Machine Learning
Problem Set 4

Ozaner Hansha

November 19, 2020

Question 1

Part a: Show that the weighted training error with updated weights is equal to $\frac{1}{2}$. (See problem set for details).

Solution: First, to make our math a bit cleaner, consider the following sets:

$$S = \{i \mid y_i \neq h_{M+1}(\mathbf{x}_i)\}$$
$$S^\complement = \{i \mid y_i = h_{M+1}(\mathbf{x}_i)\}$$

Now, before we prove the desired statement, consider the following lemmas:

$$\begin{aligned}
\alpha_{M+1} &= \frac{1}{2}\log\left(\frac{1-\epsilon_{M+1}}{\epsilon_{M+1}}\right) && \text{(def. of } \alpha_{M+1}\text{)}\\
\exp(2\alpha_{M+1}) &= \frac{1-\epsilon_{M+1}}{\epsilon_{M+1}}\\
\exp(-2\alpha_{M+1}) &= \frac{\epsilon_{M+1}}{1-\epsilon_{M+1}} && \text{(lemma 1)}
\end{aligned}$$

$$\begin{aligned}
1 &= \sum_{i=1}^{n} W_i^{(M)} && \text{(weights are normalized)}\\
&= \sum_{i\in S} W_i^{(M)} + \sum_{i\in S^\complement} W_i^{(M)} && \text{(partition of sum)}\\
\sum_{i\in S^\complement} W_i^{(M)} &= 1 - \sum_{i\in S} W_i^{(M)} && \text{(lemma 2)}
\end{aligned}$$

We can finally give the following chain of equalities:

$$\begin{aligned}
\epsilon'_{M+1} &= \sum_{i\in S} W_i^{(M+1)} && \text{(def. of WTE w/ UW)}\\
&= \sum_{i\in S} \frac{W_i^{(M)}\exp(-\alpha_{M+1} y_i h_{M+1}(\mathbf{x}_i))}{\sum_{j=1}^{n} W_j^{(M)}\exp(-\alpha_{M+1} y_j h_{M+1}(\mathbf{x}_j))} && \text{(def. of } W_i^{M+1}\text{)}\\
&= \sum_{i\in S} \frac{W_i^{(M)}\exp(\alpha_{M+1})}{\sum_{j=1}^{n} W_j^{(M)}\exp(-\alpha_{M+1} y_j h_{M+1}(\mathbf{x}_j))} && \left(\substack{(\forall i\in S)\ y_i h_{M+1}(\mathbf{x}_i)=-1 \\ \text{i.e. misclassifications}}\right)\\
&= \frac{\exp(\alpha_{M+1})}{\sum_{i=1}^{n} W_i^{(M)}\exp(-\alpha_{M+1} y_i h_{M+1}(\mathbf{x}_i))}\sum_{i\in S} W_i^{(M)}\\
&= \frac{\exp(\alpha_{M+1})\sum_{i\in S} W_i^{(M)}}{\sum_{i\in S} W_i^{(M)}\exp(-\alpha_{M+1} y_i h_{M+1}(\mathbf{x}_i)) + \sum_{i\in S^\complement} W_i^{(M)}\exp(-\alpha_{M+1} y_i h_{M+1}(\mathbf{x}_i))} && \text{(partition of sum)}\\
&= \frac{\exp(\alpha_{M+1})\sum_{i\in S} W_i^{(M)}}{\exp(\alpha_{M+1})\sum_{i\in S} W_i^{(M)} + \exp(-\alpha_{M+1})\sum_{i\in S^\complement} W_i^{(M)}} && \left(\substack{(\forall i\in S)\ y_i h_{M+1}(\mathbf{x}_i)=-1 \\ (\forall i\in S^\complement)\ y_i h_{M+1}(\mathbf{x}_i)=1}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\sum_{i\in S} W_i^{(M)}}{\sum_{i\in S} W_i^{(M)} + \exp(-2\alpha_{M+1})\sum_{i\in S^\complement} W_i^{(M)}} \\
&= \frac{\sum_{i\in S} W_i^{(M)}}{\sum_{i\in S} W_i^{(M)} + \frac{\epsilon_{M+1}}{1-\epsilon_{M+1}}\left(1-\sum_{i\in S} W_i^{(M)}\right)} && \text{(lemma 1 \& 2)} \\
&= \frac{\epsilon_{M+1}}{\epsilon_{M+1} + \frac{\epsilon_{M+1}}{1-\epsilon_{M+1}}(1-\epsilon_{M+1})} && \text{(def. of } \epsilon_{M+1}\text{)} \\
&= \frac{\epsilon_{M+1}}{\epsilon_{M+1}+\epsilon_{M+1}} = \frac{1}{2}
\end{aligned}
$$

And so we are done.

Part b: Would it be valid for our ensemble to have $h_m = h_{m+1}$ for some m?

