = c \mid x) \}$$

$$= R(h^* \mid x).$$