HW3

Juan Arroyo-Miranda

Problem 1

We need to find the expressions for $\mathbf{H}, \mathbf{f}, \mathbf{A}, \mathbf{a}, \mathbf{B}$ and $\mathbf{b}$ such that we can setup the dual optimization problem for the kernel SVM:

$$\underset{\mathbf{w}}{\operatorname{argmin}}\left\{\frac{1}{2}||\mathrm{w}||^2 + C\sum_{i=1}^{N}\max_{0\le\alpha\le C}[0, 1 - y_i(\mathrm{w}\cdot\phi(\mathrm{x_i}) - w_0)]\right\}$$

From the problem above, we see that we are dealing with the case of not linearly separable data, which means that we are interested in imposing a maximum penalty (C) on constraint violation, where $\xi_i = \max[0, 1 - y_i(\mathrm{w}\cdot\phi(\mathrm{x_i}) - w_0)]$. We can rewrite the dual problem using a Lagrangian with the following constraints and KKT conditions:

$$\begin{aligned}
\alpha_i &\ge 0,\\
y_i(w_0 + \mathrm{w}\cdot\phi(\mathrm{x_i})) - 1 + \xi_i &\ge 0,\\
\alpha_i(y_i(w_0 + \mathrm{w}\cdot\phi(\mathrm{x_i})) - 1 + \xi_i) &= 0,\\
\mu_i &\ge 0,\\
\xi_i &\ge 0,\\
\mu_i\xi_i &= 0
\end{aligned}$$

Now, we have to solve this problem:

$$L = \frac{1}{2}||\mathrm{w}||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(w_0 + \mathrm{w}\cdot\phi(\mathrm{x_i}) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i$$

Taking the derivative with respect to w, w_0, and ξ_i we get

$$\frac{\delta L}{\delta \mathrm{w}} = \mathrm{w} - \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) = 0$$

$$\Rightarrow \mathrm{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \qquad (1)$$

$$\frac{\delta L}{\delta \mathrm{w}_0} = -\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad (2)$$

$$\frac{\delta L}{\delta \mathrm{\xi_i}} = C - \alpha_i - \mu_i = 0$$

$$\Rightarrow \alpha_i = C - \mu_i \qquad (3)$$

If we replace the values for w, w_0, and $\{\xi_i\}$ in the Lagrangian, we can reformulate the original problem as follows:

$$\begin{aligned} L = {} & \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i \phi(x_i)\right) \cdot \left(\sum_{i=j}^{N} \alpha_j y_j \phi(x_j)\right) + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} (C - \alpha_i)\xi_i \\ & - \left[w_0 \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i y_i \left(\sum_{j=1}^{N} \alpha_j y_j \phi(x_j)\right) \cdot \phi(x_i) - \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \alpha_i \xi_i \right] \\ = {} & \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i \phi(x_i)\right) \cdot \left(\sum_{i=j}^{N} \alpha_j y_j \phi(x_j)\right) + \sum_{i=1}^{N} \alpha_i - \left(\sum_{i=1}^{N} \alpha_i y_i \phi(x_i)\right) \cdot \left(\sum_{j=1}^{N} \alpha_j y_j \phi(x_j)\right) \\ = {} & \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i \phi(x_i)\right) \cdot \left(\sum_{i=j}^{N} \alpha_j y_j \phi(x_j)\right) \end{aligned}$$

Before we continue, lets rewrite the last expression in terms of matrices

$$\sum_{i=1}^{N} \alpha_i y_i \phi(x_i) = \mathbf{Z}^{\mathrm{T}} \boldsymbol{\alpha} \quad \text{where} \quad \mathbf{Z} = \begin{pmatrix} y_1\phi(x_{11}) & y_1\phi(x_{21}) & \cdots & y_1\phi(x_{N1}) \\ y_2\phi(x_{12}) & y_2\phi(x_{22}) & \cdots & y_2\phi(x_{N2}) \\ y_3\phi(x_{13}) & y_3\phi(x_{23}) & \cdots & y_3\phi(x_{N3}) \\ \vdots & \ddots & \ddots & \vdots \\ y_N\phi(x_{1N}) & y_N\phi(x_{2N}) & \cdots & y_N\phi(x_{NN}) \end{pmatrix} \quad \text{and} \quad \boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix}$$

$$\sum_{i=1}^{N} \alpha_i = [1, 1, \cdots, 1]^T \boldsymbol{\alpha}$$

Therefore, the Lagrangian for the original problem can be expressed as:

$$\begin{aligned} L = \ & [1, 1, \cdots, 1]^T \boldsymbol{\alpha} - \frac{1}{2} (\mathbf{Z}^{\mathrm{T}} \boldsymbol{\alpha})^T (\mathbf{Z}^{\mathrm{T}} \boldsymbol{\alpha}) \\ = \ & [1, 1, \cdots, 1]^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Z} \mathbf{Z}^{\mathrm{T}} \boldsymbol{\alpha} \\ = \ & [1, 1, \cdots, 1]^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} \quad \text{where} \quad \mathbf{H} = \mathbf{Z}\mathbf{Z}^{\mathrm{T}} = diag(\mathrm{y}) \Phi\mathbf{(X)} \Phi\mathbf{(X)}^{\mathrm{T}} diag(\mathrm{y}) \end{aligned}$$

$$\Phi\mathbf{(X)} = \begin{pmatrix} \phi(x_{11}) & \phi(x_{21}) & \cdots & \phi(x_{N1}) \\ \phi(x_{12}) & \phi(x_{22}) & \cdots & \phi(x_{N2}) \\ \phi(x_{13}) & \phi(x_{23}) & \cdots & \phi(x_{N3}) \\ \vdots & \ddots & \ddots & \vdots \\ \phi(x_{1N}) & \phi(x_{2N}) & \cdots & \phi(x_{NN}) \end{pmatrix} \quad \text{and} \quad \mathrm{K} = \Phi\mathbf{(X)} \Phi\mathbf{(X)}^{\mathrm{T}}$$

We can rewrite the above proble as a minimization problem with respect to $\boldsymbol{\alpha}$.

$$\begin{aligned} \min_{\boldsymbol{\alpha}} L = & \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} + [-1, -1, \cdots, -1]^T \boldsymbol{\alpha} \quad \text{where} \quad \mathbf{f}^{\mathrm{T}} = [-1, -1, \cdots, -1]^T \\ = & \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} + \mathbf{f}^{\mathrm{T}} \boldsymbol{\alpha} \end{aligned}$$

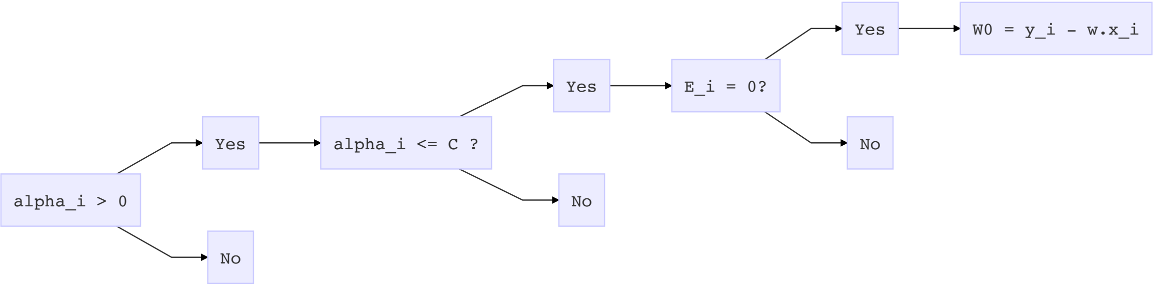
This is identical to the separable case, except that the constraints are somewhat different. To see what these constraints are, we note that $\alpha_i \geq 0$ is required because these are Lagrange multipliers. We also know that $\mu_i \geq 0$ is required because these also are Lagrange multipliers.

When we combine this second condition with the result from (3), we get $C - \alpha_i \geq 0$, which implies that $\alpha_i \leq C$.

Therefore, we have to miminze the new Lagrangian with respect to the variables $\{\alpha_i\}$ subject to

$$0 \le \alpha_i \le C$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$



We can express the constraint $0 \le \alpha_i \le C$ as follows:

$$\mathrm{A}\boldsymbol{\alpha} \le \mathbf{a} \quad \text{where}$$

$$A, \mathbf{a} = \begin{cases} \mathrm{I},\ \mathbf{a} = [C, C, \cdots, C]^T \quad \text{if} \quad \alpha_i \le C \\ \\ \gamma\mathrm{I},\ \mathbf{a} = [0, 0, \cdots, 0]^T \quad \text{if} \quad -\alpha_i \le 0 \quad \text{where} \quad \gamma = -1 \end{cases}$$

$$A = \begin{pmatrix} \mathrm{I} & 0 \\ 0 & \gamma\mathrm{I} \end{pmatrix} \quad \text{and} \quad \mathbf{a} = \begin{pmatrix} \vec{\mathrm{C}} \\ \vec{0} \end{pmatrix}$$

