CIS 520, Machine Learning, Fall 2015: Assignment 4
Due: Thursday, October 15th, 11:59pm, PDF to Canvas

Instructions. Please write up your responses to the following problems clearly and concisely. We encourage you to write up your responses using LaTeX; we have provided a LaTeX template, available on Canvas, to make this easier. **Submit your answers in PDF form to Canvas. We will not accept paper copies of the homework.**

Collaboration. You are allowed and encouraged to work together. You may discuss the homework to understand the problem and reach a solution in groups up to size **two students.** However, *each student must write down the solution independently, and without referring to written notes from the joint session.* **In addition, each student must write on the problem set the names of the people with whom you collaborated.** You must understand the solution well enough in order to reconstruct it by yourself. (This is for your own benefit: you have to take the exams alone.)

MDL and Boosting [100 points]

The details of Adaboost are in *Robert E. Schapire. The boosting approach to machine learning: An overview.* In Nonlinear Estimation and Classification. Springer, 2003. http://www.cs.princeton.edu/~schapire/uncompress-papers.cgi/msri.ps (The AdaBoost notation in Bishop's *Pattern Recognition* is slightly different and might be confusing when solving this question, so you should use the Schapire paper as your main reference here.)

1. Analyzing the training error of boosting [60 points]

Consider the AdaBoost algorithm you saw in class. In this question we will try to analyze the training error of boosting.

1. **[8 points]** Given a set of m examples, (x_i, y_i) (y_i is the class label of x_i), $i = 1, \ldots, m$, let $h_t(x)$ be the weak classifier obtained at step t, and let α_t be its weight. Recall that the final classifier is

$$H(x) = \text{sign}(f(x)), \text{ where } f(x) = \sum_{t=1}^{T} \alpha_t h_t(x).$$

   Show that the training error of the final classifier can be bounded from above by an exponential loss function:

$$\frac{1}{m}\sum_{i=1}^{m} I(H(x_i) \neq y_i) \leq \frac{1}{m}\sum_{i=1}^{m} \exp(-f(x_i)y_i),$$

   where $I(a = b)$ is the indicator function. It is equal to 1 if $a = b$, and 0 otherwise

   Hint: $e^{-x} \geq 1 \Leftrightarrow x \leq 0$.

2. **[16 points]** Remember that

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Use this recursive definition to prove the following.

$$\frac{1}{m}\sum_{i=1}^{m}\exp(-f(x_i)y_i) = \prod_{t=1}^{T} Z_t, \tag{1}$$

where Z_t is the normalization factor for distribution D_{t+1}:

$$Z_t = \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i)). \tag{2}$$

Hint: Remember that $e^{\sum_i g_i} = \prod_i e^{g_i}$,$D_1(i) = \frac{1}{m}$, and that $\sum_i D_{t+1}(i) = 1$.

3. **[10 points]** Equation 1 suggests that the training error can be reduced rapidly by greedily optimizing Z_t at each step. You have shown that the error is bounded from above:

$$\epsilon_{training} \le \prod_{t=1}^{T} Z_t.$$

Observe that $Z_1, \ldots, Z_{t-1}$ are determined by the first $(t-1)$ rounds of boosting, and we cannot change them on round t. A greedy step we can take to minimize the training error bound on round t is to minimize Z_t.

In this question, you will prove that for binary weak classifiers, Z_t from Equation 2 is minimized by picking α_t as:

$$\alpha_t^* = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right), \tag{3}$$

where ϵ_t is the training error of weak classifier h_t for the weighted dataset:

$$\epsilon_t = \sum_{i=1}^{m} D_t(i) I(h_t(x_i) \ne y_i).$$

where I is the indicator function. For this proof, only consider the simplest case of binary classifiers, i.e. the output of $h_t(x)$ is binary, $\{-1, +1\}$.

For this special class of classifiers, first show that the normalizer Z_t can be written as:

$$Z_t = (1-\epsilon_t)\exp(-\alpha_t) + \epsilon_t \exp(\alpha_t).$$

Hint: Consider the sums over correctly and incorrectly classified examples separately.

4. **[8 points]** Now, prove that the value of α_t that minimizes this definition of Z_t is given by Equation 3.

5. **[4 points]** Prove that for the above value of α_t we have $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

6. **[8 points]** Furthermore, let $\epsilon_t = \frac{1}{2} - \gamma_t$ and prove that $Z_t \le \exp(-2\gamma_t^2)$. Therefore, we will know

$$\epsilon_{training} \le \prod_t Z_t \le \exp(-2\sum_t \gamma_t^2).$$

Hint: $\log(1-x) \le -x$ for $0 < x \le 1$.

