CSE 353 - Machine Learning

Assignment 1

1 The "warm-up"

1. Assume that there exists a square S_{large} with side of length 5, with corners located at $(0,0),(0,5),(5,0),(5,5)$. Consider instances $\{x_i = (a_i, b_i) \in S_{large}\}$ are drawn from a uniform distribution $\mathcal{D}$ over S_{large}.
   Write down the feature vector for the given instances

$$\mathbf{x} = \begin{bmatrix} 3 & 1 \\ 4 & 1 \\ 3 & 4 \\ 4 & 4 \\ 2.5 & 2.5 \end{bmatrix}$$

2. The true labeling function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{0, 1\}$, assigns label 1 to instances within the smaller square (blue) and 0 to other instances (red). The smaller square S_{small} is of side length 1 and is located at the center of S_{large}. Fig. 1 shows $m = 5$ data points used as training data: $S = \{x_i, i = 0, ..., m\}$. Formally, write down the training data S.

   $S = (((3,1),0),((3,4),0),((4,1),0),((4,4),0),((2.5,2.5),1))$

3. The hypothesis for prediction h_l, is defined over S as

$$h(x) = \begin{cases} 1 \text{ if } \exists i \in [0,m] \text{ such that } x_i = x \text{ and for } x_i = (a_i, b_i), a_i > l \\ 0 \text{ otherwise} \end{cases}$$

   (a) What is the *training error* and *true error*?

	training error	*true error*
h_2	$\frac{4}{5}$ or 0.8	$\frac{1}{25}$
h_3	$\frac{3}{5}$ or 0.6	$\frac{1}{25}$
h_5	$\frac{1}{5}$ or 0.2	$\frac{1}{25}$

   (b) Using empirical risk minimization, choose the best hypothesis. Formally show that this choice is, indeed, the best. Do any of the above hypotheses suffer from overfitting?
   According to the principle of empirical risk minimization, the hypothesis with the lowest training error or empirical risk would be best, which in this case would be h_5. However, the true error are the same for all, even if the training data seems to indicate h_5 is the best.
   None of the hypotheses suffer from overfitting since they are not fitted to the sample set.

4. Consider instances of X drawn from the uniform distribution $\mathcal{D}$ on $[-1, 1]$. Let f denote the actual labelling function mapping each instance to its label $y \in \{0, 1\}$ with probabilities
$P(Y = 1|x > 0) = 0.9$
$P(Y = 0|x > 0) = 0.1$
$P(Y = 1|x < 0) = 0.1$
$P(Y = 0|x < 0) = 0.9$
The hypothesis h predicts the label for each instance as defined below

$$h(x) = \begin{cases} 1 \text{ if x>0} \\ 0 \text{ otherwise} \end{cases}$$

Measure the success of this predictor by calculating the training error for h.
The training error, $\mathcal{L}_s(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$.
$\mathcal{L}_s = P(h(x) \neq Y \wedge x < 0) + P(h(x) \neq Y \wedge x > 0)$
$\mathcal{L}_s = P(h(x) \neq Y|x < 0) * P(x < 0) + P(h(x) \neq Y|x > 0) * P(x > 0)$
$\mathcal{L}_s = 0.1 * 0.5 + 0.1 * 0.5$
$\mathcal{L}_s = 0.1$

2 Learning Models

1. Let $\mathcal{H}$ be a hypothesis class for a binary classification task. Let $m_{\mathcal{H}}(\epsilon, \delta)$ denote its sample complexity (depending on the two parameters ϵ and δ). Show that $m_{\mathcal{H}}$ is monotonically non-increasing in each parameter.

The sample complexity function $m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{log(\frac{|\mathcal{H}|}{\delta})}{\epsilon} \rceil$.
If we consider the function in terms of only ϵ, $m_{\mathcal{H}} = \frac{i}{\epsilon}$ where $i = log(\frac{|H|}{\delta})$
If we consider the function in terms of only δ, $m_{\mathcal{H}} = j * log(\frac{k}{\delta})$, where $j = \frac{1}{\epsilon}$, $k = |\mathcal{H}|$.
The sample complexity is inversely proportional to both, so as either parameter approaches infinity, $m_{\mathcal{H}}$ approaches 0. This can be rationalized - as the need for a more confidence or a more precise hypothesis decreases, then fewer training samples are needed.