Finally, we need to write an expression for the second constraint, that is, $\sum_{i=1}^{N} \alpha_i y_i = 0$

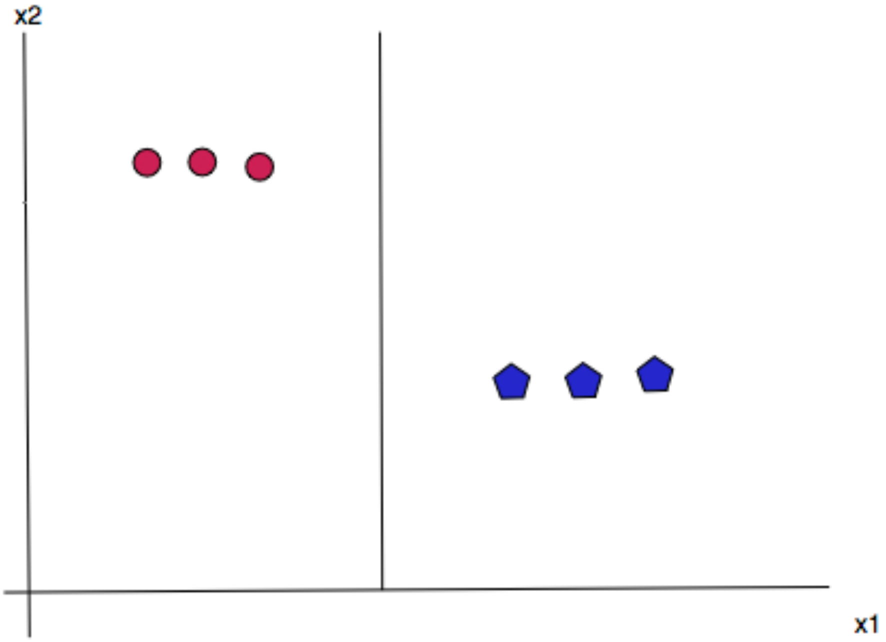
$$\mathrm{B}\boldsymbol{\alpha} = 0 \quad \text{where} \quad \mathrm{B} = \boldsymbol{y}^T$$

Problem 3

Yes, we can use a decision tree in this instance. To construct oru answer we used the following handout from University of Toronto (http://www.cs.toronto.edu/~toni/Courses/265-2010/handouts/dec-trees.pdf)

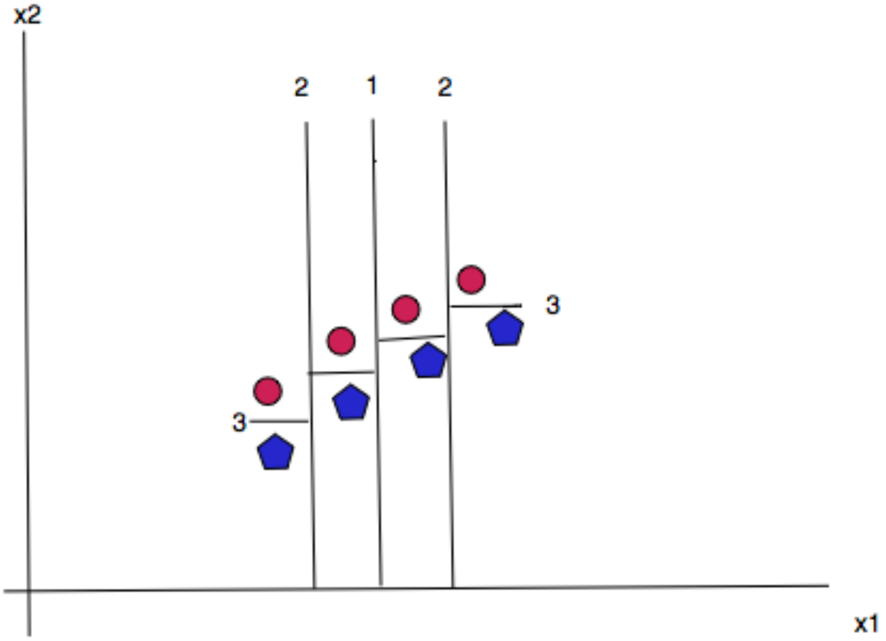
To illustrate this idea, let us examine two easy classifcation problems to get a better sense of the behavior between the sample size and the depth of the tree for the case of linearly separable data. We have to take into consideration that in the worst-case scenario, the decision tree would always split the remaining are in half.

