Machine Learning and Applications

Assignment 2

1. Alternative objective functions.

   This problem studies boosting-type algorithms defined with objective functions different from that of AdaBoost. We assume that the training data are given as m labeled examples $(x_1, y_1), \ldots, (x_m, y_m) \in X \times \{-1, +1\}$. We further assume that Φ is a strictly increasing convex and differentiable function over $\mathbb{R}$ such that: $\forall x \geq 0, \Phi(x) \geq 1$ and $\forall x < 0, \Phi(x) > 0$.

   (a) (2) Consider the loss function $L(\alpha) = \sum_{i=1}^{m} \Phi(-y_i g(x_i))$ where g is a linear combination of base classifiers, i.e., $g = \sum_{t=1}^{T} \alpha_t h_t$ (as in AdaBoost). Derive a new boosting algorithm using the objective function L. In particular, characterize the best base classifier h_u to select at each round of boosting if we use coordinate descent.

   (b) (2) Consider the following functions:

      i. zero-one loss $\Phi_1(-u) = 1_{u \leq 0}$;

      ii. least squared loss $\Phi_2(-u) = (1-u)^2$;

      iii. SVM loss $\Phi_3(-u) = \max\{0, 1-u\}$;

      iv. logistic loss $\Phi_4(-u) = \log(1 + e^{-u})$.

      Which functions satisfy the assumptions on Φ stated earlier in this problem?

   (c) (4) For each loss function satisfying these assumptions, derive the corresponding boosting algorithm. How do the algorithm(s) differ from AdaBoost?

2. (4) Weighted instances. Let the training sample be $S = ((x_1, y_1), \ldots, (x_m, y_m))$. Suppose we wish to penalize differently errors made on x_i versus x_j. To do that, we associate some non-negative importance weight w_i to each point x_i and define the objective function $F(\alpha) = \sum_{i=1}^{m} w_i e^{-y_i g(x_i)}$, where $g = \sum_{t=1}^{T} \alpha_t h_t$. Show that this function is convex and differentiable and use it to derive a boosting-type algorithm.

3. (2) Define the unnormalized correlation of two vectors $\mathbf{x}$ and $\mathbf{x}'$ as the inner product between these vectors. Prove that the distribution vector $(D_{t+1}(1), \ldots, D_{t+1}(m))$ defined by AdaBoost ($D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$, $D_1 = \frac{1}{m}$, Z_t - is a normalizatoin constant, i.e. $\sum_{i=1}^m D_t(i) = 1$) and the vector of components $y_i h_t(x_i)$ are uncorrelated.

4. (*) Noise-tolerant AdaBoost. AdaBoost may significantly overfitting in the presence of noise, in part due to the high penalization of misclassified examples. To reduce this effect, one could use instead the following objective function:

$$F = \sum_{i=1}^{m} G(-y_i g(x_i))$$

where G is the function defined on $\mathbb{R}$ by

$$G(x) = \begin{cases} e^x, & \text{if } x \leq 0 \\ x+1, & \text{otherwise.} \end{cases}$$

(a) (2) Show that the function G is convex and differentiable.

(b) (optional, extra 5 points) Use F and greedy coordinate descent to derive an algorithm similar to AdaBoost.