Solution: Continuing from part a, consider what would happen if we choose $h_{M+2} = h_{M+1}$ (this is identical to the question as we have just set $m = M+1$). We first calculate the weighted training update error ϵ_{M+2} of our new classifier:

$$
\begin{aligned}
\epsilon_{M+2} &= \sum_{i:y_i\neq h_{M+2}(\mathbf{x}_i)} W_i^{(M+1)} && \text{(def. of WTE)} \\
&= \sum_{i:y_i\neq h_{M+1}(\mathbf{x}_i)} W_i^{(M+1)} && (h_{M+2} = h_{M+1}) \\
&= \epsilon'_{M+1} && \text{(def. of WTE w/ UW)} \\
&= \frac{1}{2} && \text{(part a)}
\end{aligned}
$$

Now let us compute the weighting α_{M+2} our new classifier will have in the ensemble:

$$
\begin{aligned}
\alpha_{M+2} &= \frac{1}{2}\log\frac{1-\epsilon_{M+2}}{\epsilon_{M+2}} && \text{(def. of } \alpha_{M+2}\text{)} \\
&= \frac{1}{2}\log\frac{1-\frac{1}{2}}{\frac{1}{2}} && (\epsilon_{M+2} = \tfrac{1}{2}) \\
&= \frac{1}{2}\log 1 \\
&= 0
\end{aligned}
$$

And so, our choice of classifier $h_{M+2} = h_{M+1}$ is a degenerate one since our ensemble at step $M+2$ is identical to the one at $M+1$:

$$
\begin{aligned}
H_{M+2}(\mathbf{x}) &= \sum_{m=1}^{M+2} \alpha_m h_m(\mathbf{x}) && \text{(def. of ensemble)} \\
&= \left(\sum_{m=1}^{M+1} \alpha_m h_m(\mathbf{x})\right) + \alpha_{M+2} h_{M+2}(\mathbf{x}) \\
&= \sum_{m=1}^{M+1} \alpha_m h_m(\mathbf{x}) && (\alpha_{M+2} = 0) \\
&= H_{M+1}(\mathbf{x}) && \text{(def. of ensemble)}
\end{aligned}
$$

And so there is no point in choosing the classifier in round $m+1$ of AdaBoost to be identical to the one in round m.

Part c: Can we have $h_{m+k} = h_m$ for some $k > 1$? Why or why not?

Solution: While we have shown in part b that the classifier in round $m + k$ cannot be identical to that of round m for $k = 1$, this does not hold for general $k > 1$. To see this, consider the weighted training error of h_{m+k}:

$$\epsilon_{m+k} = \sum_{i:y_i \neq h_{m+k}(\mathbf{x}_i)} W^{(m+k-1)}$$

Since $k > 1$, the weights $W^{(m+k-1)}$ are not necessarily equal to $W^{(m)}$ and so ϵ_{m+k} isn't necessarily $\frac{1}{2}$. Below we calculate the bounds of ϵ_{m+k}:

$$\begin{aligned}
1 &= \sum_{i=1}^{n} W_i^{(m+k)} && \text{(weights are normalized)} \\
&= \sum_{i:y_i \neq h_{m+k}(\mathbf{x}_i)} W_i^{(m+k)} + \sum_{i:y_i = h_{m+k}(\mathbf{x}_i)} W_i^{(m+k)} && \text{(partition of sum)} \\
1 &\geq \sum_{i:y_i \neq h_{m+k}(\mathbf{x}_i)} W_i^{(m+k)} \geq 0 && \text{(weights are positive)} \\
1 &\geq \epsilon_{m+k} \geq 0 && \text{(def. of WTE)}
\end{aligned}$$

In other words, $\epsilon_{m+k} \in [0, 1]$. Now consider the weighting of our classifier h_{m+k}:

$$\alpha_{m+k} = \frac{1}{2} \log \frac{1 - \epsilon_{m+k}}{\epsilon_{m+k}}$$

Note that the value of α_{m+k} could be anywhere from $[0, \infty)$ for $\epsilon_{m+k} \in [0, 1]$.

And so, even though our classifier at step $m + k$ is identical to that of step m, it is entirely possible that the classifier can produce a WTE not equal to $\frac{1}{2}$ and thus a ensemble weighting not equal to 0. As such it may be a reasonable choice of classifier.

Question 2

Problem: Show that the choice of α_m at time step m in AdaBoost minimizes the empirical exponential loss of the ensemble given the selection of the classifier h_m.

