CS 4375 - ASSIGNMENT 3

Please read the instructions below before starting the assignment.

- This assignment has two parts. The first part requires you to answer theoretical questions, while the second part involves coding the K-means unsupervised learning algorithm.
- Please create two different folders, named part I, and part II, and keep your files in the respective folders.
- For each of the code folders, please include a README file indicating how to compile and run your code. Also, mention clearly which packages/libraries you have used.
- You should use a cover sheet, which can be downloaded at: http://www.utdallas.edu/~axn112530/cs4375/CS4375_CoverPage.docx
- You are allowed to work in pairs i.e. a group of two students is allowed. Please write the names of the group members on the cover page.
- You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After that, there will be a penalty of 10% for each late day. **The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions through Piazza, and not through email.

PART I

(10 points)

1. Consider a regression problem of trying to estimate the function f: X -> y where X is a vector of feature attributes and y is a continuous real valued output variable. You would like to use a bagging model where you first create M bootstrap samples and then use them to create M different models – $h_1, h_2, \ldots, h_M$. You can assume that all models are of the same type.

The error for each of the models would be described as:

$$\epsilon_i(x) = f(x) - h_i(x)$$

where x is the input data and h_i is the model created using i^{th} bootstrap sample

The expected value of the squared error for any of the models will be defined as:

$$E(\varepsilon_i(x)^2) = E[\left(f(x) - h_i(x)\right)^2]$$

The average value of the expected squared error for each of the models acting individually is defined as:

$$E_{avg} = \frac{1}{M}\sum_{i=1}^{M} E(\varepsilon_i(x)^2)$$

Now, you decide to aggregate the models using a committee approach as follows:

$$h_{agg}(x) = \frac{1}{M}\sum_{i=1}^{M} h_i\,(x)$$

The error using the aggregated model is defined as:

$$E_{agg}(x) = E[\left\{\frac{1}{M}\sum_{i=1}^{M} h_i\,(x) - f(x)\right\}^2]$$

which can be simplified as:

$$E_{agg}(x) = E[\left\{\frac{1}{M}\sum_{i=1}^{M} \epsilon_i\,(x)\right\}^2]$$

where we used the value of ε_i is defined above.

Prove that

$$E_{agg} = \frac{1}{M} E_{avg}$$

provided you make the following assumptions:

1. Each of the errors have a 0 mean

$$E(\epsilon_i(x)) = 0 \text{ for all i}$$

2. Errors are uncorrelated

$$E\big(\epsilon_i(x)\epsilon_j(x)\big) = 0 \text{ for all i} \neq \text{j}$$

$$E_{agg}(x) = \left(\frac{1}{M}\right)^2 E\left[\left(\sum \varepsilon_i(x)\right)^2\right]$$

$$= \frac{1}{M^2} E\left[(\varepsilon_1)^2 + \cdots + (\varepsilon_n)^2 + \underbrace{(\varepsilon_1)(\varepsilon_2) + \cdots + (\varepsilon_{n-1})(\varepsilon_n)}_{=0}\right]$$

$$= \frac{1}{M^2} E\left[(\varepsilon_1)^2 \cdots (\varepsilon_n)^2\right]$$

$$= \frac{1}{M^2} E\left[\sum (\varepsilon_i(x))^2\right]$$

$$= \frac{1}{M} \cdot \underbrace{\frac{1}{M} \cdot E\left[\sum (\varepsilon_i(x))^2\right]}_{E_{avg}}$$

$$= \frac{1}{M} \cdot E_{avg}$$

$\square$

(10 points)
2. Jensen's inequality states that for any *convex* function *f*:

$$f(\sum_{i=1}^{M} \lambda_i x_i) \leq \sum_{i=1}^{M} \lambda_i f(x_i)$$

In question 1, we had assumed that each of the errors are uncorrelated i.e.

$$E\big(\epsilon_i(x)\epsilon_j(x)\big) = 0 \text{ for all i} \neq \text{j}$$

This is not really true, as the models are created using bootstrap samples and have correlation with each other. Now, let's remove that assumption. Show that using Jensen's inequality, it is still possible to prove that:

$$E_{agg} \leq E_{avg}$$

$$E_{agg}(x) = \left(\frac{1}{M}\right)^2 E\left[\left(\sum \varepsilon_i(x)\right)^2\right]$$

$$= \frac{1}{M^2} E\left[(\varepsilon_1)^2 + \cdots + (\varepsilon_n)^2 + \underbrace{(\varepsilon_1)(\varepsilon_2) + \cdots + (\varepsilon_{n-1})(\varepsilon_n)}_{\neq 0}\right]$$

However, we can see that $\left(\frac{\sum \varepsilon_i(x)}{M}\right)^2$

is like z^2 w/ $z = \frac{\sum \varepsilon_i(x)}{M}$

z^2 is a convex function

$\therefore$ z^2 follows Jensen's inequality

so

$$E\left(\frac{\sum \varepsilon_i(x)}{M}\right)^2 \leq E\left(\frac{\sum (\varepsilon_i(x))^2}{M}\right)$$