Example 1:



In example 1, one split is sufficient to get a correct classification of 6 data points. Now, lets look at a slighty more complex example.

Example 2:



In example 2, we can see that three split are necessary to correctly classify 8 data points. From this we can generate a rough rule of thumb, that for N data points we require $log_2 N$ splits. It seems that this rule of thumb provides us with a lower bound for the worst-case complexity of the tree. In the above case, $log_2 8 = 3$, matches the number of splits in example 2. However, this intuition might not be correct as we will argue.

In general, before making a split, with need to evaluate the performance of each classifier at each node, that is, how good is a classifier in terms of arraging the data into homogenous sets. Some of these measures are information gain and entropy. Finally, we pick the classifier with the best (lowest information gain for example) performance and repeat these steps until we classified all the data.

However, we do not know which is the best classifier at each step until we compute any of these measures (entropy or information gain), that is, we are dealing with an unsorted set of classifiers at each point.For this reason, we cannot rely on $log_2 N$ as a measure for the tree's depth because this measure only holds for sorted data.

So, we start by sorting k unordered classifiers, then we remove the best of them and we sort k-1 unordered classifiers, and we continue this process until we have one classifier left. Since we have k distinct classifiers, there are $k!$ permutations of the classifiers and therefore $k!$ outputs for sorting them. The information-theoretic argument says that we need at least $\lceil log_2 k! \rceil$ comparisons. But $log_2 k! = \sum_{i=1}^{n} log_2 i$ and this quantity is $O(k \, log \, k)$.

Problem 4

There are two forms of non-linear separability.

Form 1: We have two points belonging to two different classes which lie on the same point $\mathbf{x}$ in the space, where $\mathbf{x} = (x_1, x_2)$. In this instance, there is no tree which can be constructed that can separate these points. As such, we will always have 1 point that is misclassified.

Form 2: If there are no points overlap, then the problem is similar to the one we discussed above. In this case, we will start with k unordered classifiers and ordered them according to some performance measure, and repeat this proccess until we have one classifier left. Therefore, the complexity for the non-separable case is also $O(k \, log \, k)$.

Problem 5

To answer this problem, suppose that we can express $W_i^{(T+1)}$ as follows (See Bishop p. 661, (14.24)):

$$W_i^{(T+1)} = \frac{W_i^{(T)}}{Z} e^{-y_i \alpha_T h_T(\mathbf{x}_\mathrm{i})} \qquad (1)$$

where *Z* represents some normalizing constant.

We know that the training error for the period $T+1$ is defined as $\epsilon_{T+1} = \sum\limits_{i:yi\neq h_{T+1}(\mathbf{x}_{\mathrm{i}})} W_i^{(T+1)}$, but before we calculate the sum over the misclassified weights in period $T+1$, it will be useful to compute $\sum_{i=1}^{N} W_i^{(T+1)}$ and we will assume that this sum is equal to 1.

$$
\begin{aligned}
\sum_{i=1}^{N} W_i^{(T+1)} &= \sum_{i=1}^{N} \frac{W_i^{(T)}}{Z} e^{-y_i \alpha_T h_T(\mathbf{x}_{\mathrm{i}})} \\
&= \sum_{i:\, y_i = h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z} e^{-\alpha_T} + \sum_{i:\, y_i \neq h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z} e^{\alpha_T} \\
&= e^{-\alpha_T} \sum_{i:\, y_i = h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z} + e^{\alpha_T} \sum_{i:\, y_i \neq h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z} \quad \text{substitute} \quad \alpha_T = \frac{1}{2}\, log \frac{1-\epsilon_T}{\epsilon_T} \\
&= \frac{1}{e(log(\frac{1-\epsilon_T}{\epsilon_T})^{1/2})} \sum_{i:\, y_i = h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z} + e\left(log\left(\frac{1-\epsilon_T}{\epsilon_T}\right)^{1/2}\right) \sum_{i:\, y_i \neq h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z} \\
&= \left(\frac{\epsilon_T}{1-\epsilon_T}\right)^{1/2} \sum_{i:\, y_i = h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z} + \left(\frac{1-\epsilon_T}{\epsilon_T}\right)^{1/2} \sum_{i:\, y_i \neq h_T(\mathbf{x}_{\mathrm{i}})} \frac{W_i^{(T)}}{Z}
\end{aligned}
$$

Since, $\epsilon_T = \sum\limits_{i:\, y_i \neq h_T(\mathbf{x}_{\mathrm{i}})} W_i^{(T)}$ and we assumed that $\sum\limits_{i=1}^{N} W_i^{(T)} = 1$, we have $1 - \epsilon_T = \sum\limits_{i:\, y_i = h_T(\mathbf{x}_{\mathrm{i}})} W_i^{(T)}$

.