Solution: Let us first note the following lemma:

$$(\forall c \in [0, 1]) \; ce^{x} + (1 - c)e^{-x} \text{ is convex} \qquad \text{(lemma 3)}$$

Now we will find the α_m that minimizes the empirical exponential loss of H_m given H_{m-1} and h_m:

$$\begin{aligned}
\arg\min_{\alpha_m} L(H_m) &= \arg\min_{\alpha_m} \sum_{i=1}^{n} e^{-y_i H_m(\mathbf{x}_i)} && \text{(def. of exponential loss)} \\
&= \arg\min_{\alpha_m} \sum_{i=1}^{n} e^{-y_i (H_{m-1}(\mathbf{x}_i) + \alpha_m h_m(\mathbf{x}_i))} \\
&= \arg\min_{\alpha_m} \sum_{i=1}^{n} e^{-y_i (H_{m-1}(\mathbf{x}_i))} e^{-y_i \alpha_m h_m(\mathbf{x}_i)} \\
&= \arg\min_{\alpha_m} \sum_{i=1}^{n} W_i^{(m-1)} e^{-y_i \alpha_m h_m(\mathbf{x}_i)} && (W_i^{(m-1)} \propto e^{-y_i(H_{m-1}(\mathbf{x}_i))}) \\
&= \arg\min_{\alpha_m} \sum_{i:y_i \neq h_m(\mathbf{x}_i)}^{n} W_i^{(m-1)} e^{-y_i \alpha_m h_m(\mathbf{x}_i)} \\
&\qquad + \sum_{i:y_i = h_m(\mathbf{x}_i)}^{n} W_i^{(m-1)} e^{-y_i \alpha_m h_m(\mathbf{x}_i)} && \text{(partition sum)}
\end{aligned}$$

$$
\begin{aligned}
&= \operatorname*{arg\,min}_{\alpha_m} \quad e^{\alpha_m} \sum_{i:y_i \neq h_m(\mathbf{x}_i)}^{n} W_i^{(m-1)} + e^{-\alpha_m} \sum_{i:y_i = h_m(\mathbf{x}_i)}^{n} W_i^{(m-1)} && \left(\substack{(\forall i \in S)\ y_i h_m(\mathbf{x}_i) = -1 \\ (\forall i \in S^{\complement})\ y_i h_m(\mathbf{x}_i) = 1}\right) \\
&= \operatorname*{arg\,min}_{\alpha_m} \quad e^{\alpha_m}\epsilon_m + e^{-\alpha_m}(1-\epsilon) && \text{(lemma 2 and def. of WTE)} \\
&= \alpha_m \text{ s.t. } \frac{\mathrm{d}}{\mathrm{d}x} e^{\alpha_m}\epsilon_m + e^{-\alpha_m}(1-\epsilon) = 0 && \text{(lemma 3 \& min of convex func.)} \\
&= \alpha_m \text{ s.t. } e^{\alpha_m}\epsilon_m - e^{-\alpha_m}(1-\epsilon) = 0 \\
&= \alpha_m \text{ s.t. } e^{\alpha_m}\epsilon_m = e^{-\alpha_m}(1-\epsilon) \\
&= \alpha_m \text{ s.t. } \frac{e^{\alpha_m}}{e^{-\alpha_m}} = \frac{1-\epsilon}{\epsilon_m} \\
&= \alpha_m \text{ s.t. } \frac{1}{e^{-2\alpha_m}} = \frac{1-\epsilon}{\epsilon_m} \\
&= \alpha_m \text{ s.t. } 2\alpha_m = \log \frac{1-\epsilon}{\epsilon_m} \\
&= \frac{1}{2} \log \frac{1-\epsilon_m}{\epsilon_m}
\end{aligned}
$$

And so we are done.

Question 3

Part a: Show that logistic regression on a linear combination of feature vectors can be phrased in terms of kernel values, forming a kernel logistic regression (KLR).

Solution: Note the following:

$$
\begin{aligned}
\hat{p}(y = 1 \mid \mathbf{x}; \mathbf{w}) &= \sigma\left(\sum_{j=1}^{d} w_j \phi_j(\mathbf{x})\right) && \text{(logistic regression model)} \\
&= \sigma\left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})\right) \\
&= \sigma\left(\sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x})\right) && \text{(representor theorem)} \\
&= \sigma\left(\sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x})\right) && \text{(def. of kernel)}
\end{aligned}
$$

And so we have reformulated our logistic regression into a kernelized one. Note that in line 3 we invoke the representor theorem which tells us that our weight vector is some linear combination of our training data.

Part b: Give the gradient of the log loss of the KLR model with L_2 regularization. Show that it too can be phrased in terms of kernel values, meaning that gradient descent over KLR can be done with just kernel values.