$\square$

(10 points)
3. Deriving the training error for AdaBoost:
In class, we discussed the steps of Adaboost algorithm. Recall that the final hypothesis for a Boolean classification problem at the end of T iterations is given by:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

The above equation says that the final hypothesis is the weighted hypothesis generated at the end of each individual step.
Also recall that the weight for the point i at step t+1 is given by:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times e^{-\alpha_t h_t(i) y(i)}$$

where:
$D_t(i)$ is the normalized weight of point i in step t
$h_t(i)$ is the hypothesis (prediction) at step t for point i
α_t is the final "voting power" of hypothesis h_t
$y(i)$ is the true label for point i
Z_t is the normalization factor at step t (it ensures that the weights sum up to 1.0)

Note that at step 1, the points have equal weight

$$D_1 = \frac{1}{N}$$

where N is the total number of data points.
At each of the steps, the total error of h_t will be defined as $\varepsilon_t = \frac{1}{2} - \gamma_t$, which is a way of saying that the error will be better than 50% by a value γ_t.

Prove that at the end of T steps, the overall training error will be bounded by:

$$\exp(-2\sum_{i=1}^{T} \gamma_t^2)$$

That is, the overall training error of the hypothesis H will be less than or equal to the amount indicated above.

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

$$= \underbrace{D_1(i)}_{\frac{1}{N}} \cdot \frac{e^{-\alpha_1 y_i h_1(x_i)}}{Z_1} \times \cdots \times \frac{e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

$$= \frac{1}{N} \frac{e^{-y_i \left(\sum_{t=1}^{T} \alpha_t h_t(x_i) \right)}}{\prod Z_t} = H(x)$$

$$= \frac{1}{N} \frac{e^{-y_i H(x)}}{\prod Z_t}$$

if $y_i \neq H(x) \Rightarrow$ error $= 1$

else $= 0$

$$\frac{\frac{1}{N} \sum e^{-y_i H(x)}}{\prod Z_t} = D_{T+1}(i)$$

$\therefore$

$$\text{training error} \leq \frac{1}{N} \sum e^{-y_i H(x)} = \underbrace{D_{T+1}(i)}_{1} \cdot \prod Z_t$$

$$= \prod Z_t$$

$$Z_t = \underbrace{\sum D_t(i)}_{h_t(x) = y_i \;=\; 1-\varepsilon} \cdot e^{-\alpha_t} + \underbrace{\sum D_t(i)}_{h_t(x) \neq y_i \;=\; \varepsilon} e^{\alpha_t}$$

$$= e^{-\alpha_t}(1-\varepsilon_t) + e^{\alpha_t} \varepsilon_t$$

Take $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon}{\varepsilon} \right)$

$$= e^{-\ln\left(\left(\frac{1-\varepsilon}{\varepsilon}\right)^{\frac{1}{2}}\right)} (1-\varepsilon_t) + e^{\ln\left(\left(\frac{1-\varepsilon}{\varepsilon}\right)^{\frac{1}{2}}\right)} \varepsilon_t$$

$$= \left(\frac{\varepsilon_t}{1-\varepsilon_t} \right)^{\frac{1}{2}} (1-\varepsilon_t) + \left(\frac{1-\varepsilon_t}{\varepsilon} \right)^{\frac{1}{2}} \varepsilon_t$$

$$= \left(\frac{\varepsilon_t (1-\varepsilon_t)^{\cancel{2}}}{\cancel{1-\varepsilon_t}}\right)^{\frac{1}{2}} + \left(\frac{(1-\varepsilon_t)\varepsilon_t^{\cancel{2}}}{\cancel{\varepsilon_t}}\right)^{\frac{1}{2}}$$

$$= (\varepsilon_t (1-\varepsilon_t))^{\frac{1}{2}} + (\varepsilon_t (1-\varepsilon_t))^{\frac{1}{2}}$$

$$= 2(\varepsilon_t (1-\varepsilon_t))^{\frac{1}{2}}$$

$$= 2\left(\left(\frac{1}{2} - \gamma_t\right)\left(1 - \left(\frac{1}{2} - \gamma_t\right)\right)\right)^{\frac{1}{2}}$$

$$= 2\left(\left(\frac{1}{2} - \gamma_t\right)\left(\frac{1}{2} + \gamma_t\right)\right)^{\frac{1}{2}}$$

$$= 2\left(\frac{1}{4} - \gamma_t^2\right)^{\frac{1}{2}}$$

$$= (1 - 4\gamma_t^2)^{\frac{1}{2}}$$

$$\le e^{-2\gamma_t^2}$$

Since $\prod Z_t \le e^{-2\sum \gamma_t^2}$

& training error $\le \prod Z_t$

training error $\le e^{-2\sum \gamma_t^2}$