With this last two pieces of information, we can express the ensemble loss as:

$$
\begin{aligned}
\sum_{i=1}^{N} W_i^{(T+1)} &= \frac{1}{Z}\left[\left(\frac{\epsilon_T}{1-\epsilon_T}\right)^{1/2} \sum_{i:\, y_i = h_T(\mathbf{x}_{\mathrm{i}})} W_i^{(T)} + \left(\frac{1-\epsilon_T}{\epsilon_T}\right)^{1/2} \sum_{i:\, y_i \neq h_T(\mathbf{x}_{\mathrm{i}})} W_i^{(T)}\right] \\
&= \frac{1}{Z}\left[\left(\frac{\epsilon_T}{1-\epsilon_T}\right)^{1/2} (1-\epsilon_T) + \left(\frac{1-\epsilon_T}{\epsilon_T}\right)^{1/2} \epsilon_T\right] \\
&= \frac{1}{Z}\left[(\epsilon_T(1-\epsilon_T))^{1/2} + (\epsilon_T(1-\epsilon_T))^{1/2}\right] \\
&= \frac{1}{Z}\left[2(\epsilon_T(1-\epsilon_T))^{1/2}\right]
\end{aligned}
$$

Since we assumed that $\sum_{i=1}^{N} W_i^{(T+1)} = 1$, this implies that $Z = 2(\epsilon_T(1-\epsilon_T))^{1/2}$.

We can rewrite (1) to see the value that $W_i^{(T+1)}$ takes when we classify samples correctly and when we misclassify samples using again $\alpha_T = \frac{1}{2}\, log \frac{1-\epsilon_T}{\epsilon_T}$ and substituting our value for *Z*.

$$W_i^{(T+1)} = \begin{cases} \frac{W_i^{(T)}}{Z} e^{-\alpha_T} \quad \text{if} \quad y_i = h_T(\mathbf{x}_\mathrm{i}) \\ \frac{W_i^{(T)}}{Z} e^{\alpha_T} \quad \text{if} \quad y_i \neq h_T(\mathbf{x}_\mathrm{i}) \end{cases}$$

$$= \begin{cases} \frac{W_i^{(T)}}{Z} \left(\frac{\epsilon_T}{1-\epsilon_T} \right)^{1/2} \quad \text{if} \quad y_i = h_T(\mathbf{x}_\mathrm{i}) \\ \frac{W_i^{(T)}}{Z} \left(\frac{1-\epsilon_T}{\epsilon_T} \right)^{1/2} \quad \text{if} \quad y_i \neq h_T(\mathbf{x}_\mathrm{i}) \end{cases}$$

$$= \begin{cases} \frac{W_i^{(T)}}{2(\epsilon_T(1-\epsilon_T))^{1/2}} \left(\frac{\epsilon_T}{1-\epsilon_T} \right)^{1/2} \quad \text{if} \quad y_i = h_T(\mathbf{x}_\mathrm{i}) \\ \frac{W_i^{(T)}}{2(\epsilon_T(1-\epsilon_T))^{1/2}} \left(\frac{1-\epsilon_T}{\epsilon_T} \right)^{1/2} \quad \text{if} \quad y_i \neq h_T(\mathbf{x}_\mathrm{i}) \end{cases}$$

$$= \begin{cases} \frac{W_i^{(T)}}{2} \frac{1}{1-\epsilon_T} \quad \text{if} \quad y_i = h_T(\mathbf{x}_\mathrm{i}) \\ \frac{W_i^{(T)}}{2} \frac{1}{\epsilon_T} \quad \text{if} \quad y_i \neq h_T(\mathbf{x}_\mathrm{i}) \end{cases}$$

Finally, if we sum over the misclassified samples we get:

$$\epsilon_{T+1} = \sum_{i:y_i \neq h_{T+1}(\mathbf{x}_\mathrm{i})} W_i^{(T+1)}$$

$$= \frac{1}{2} \frac{1}{\epsilon_T} \sum_{i:y_i \neq h_T(\mathbf{x}_\mathrm{i})} W_i^{(T)}$$

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

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

This result also implies that $1 - \epsilon_{T+1} = \sum_{i:y_i = h_{T+1}(\mathbf{x}_\mathrm{i})} W_i^{(T+1)} = \frac{1}{2}$.

