A. Boosting

1. Implement AdaBoost with boosting stumps and apply the algorithm to the spambase data set of HW2 with the same training and test sets. Plot the average cross-validation error plus or minus one standard deviation as a function of the number of rounds of boosting $T$ by selecting the value of this parameter out of $\{10, 10^2, \ldots, 10^k\}$ for a suitable value of $k$, as in HW2. Let $T^*$ be the best value found for the parameter. Plot the error on the training and test set as a function of the number of rounds of boosting for $t \in [1, T^*]$. Compare your results with those obtained using SVMs in HW2.

2. Consider the following variant of the classification problem where, in addition to the positive and negative labels $+1$ and $-1$, points may be labeled with 0. This can correspond to cases where the true label of a point is unknown, a situation that often arises in practice, or more generally to the fact that the learning algorithm incurs no loss for predicting $-1$ or $+1$ for such a point. Let $X$ be the input space and let $Y = \{-1, 0, +1\}$. As in standard binary classification, the loss of $f : X \rightarrow \mathbb{R}$ on a pair $(x, y) \in X \times Y$ is defined by $1_y f(x) < 0$.

Consider a sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times Y)^m$ and a hypothesis set $H$ of base functions taking values in $\{-1, 0, +1\}$. For a base hypothesis $h_t \in H$ and a distribution $D_t$ over indices $i \in [1, m]$, define $\epsilon_t^s$ for $s \in \{-1, 0, +1\}$ by $\epsilon_t^s = \mathbb{E}_{i \sim D_t}[1_{y_i h_t(x_i) = s}]$.

(a) Derive a boosting-style algorithm for this setting in terms of $\epsilon_t^s$s, using the same objective function as that of AdaBoost. You should carefully justify the definition of the algorithm.

(b) What is the weak-learning assumption in this setting?

(c) Write the full pseudocode of the algorithm.

(d) Give an upper bound on the training error of the algorithm as a function of the number of rounds of boosting and $\epsilon_t^s$s.
B. On-line learning

The objective of this problem is to show how another regret minimization algorithm can be defined and studied. Let $L$ be a loss function convex in its first argument and taking values in $[0, M]$.

We will adopt the notation used in the lectures and assume $N > e^2$. Additionally, for any expert $i \in [1, N]$, we denote by $r_{t,i}$ the instantaneous regret of that expert at time $t \in [1, T]$, $R_{t,i} = L(\hat{y}_t, y_t) - L(y_{t,i}, y_t)$, and by $R_{t,i}$ his cumulative regret up to time $t$: $R_{t,i} = \sum_{s=1}^{t} r_{t,i}$. For convenience, we also define $R_{0,i} = 0$ for all $i \in [1, N]$. For any $x \in \mathbb{R}$, $(x)_+$ denotes $\max(x, 0)$, that is the positive part of $x$, and for $x = (x_1, \ldots, x_N)^\top \in \mathbb{R}^N$, $(x)_+ = ((x_1)_+, \ldots, (x_N)_+)^\top$.

Let $\alpha > 2$ and consider the algorithm that predicts at round $t \in [1, T]$ according to $\hat{y}_t = \sum_{i=1}^{n} w_{t,i} y_{t,i}$, with the weight $w_{t,i}$ defined based on the $\alpha$th power of the regret up to time $t-1$: $w_{t,i} = (R_{t-1,i})^{2-1}$. The potential function we use to analyze the algorithm is based on the function $\Phi$ defined over $\mathbb{R}^N$ by $\Phi: x \mapsto \|x\|_B^2 = \left[\sum_{i=1}^{N} (x_i)^q\right]^{\frac{2}{q}}$.

1. Show that $\Phi$ is twice differentiable over $\mathbb{R}^N - B$, where $B$ is defined as follows:
   $$B = \{ u \in \mathbb{R}^N : (u)_+ = 0 \}.$$

2. For any $t \in [1, T]$, let $r_t$ denote the vector of instantaneous regrets, $r_t = (r_{t,1}, \ldots, r_{t,N})^\top$, and similarly $R_t = (R_{t,1}, \ldots, R_{t,N})^\top$. We define the potential function as $\Phi(R_t) = \|R_t\|_B^2$. Computed $\nabla \Phi(R_{t-1})$ for $R_{t-1} \not\in B$ and show that $\nabla \Phi(R_{t-1}) \cdot r_t \leq 0$ (hint: use the convexity of the loss with respect to the first argument).

3. (Bonus question) Prove the inequality $r^\top [\nabla^2 \Phi(u)] r \leq 2(\alpha - 1)\|r\|_B^2$ valid for all $r \in \mathbb{R}^N$ and $u \in \mathbb{R}^N - B$ (hint: write the Hessian $\nabla^2 \Phi(u)$ as a sum of a diagonal matrix and a positive semi-definite matrix multiplied by $(2 - \alpha)$).

Also, use Hölder’s inequality generalizing Cauchy-Schwarz: for any $p > 1$ and $q > 1$ with $\frac{1}{p} + \frac{1}{q} = 1$ and $u, v \in \mathbb{R}^N$, $|u \cdot v| \leq \|u\|_p \|v\|_q$.

4. Using the answers to the two previous questions and Taylor’s formula, show that for all $t \geq 1$, $\Phi(R_t) - \Phi(R_{t-1}) \leq (\alpha - 1)\|r_t\|_B^2$, if $\gamma R_{t-1} + (1-\gamma)R_t \not\in B$ for all $\gamma \in [0, 1]$.

5. Suppose there exists $\gamma \in [0, 1]$ such that $(1-\gamma)R_{t-1} + \gamma R_t \in B$. Show that $\Phi(R_t) \leq (\alpha - 1)\|r_t\|_B^2$.

6. Using the two previous questions, derive an upper bound on $\Phi(R_T)$ expressed in terms of $T$, $N$, and $M$. 

7. Show that $\Phi(R_T)$ admits as a lower bound the square of the regret $R_T$ of the algorithm.

8. Using the two previous questions give an upper bound on the regret $R_T$. For what value of $\alpha$ is the bound the most favorable? Give a simple expression of the upper bound on the regret for a suitable approximation of that optimal value.