7. **[4 points]** If each weak classifier is slightly better than random, so that $\gamma_t \ge \gamma$, for some $\gamma > 0$, then the training error drops exponentially fast in T, i.e.

$$\epsilon_{training} \le \exp(-2T\gamma^2).$$

Argue that in each round of boosting, there always exists a weak classifier h_t such that its training error on the weighted dataset $\epsilon_t \le 0.5$.

8. **[2 points]** Show that for $\epsilon_t = 0.5$ the training error can get "stuck" above zero. *Hint*: $D_t(i)$s may get stuck.

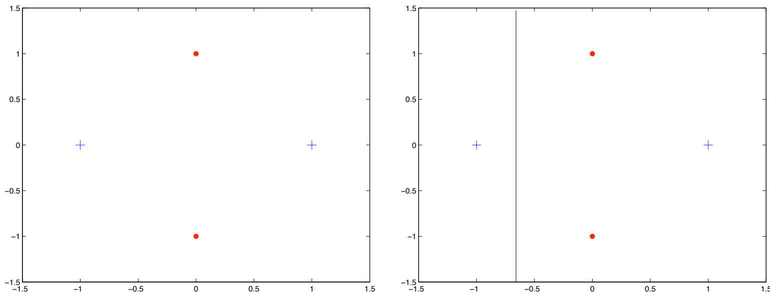

Figure 1: a) Toy data in Question . b) h_1 in Question

2. Adaboost on a toy dataset [20 points]

Now we will apply Adaboost to classify a toy dataset. Consider the dataset shown in Figure 1a. The dataset consists of 4 points: $(X_1 : 0, -1, -)$, $(X_2 : 1, 0, +)$, $(X_3 : -1, 0, +)$ and $(X_4 : 0, 1, -)$. For this part, you may find it helpful to use MATLAB as a calculator rather than doing the computations by hand. (You do not need to submit any code though.)

1. **[8 points]** For $T = 4$, show how Adaboost works for this dataset, using simple decision stumps (depth-1 decision trees that simply split on a single variable once) as weak classifiers. For each timestep compute the following:

$$\epsilon_t, \alpha_t, Z_t, D_t(i)\ \forall i,$$

   Also for each timestep draw your weak classifier. For example h_1 can be as shown in 1b).

2. **[4 points]** What is the training error of Adaboost for this toy dataset?

3. **[8 points]** Is the above dataset linearly separable? Explain why Adaboost does better than a decision stump on the above dataset.

3. MDL on a toy dataset [20 points]

Consider the following problem

We provide a data set generated from a particular model with $N = 64$.
where we want to estimate

$$\hat{y} = w_1x_1 + w_2x_2 + w_3x_3$$

We want to use MDL to find the 'optimal' L_0-penalized model. We will use the standard formulas $AIC = n\ log(Err_q/n) + 2q$, $BIC = n\ log(Err_q/n) + log(n)q$, which are 2 times the forms we usually use in class. (see slides 10 and 11 of `http://www.seas.upenn.edu/~cis520/lectures/MDL.pdf`)

1. **[10 points]** Estimate the three linear regressions

   (We could actually try all possible subsets here, but instead we'll just try three.)

$$\begin{aligned} y_1 &= w_1x_1 \\ y_2 &= w_1x_1 + w_2x_2 \\ y_3 &= w_1x_1 + w_2x_2 + w_3x_3 \end{aligned}$$

   For each of the three cases, what is

(a) the sum of square error
i) $\mathrm{Err}_1 =$
ii) $\mathrm{Err}_2 =$
iii) $\mathrm{Err}_3 =$

(b) 2 times the estimated bits to code the residual ($n \log \frac{Error}{n}$)
i) $\mathrm{ERR\_bits}_1 =$
ii) $\mathrm{ERR\_bits}_2 =$
iii) $\mathrm{ERR\_bits}_3 =$

(c) 2 times the estimated bits to code each residual plus model under AIC ($2 * 1$ bit to code each feature)
i) $\mathrm{AIC\_bits}_1 =$
ii) $\mathrm{AIC\_bits}_2 =$
iii) $\mathrm{AIC\_bits}_3 =$

(d) 2 times the estimated bits to code each residual plus model under BIC ($2 * (1/2) log(n)$ bits to code each feature)
i) $\mathrm{BIC\_bits}_1 =$
ii) $\mathrm{BIC\_bits}_2 =$
iii) $\mathrm{BIC\_bits}_3 =$

2. **[5 points]** Which model has the smallest minimum description length?
a) for AIC
b) for BIC

3. **[5 points]** Included in the kit is a test data set; does the error on the test set for the three models correspond to what is expected from MDLs?