The fact that $\epsilon_{T+1} = \epsilon_T = \frac{1}{2}$ implies that in order to get the weights for the next generation, we just need to take the sum of the correctly classified weights and rescale these weights so that they add up to one half. However, the rule for updating the weights is not related to our choice for the classifier in the next period. In period $T+2$, we want to pick a classifier h_{T+2} that can better classify the previously misclassified points. Therefore, $h_{T+2} \neq h_{T+1}$.

Problem 6

Suppose that the base classifiers $h_1(x), \ldots, h_{m-1}(x)$ are fixed, as are their coefficients $\alpha_1, \ldots, \alpha_{m-1}$, and so we are minimizing only with respect to α_m and $h_m(x)$.

Separating off the contribution from base classifier $h_m(x)$, we can then write the error function in the form:

$$\begin{aligned}
L(\mathrm{H}_m, X) &= \sum_{i=1}^{N} e^{-y_i\cdot[\mathrm{H}_{m-1}(\mathbf{x}_\mathrm{i})+\alpha_m h_m(\mathbf{x}_\mathrm{i})]} \\
&= \sum_{i=1}^{N} e^{-y_i\mathrm{H}_{m-1}(\mathbf{x}_\mathrm{i})-\alpha_m h_m(\mathbf{x}_\mathrm{i})} \\
&= \sum_{i=1}^{N} e^{-y_i\mathrm{H}_{m-1}(\mathbf{x}_\mathrm{i})} \cdot e^{-\alpha_m h_m(\mathbf{x}_\mathrm{i})} \\
&= \sum_{i=1}^{N} W_i^{(m-1)} e^{-y_i\alpha_m h_m(\mathbf{x}_\mathrm{i})} \quad \text{where} \quad W_i^{(m-1)} = e^{-y_i H_{m-1}(\mathbf{x}_\mathrm{i})} \\
&= \sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} e^{-\alpha_m} + \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} e^{\alpha_m} \\
&= e^{-\alpha_m} \sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} + e^{\alpha_m} \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \\
&= e^{-\alpha_m} \left(\sum_{i=1}^{N} W_i^{(m-1)} - \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \right) + e^{\alpha_m} \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \\
&= (e^{\alpha_m} - e^{-\alpha_m}) \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} + e^{-\alpha_m} \sum_{i=1}^{N} W_i^{(m-1)}
\end{aligned}$$

Taking the derivative with respect to α_m, we get the following:

$$\frac{\delta L}{\delta \alpha_m} = e^{\alpha_m} \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} + \quad e^{-\alpha_m} \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} - \quad e^{-\alpha_m} \sum_{i=1}^{N} W_i^{(m-1)} = 0$$

$$= e^{\alpha_m} \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \quad + \quad e^{-\alpha_m} \left(\sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} - \quad \sum_{i=1}^{N} W_i^{(m-1)} \right) = 0$$

$$= e^{\alpha_m} \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \quad - \quad e^{-\alpha_m} \sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} = 0$$

$$\Rightarrow e^{\alpha_m} \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \quad = \quad e^{-\alpha_m} \sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)}$$

$$\Rightarrow \alpha_m + log\left(\sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \right) = -\alpha_m + log\left(\sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \right)$$

$$\Rightarrow 2\alpha_m = log\left(\sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \right) - log\left(\sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)} \right)$$

$$\Rightarrow \alpha_m = \frac{1}{2}\, log\left(\frac{\sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)}}{\sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)}} \right)$$

Since, $\epsilon_m = \sum_{i:\, y_i \neq h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)}$ and we assumed that $\sum_{i=1}^{N} W_i^{(m-1)} = 1$, we have $1 - \epsilon_m = \sum_{i:\, y_i = h_m(\mathbf{x}_\mathrm{i})} W_i^{(m-1)}$.

We get that the α_m that minimizes the empirical exponential loss is:

$$\alpha_m = \frac{1}{2}\, log\left(\frac{1 - \epsilon_m}{\epsilon_m} \right)$$