Problem 1: Perceptron

[25 points]

(1) [25 points] Consider the points $x_i = (0, \ldots, 0, 1, 0, \ldots 0)$, $i = 1, \ldots, m$, on the sphere of radius $R = 1$ in $\mathbb{R}^m$. Then the set is separated with the weight vector $w = (y_1, \ldots, y_m)$, margin $\rho = \frac{|w \cdot x_i|}{\sqrt{m}} = \frac{R}{\sqrt{m}}$, and with exactly $m = \frac{R^2}{\rho^2}$ updates.

Problem 2: Boosting

(1) [25 points] As seen in Lecture 8, at each boosting round $t$, Adaboost assigns an equal probability mass to the examples correctly classified and those misclassified by $h_t$. $h_t$ cannot be selected at the next round since its error rate is exactly $\frac{1}{2}$ with that distribution and it would need to be $\frac{1}{2} - \epsilon$ for some $0 < \epsilon \leq \frac{1}{2}$.

(2) [50 points]

(a) [10 points] Let $G_1(x) = e^x$ and $G_2(x) = x + 1$. $G_1$ and $G_2$ are continuously differentiable over $\mathbb{R}$ and $G_1'(0) = G_2'(0)$. Thus, $G$ is differentiable over $\mathbb{R}$. Note that $G' \geq 0$.

Both $G_1$ and $G_2$ are convex, thus

$$G(y) - G(x) \geq G'(x)(y - x)$$

for $x, y \leq 0$ or $x, y \geq 0$. Assume now that $y \leq 0$ and $x \geq 0$, then

$$G(y) - G(x) = e^y - (x + 1) \geq (y+1) - (x+1) = G'(x)(y - x),$$

since $G'(x) = 1$. Thus $G$ is convex.
(b) [35 points] The direction \( e_u \) taken by coordinate descent after \( T-1 \) rounds is the argmin\(_u\) of:

\[
\frac{dG(\alpha + \beta e_u)}{d\beta} \bigg|_{\beta=0} = -\sum_{i=1}^{m} y_i h_u(x_i) G'(-y_i f(x_i))
\]

(since \( G' \geq 0 \))

\[
\propto -\sum_{i=1}^{m} y_i h_u(x_i) \frac{G'(-y_i f(x_i))}{\sum_{i=1}^{m} G'(-y_i f(x_i))}
\]

\[
\propto -\sum_{i=1}^{m} y_i h_u(x_i) D_{T-1}(i)
\]

\[
= -(1 - 2\epsilon_u),
\]

with \( D_{T-1}(i) = \frac{1}{m} \sum_{i=1}^{m} G'(-y_i f(x_i)) \). Thus, the base classifier \( h_u \) selected at each round is the one with the minimal error rate over the training data.

The step size \( \beta \) is the solution of:

\[
\frac{dF(\alpha + \beta e_u)}{d\beta} = -\sum_{i=1}^{m} y_i h_u(x_i) G'(-y_i f(x_i) - \beta y_i h_u(x_i)) = 0,
\]

which can be solved numerically. A closed form solution can be given under certain conditions, e.g., if

\[
\beta \leq \rho = \min_{i \in [1,m]} |f(x_i)|,
\]

(c) [5 points] By definition of the objective function, this algorithm is less aggressively reducing the empirical error rate than AdaBoost.