2. Let $X = \mathbb{R}^2, Y = \{0, 1\}$, and let $\mathcal{H} = h_r : r \in \mathbb{R}_+$, where h_r is the indicator function $h_r(x) = 1_{||x|| \leq r}$. Prove that under the realizability assumption, $\mathcal{H}$ is PAC learnable. Also show that its sample complexity is bounded above as follows:

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{log(\frac{1}{\delta})}{\epsilon} \rceil$$

According to the definition of PAC learnability, a hypothesis is PAC learnable if there exists a sample complexity function $m_{\mathcal{H}} : (0, 1)^2 \to \mathbb{N}$ and an algorithm that when run on a data set of size m where $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, the algorithm returns a hypothesis h such that $L_{(\mathcal{D},f)}(h) \leq \epsilon$ with probability greater than $1 - \delta$. ϵ represents the accuracy of the

hypothesis while δ represents the confidence of the hypothesis' accuracy.

The described hypothesis class consists of a hypothesis that for some positive real, r, returns 1 when the l_2 norm of the input is less than or equal to r, essentially the class of concentric circles on the plane with radius r with the center at the origin.

Since the domain is for this problem is infinite, for any hypothesis h_r with any arbitrary $r \in R_+$, the size of the "error region" (the area difference between the concentric circles represented by the chosen hypothesis and the labelling function) where misclassifications occur is finite and the true error approximates to 0: $L_{\mathcal{D}} = 0$. So for any $\delta, \epsilon \in (0, 1)$, we can choose any hypothesis and it will always be the case that $L_{(\mathcal{D},f)}(h_r) \leq \epsilon$. Hence, the hypothesis is PAC learnable under the realizability assumption.

Since any hypothesis with an $r \in \mathbb{R}_+$ will satisfy the requirements that $L_{(\mathcal{D},f)}(h_r) \leq \epsilon$, no training samples are required to pick a suitable hypothesis. As a result we set $m_{\mathcal{H}} = 0$. To complete this proof, we simply need to show $0 \leq \lceil \frac{log(\frac{1}{\delta})}{\epsilon} \rceil$

$$\lceil \frac{log(\frac{1}{\delta})}{\epsilon} \rceil \geq \frac{log(\frac{1}{\delta})}{\epsilon}$$

Since $\delta, \epsilon \in (0, 1)$.

$$\frac{log(\frac{1}{\delta})}{\epsilon} \geq \frac{0}{\epsilon} = 0$$

So

$$0 = m_{\mathcal{H}} \leq \lceil \frac{log(\frac{1}{\delta})}{\epsilon} \rceil$$

3. Let $\mathcal{X}$ be a domain and let $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m$ be a sequence of distributions over $\mathcal{X}$ and let $f \in \mathcal{H}$. Suppose we are getting a sample $\mathcal{S}$ of m examples such that the instances are independent but not identically distributed, the i^{th} instance is sampled from $\mathcal{D}_i$, and then y_i is set to be $f(x_i)$. Let $\overline{\mathcal{D}}_m = (\mathcal{D}_1 + ... + \mathcal{D}_m)/m$.
Fix an accuracy parameter $\epsilon \in (0, 1)$. Show that

$$\mathbb{P}[\exists h \in \mathcal{H} \text{ s.t. } L_{(\overline{\mathcal{D}}_{m,f})}(h) > \epsilon \text{ and } L_{(S,f)}(h) = 0] \leq |\mathcal{H}|e^{-\epsilon m}$$

Since $\mathcal{H}$ is a finite hypothesis class containing the labelling function, it is PAC learnable. In other words, there exists an algorithm that can find a hypothesis with a true error of less than ϵ with probability $1 - \delta$ when run on $m \geq m_{\mathcal{H}}$ samples where

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{log(\frac{|\mathcal{H}|}{\delta})}{\epsilon} \rceil$$

We will first define a subset of hypotheses that have a true error greater than ϵ, $\mathcal{H}_B = \{h : L_{\mathcal{D},f} > \epsilon\}$.

