1. **Problem 1:** Consider the following formulation of Adaboost. As in class, we start with a training set of labelled examples: \( \{(x_i, y_i)\}_{i=1,...,m} \), where \((x_i, y_i) \in \chi \times \{-1, 1\}\). Let \( \mathcal{H} = \{h_1, \ldots, h_n\} \) be the set of weak classifiers where \( h_j : \chi \to \{1, -1\} \) (note: we assume a finite number \( n \) of weak classifiers, where \( m \ll n \)). We define an \( m \times n \) matrix \( M \) where \( M_{ij} = y_i h_j(x_i) \), i.e., \( M_{ij} = +1 \) if training example \( i \) is classified correctly by weak classifier \( h_j \), and \(-1\) otherwise. Let \( d_t, \lambda_t \in \mathbb{R}^n \), \( \|d_t\|_1 = 1 \) and \( d_{t,i} \) (respectively \( \lambda_{t,i} \)) equal \( i^{th} \) component of \( d_t \) (respectively \( \lambda_t \)). Let \( d^T \) denote the transpose of the vector \( d \). Now we define the following algorithm:

(a) **Input:** Matrix \( M \), Number of iterations \( t_{max} \)
(b) **Initialize:** \( \lambda_{1,j} = 0 \) for \( j = 1, \ldots, n \)
(c) **Loop for** \( t = 1, \ldots, t_{max} \)
   i. \( d_{t,i} = \frac{\exp(-\langle M\lambda_t \rangle_i)}{\sum_{k=1}^n \exp(-\langle M\lambda_t \rangle_k)} \) for \( i = 1, \ldots, m \)
   ii. \( j_t \in \text{argmax}_j (d_t^T M)_j \)
   iii. \( r_t = (d_t^T M)_{j_t} \)
   iv. \( \alpha_t = \frac{1}{2} \ln \left( \frac{1+r_t}{1-r_t} \right) \)
   v. \( \lambda_{t+1} = \lambda_t + \alpha_t e_{j_t} \), where \( e_{j_t} \) is 1 in position \( j_t \) and 0 elsewhere.
(d) **Output:** \( \frac{\lambda_{t_{max}}}{\|\lambda_{t_{max}}\|_1} \)

(a) Is this approach of explicitly using \( M \) practical? Why/Why not?
(b) What does \( d_{1,i} \) equal for \( t = 1 \) for each value of \( i \)?
(c) In one sentence, explain what is happening in step (c).i.
(d) \((d_t^T M)_j\) is called the "edge" of weak classifier \(j\) at time \(t\) w.r.t. the training examples. What are the max and min values for the edge of a weak classifier at time \(t\)?

(e) What do large and small values of \(r_t\) tell us about the classifier?

(f) How would you write the combined classifier \(H(x)\) as defined in lecture in terms of \(\lambda_{t_{\text{max}}}\)?

2. **Problem 2**: The explicit mapping between \(d_t\) and \(D_{t-1}\) for the algorithm presented in Problem 1 can defined as follows:

(a) \(j_t \in \arg\max_j (d_t^T M)_j\)

(b) \(r_t = (d_t^T M)_{j_t}\)

(c) \(d_{t+1, i} = \frac{d_{t, i}}{1 + M_{i,j_t}r_t}\) for \(i = 1, \ldots, m\)

(a) Let \(d_t\) be the probability of error of weak classifier \(h_{j_t}\) at iteration \(t\). Define \(d_t\) as a summation over entries in \(M\).

(b) Write an expression for edge \(r_t\) in terms of \(d_t\).

(c) Assuming \(d_t\) is normalized, show that \(d_{t+1}\) remains normalized, i.e., \(\sum_i d_{t+1, i} = 1\).

(d) Show that Adaboost sets the edge of the previous weak classifier to 0, i.e., \((d_{t+1}^T M)_{j_t} = 0\).

3. **Problem 3**:

(a) Observe the \(M\) defined below, with 8 training points and 8 weak classifiers. As defined in Problem 1, the \(i\)\(^{\text{th}}\) column of \(M\) represents weak classifier \(i\) applied to the training points.

\[
M = \begin{pmatrix}
-1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\
1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\
\end{pmatrix}
\]
Assume that we start with the following initial distribution over the datapoints:

\[ \mathbf{d}_1 = \left( \frac{3 - \sqrt{5}}{8}, \frac{3 - \sqrt{5}}{8}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{\sqrt{5} - 1}{8}, \frac{\sqrt{5} - 1}{8}, 0 \right)^T \]

Perform Adaboost using the algorithm defined in Problem 2 using \( \mathbf{M}, \mathbf{d}_1 \), and \( t_{max} = 7 \). What weak classifier is picked at each round of boosting? Do you notice any pattern?

(b) What is the norm-1 margin produced by Adaboost for this example?

(c) Instead of using Adaboost, imagine we combined our classifiers using the following coefficients: \( [2, 3, 4, 1, 2, 1, 1] \times \frac{1}{M} \). What is the margin in this case? Does Adaboost maximize the margin?