Solution: Before we continue, note the following notation:

$$
K_j(\mathbf{x}) = K(\mathbf{x}_j, \mathbf{x}) \implies \boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j K(\mathbf{x}_j, \mathbf{x})
$$

The gradient of the log loss function w.r.t to the parameters $\boldsymbol{\alpha}$ is given by:

$$
\begin{aligned}
\nabla_{\boldsymbol{\alpha}} L(\hat{p}_{\boldsymbol{\alpha}}) &= \nabla_{\boldsymbol{\alpha}} \left(\lambda \|\boldsymbol{\alpha}\|^2 + \frac{-1}{n} \sum_{i=1}^{n} y_i \log(\hat{p}_{\boldsymbol{\alpha}}(\mathbf{x}_i)) + (1-y_i)\log(1-\hat{p}_{\boldsymbol{\alpha}}(\mathbf{x}_i)) \right) && \text{(regularized log loss)} \\
&= \nabla_{\boldsymbol{\alpha}} \lambda \|\boldsymbol{\alpha}\|^2 - \frac{1}{n} \sum_{i=1}^{n} y_i \nabla_{\boldsymbol{\alpha}} \log(\hat{p}_{\boldsymbol{\alpha}}(\mathbf{x}_i)) + (1-y_i)\nabla_{\boldsymbol{\alpha}} \log(1-\hat{p}_{\boldsymbol{\alpha}}(\mathbf{x}_i)) \\
&= 2\lambda\boldsymbol{\alpha} - \frac{1}{n} \sum_{i=1}^{n} y_i \nabla_{\boldsymbol{\alpha}} \log\big(\sigma(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))\big) + (1-y_i)\nabla_{\boldsymbol{\alpha}} \log\big(1-\sigma(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))\big) \\
&= 2\lambda\boldsymbol{\alpha} - \frac{1}{n} \sum_{i=1}^{n} y_i \nabla_{\boldsymbol{\alpha}} \log\big(\sigma(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))\big) + (1-y_i)\nabla_{\boldsymbol{\alpha}} \log\big(\sigma(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))\big) && (\sigma \text{ is odd}) \\
&= 2\lambda\boldsymbol{\alpha} + \frac{1}{n} \sum_{i=1}^{n} y_i \nabla_{\boldsymbol{\alpha}} \log\big(1+\exp(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))\big) + (1-y_i)\nabla_{\boldsymbol{\alpha}} \log\big(1+\exp(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))\big) \\
& && (\text{def. of } \sigma) \\
&= 2\lambda\boldsymbol{\alpha} + \frac{1}{n} \sum_{i=1}^{n} y_i \sigma(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)) \nabla_{\boldsymbol{\alpha}} \exp\big(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big) \\
&\quad + (1-y_i)\sigma(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)) \nabla_{\boldsymbol{\alpha}} \exp\big(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big) && \text{(chain rule)} \\
&= 2\lambda\boldsymbol{\alpha} + \frac{1}{n} \sum_{i=1}^{n} -y_i \sigma(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)) \exp\big(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big) \nabla_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i) \\
&\quad + (1-y_i)\sigma(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)) \exp\big(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big) \nabla_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i) && \text{(chain rule)} \\
&= 2\lambda\boldsymbol{\alpha} + \frac{1}{n} \sum_{i=1}^{n} -y_i \sigma(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)) \exp\big(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big) \mathbf{K}(\mathbf{x}_i) \\
&\quad + (1-y_i)\sigma(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)) \exp\big(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big) \mathbf{K}(\mathbf{x}_i) \\
&= 2\lambda\boldsymbol{\alpha} + \frac{1}{n} \sum_{i=1}^{n} \frac{(1-y_i)\mathbf{K}(\mathbf{x}_i)\exp\big(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big)}{1+\exp(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))} - \frac{y_i \mathbf{K}(\mathbf{x}_i)\exp\big(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big)}{1+\exp(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))} && (\text{def. of } \sigma) \\
&= 2\lambda\boldsymbol{\alpha} + \frac{1}{n} \sum_{i=1}^{n} \left(\frac{(1-y_i)\exp\big(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big)}{1+\exp(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))} - \frac{y_i \exp\big(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big)}{1+\exp(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))} \right) \mathbf{K}(\mathbf{x}_i)
\end{aligned}
$$

While complicated, we can see that the gradient depends on the training data X via the kernel values $\mathbf{K}(\mathbf{x}_i)$. And thus the same is true for training a KLR model using gradient descent:

$$
\begin{aligned}
\boldsymbol{\alpha}^{(t+1)} &= \boldsymbol{\alpha}^{(t)} - \eta \nabla_{\boldsymbol{\alpha}} \\
&= \boldsymbol{\alpha}^{(t)} - \eta \left(2\lambda\boldsymbol{\alpha} + \frac{1}{n} \sum_{i=1}^{n} \left(\frac{(1-y_i)\exp\big(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big)}{1+\exp(\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))} - \frac{y_i \exp\big(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i)\big)}{1+\exp(-\boldsymbol{\alpha}^\top \mathbf{K}(\mathbf{x}_i))} \right) \mathbf{K}(\mathbf{x}_i) \right)
\end{aligned}
$$