We also define the set of samples for which the bad hypotheses have an empirical error of 0 $M = \{S|_x : \exists h \in \mathcal{H}_B, L_S = 0\}$
The set of samples for which the learner returns a bad hypothesis is a subset of M, $\{S|_x : L_{\mathcal{D},f} > \epsilon\} \subseteq M$
The probability of a bad hypothesis on the size-m sample is lower than the probability of any of the hypotheses in M being selected. $D^m(\{S|_x : L_{\mathcal{D},f}(h) > \epsilon\}) \leq D^m(\cup_{h \in H}\{S|_x : L_S(h) = 0\})$
Since $P(a \cup b) \leq P(a) + P(b)$, we can say $D^m(\{S|_x : L_{\mathcal{D},f}(h) > \epsilon\}) \leq \sum_{h \in H_B} D^m(\{S|_x : L_S(h) = 0\})$. We will refer to this as equation (A) later.
For each sample data point, since it is independently distributed, $D^m(\{S|_x : L_S(h) = 0\}) = \Pi_{i=0}^m D_i(\{x_i : h(x_i) = y_i\})$
For a single data point, since the hypothesis comes from $\mathcal{H}_B = \{h : L_{\mathcal{D},f} > \epsilon\}$, $\Pi_{i=0}^m D_i(\{x_i : h(x_i) = y_i\})$
Since $\epsilon, 1 - \epsilon \in (0,1) \rightarrow 1 - \epsilon \leq e^{-\epsilon}$, $[\Pi_{i=0}^m (1-\epsilon) = (1-\epsilon)^m] \leq \sum_{h \in H_B} D^m(\{S|_x : L_S(h) = 0\}) \leq e^{-\epsilon m}$
Plug this into equation A: $D^m(\{S|_x : L_{\mathcal{D},f}(h) > \epsilon\}) \leq |H_B| e^{-\epsilon m} \leq |H| e^{-\epsilon m}$
Since the hypothesis comes from $\mathcal{H}_B$: $D^m(\{S|_x : L_{\mathcal{D},f}(h) > \epsilon \text{ and } L_{(S,f)}(h) = 0\}) \leq |H| e^{-\epsilon m}$
This defines the bounds on the size m sample such that the probability of a hypothesis with a true error greater than ϵ and the empirical error is 0.

$$\mathbb{P}[\exists h \in \mathcal{H} \text{ s.t. } L_{(\overline{\mathcal{D}}_{m,f})}(h) > \epsilon \text{ and } L_{(S,f)}(h) = 0] \leq |\mathcal{H}| e^{-\epsilon m}$$

4. Let $\mathcal{H}$ be a hypothesis class of binary classifiers. Show that if $\mathcal{H}$ is agnostic PAC learnable, then $\mathcal{H}$ is PAC learnable as well. Furthermore, if A is a successful agnostic PAC learner for $\mathcal{H}$, then A is also a successful PAC learner for $\mathcal{H}$.

   According to the definition of PAC learnability, a hypothesis is PAC learnable if there exists a sample complexity function $m_{\mathcal{H}} : (0,1)^2 \rightarrow \mathbb{N}$ and an algorithm that for every $\epsilon, \delta \in (0,1)$, distribution $\mathcal{D}$ over $\mathcal{X}$, and labelling function $f : \mathcal{X} \rightarrow \{0,1\}$, ***if*** the realizability assumption holds with respect to $\mathcal{H}, \mathcal{D}, f$, then when running the learning algorithm on $m > m_{\mathcal{H}}(\epsilon, \delta)$ independently and identically drawn from $\mathcal{D}$ and labelled by f, the algorithm returns a hypothesis h such that with probability greater than $1 - \delta$, $L_{(\mathcal{D},f)}(h) \leq \epsilon$.

   According to the definition of agnostic PAC learnability, a hypothesis is agnostic PAC learnable if there exists a sample complexity function $m_{\mathcal{H}} : (0,1)^2 \rightarrow \mathbb{N}$ and an algorithm that for every $\epsilon, \delta \in (0,1)$, distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, then when running the learning algorithm on $m > m_{\mathcal{H}}(\epsilon, \delta)$ independently and identically drawn from $\mathcal{D}$, the algorithm returns a hypothesis h such that with probability greater than $1 - \delta$, $L_{(\mathcal{D},f)}(h) \leq min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$.

   If we know that $\mathcal{H}$ is agnostic PAC learnable, there already exists a $m_{\mathcal{H}}^{agnostic}(\epsilon, \delta)$ that generates a minimal sample complexity. As stated in the definition of PAC learnability, if the realizability condition is met, then $m_{\mathcal{H}}^{agnostic}(\epsilon, \delta) \leq 0 + \epsilon$, which is the

same as the sample complexity function for PAC learning. Since PAC learning requires a labelling function and distribution over $\mathcal{X}$ while agnostic PAC learning uses a distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, we let the first terms for all points in $\mathcal{Z}$ serve as the domain $\mathcal{X}$, $X = \{x_i \forall z_i = (x_i, y_i) \in \mathcal{Z}\}$, and the labelling function $f(x)$ return y_i if $\exists z_i = (x_i, y_i) \in \mathcal{Z}, x_i = x$. If the realizability assumption is not met, it still logically completes the definition of PAC learnable. Thus, since it is agnostic PAC learnable, it is also PAC learnable.

If A is a successful agnostic PAC learner for $\mathcal{H}$, then when run on a sample of $m_{\mathcal{H}}^{agnostic}(\epsilon, \delta) \leq 0 + \epsilon$, it returns a hypothesis h such that with probability greater than $1 - \delta$, $L_{(\mathcal{D},f)}(h) \leq min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$. Since we have already proposed a scheme for creating a labelling function and domain suitable for PAC learning from the domain for A, if the realizability assumption holds, with probability greater than $1 - \delta$, $L_{(\mathcal{D},f)}(h) \leq min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$. If it doesn't hold, then it is still by definition a successful PAC learner.

5. Show that for every probability distribution $\mathcal{D}$, the Bayes optimal predictor $f_{\mathcal{D}}$ is optimal, in the sense that for every classifier g from $\mathcal{X}$, $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$.

   To simplify this, we can break this down in a case-by-case basis.

   - When $P(y = 1) \geq 0.5$
     $f_{\mathcal{D}}(x) = 1$, so $P[f_{\mathcal{D}}(x) = y] = P(y = 1)$
     $0 \leq g(x) \leq 1$, so $P[g(x) = y] \leq P(y = 1)$
   - When $P(y = 0) \geq 0.5$
     $f_{\mathcal{D}}(x) = 0$, so $P[f_{\mathcal{D}}(x) = y] = P(y = 0)$
     $0 \leq g(x) \leq 1$, so $P[g(x) = y] \leq P(y = 0)$

   For each subset of the outcome space, the probability of success for the Bayes optimal classifier is greater than or equal to the the probability of success for any other classifier - $P(f_{\mathcal{D}}(x) = y) \geq P(g(x) = y)$.
   Since the true error $\mathcal{L}_{\mathcal{D}}(h) = 1 - P(h = y)$

$$P(f_{\mathcal{D}}(x) = y) \geq P(g(x) = y)$$

$$1 - P(f_{\mathcal{D}}(x) = y) \leq 1 - P(g(x) = y)$$

$$L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$$

3 Learning Algorithms

1. Explain briefly (no more than 3-4 sentences) how the k-nearest neighbors algorithm can be used for both classification and regression.
   The k-nearest neighbors algorithm works using some function called a metric fuction to identify the degree of difference between any two given data points based on the domain values. It uses the metric function to identify k points in the training set most

similar to the input and assigns an output based on the labels of these "neighboring" points. In a classification system, the label may be chosen from neighboring points based while regression systems may choose an average or center value of the neighbors.

2. Show that the ERM problem of linear regression with respect to the absolute value loss function $l(h,(x,y)) = |h(x) - y|$ can be solved as a linear program. Show this by explicity casting the ERM problem as a linear program.

   The ERM problem associated with this problem is

   $$argmin_w \frac{1}{m} \sum_{i=1}^{m} | < w, x_i > -y_i|$$

   The nature of the absolute value means that this loss function is not differentiable, making it difficult to minimize. As a result, we introduce a vector u of same size as w.

   $$argmin_u \frac{1}{m} \sum_{i=1}^{m} u_i, -u_i \leq (< w, x_i > -y_i), (< w, x_i > -y_i) \leq u_i$$

3. Let us modify the perceptron learning algorithm as follows:
   In the update step, instead of setting $w^{(t+1)} = w^t + y_i x_i$ every time there is a misclassification, we set $w^{t+1} = w^t + \eta y_i x_i$ for some fixed $\eta > 0$. Show that this modified perceptron will

   - perform the same number of iterations as the original, and
   - converge to a vector that points to the same direction as the output of the original perceptron

   If the data is separable, the perceptron stops when it has separated the examples.
   Let w^* be a vector that $\{\forall i \in [m], (y_i < w^*, x_i >) \geq 1\}$
   So after some iteration t,

   $$< w^*, w^{(t+1)} > - < w^*, w^{(t)} > =< w^*, w^{(t+1)} - w^{(t)} >$$

   $$=< w^*, (w^t + \eta y_i x_i) - w^{(t)} >$$

   $$=< w^*, \eta y_i x_i >$$

   $$= \eta y_i < w^*, x_i >$$

   $$\geq \eta$$

   After T iterations,

   $$[\sum_{t=1}^{T} (< w^*, w^{(t+1)} > - < w^*, w^t >) =< w^*, w^{(T+1)} >] \geq \eta T$$

We will later refer to the above as equation A
We then consider the norms:

$$||w^{(t+1)}||^2 = ||w^{(t)} + \eta y_i x_i||^2 = ||w^{(t)}||^2 + 2\eta y_i < w^{(t)}, x_i > +\eta^2 y_i^2 ||x_i||^2$$

if we let $R = max||x_i||$ and consider the fact that the perceptron only iterates when $[\eta y_i < w^{(t)}, x_i >] \leq 0$ and η is positive

$$[||w^{(t+1)}|| = ||w^{(t)}||^2 + 2\eta y_i < w^{(t)}, x_i > +\eta^2 y_i^2 ||x_i||^2] \leq ||w^{(t)}||^2 + \eta^2 R^2$$

We apply this after T iterations,

$$||w^{(T+1)}||^2 \leq T\eta^2 R^2$$

$$||w^{(T+1)}|| \leq \sqrt{T}\eta R$$

Combine this with equation A and let $B = ||w^*||$, just as we do when proving the upper bound for this algorithm using an update step of $w^{(t+1)} = w^t + y_i x_i$.

$$\frac{< w^{(T+1)}, w^* >}{||w^*||||w^{(T+1)}||} \geq \frac{\eta T}{B\eta\sqrt{T}R} = \frac{\sqrt{T}}{BR}$$

The Cauchy Shwarz inequality states that the left side is less than 1.

$$1 \geq \frac{< w^{(T+1)}, w^* >}{||w^*||||w^{(T+1)}||} \geq \frac{T}{B\sqrt{T}R} = \frac{\sqrt{T}}{BR}$$

$$1 \geq \frac{\sqrt{T}}{BR}$$

or equivalently

$$T \leq (RB)^2$$

We have shown that even for values of $\eta \neq 1$, the number of iterations T using the perceptron update step $w^{t+1} = w^t + \eta y_i x_i$ is still bounded by the same value: $T \leq (RB)^2$, where $R = max_i||x_i||, B = min\{||w|| : \forall i \in [m], y_i < w, x_i >\geq 1\}, \eta > 0$.

Furthermore, in the process, we have shown that after the final iteration T,

$$< w^*, w_\eta^{(T+1)} >= \sum_{t=1}^{T} (< w^*, w_\eta^{(t+1)} > - < w^*, w^t >)$$

$$= \sum_{t=1}^{T} \eta y_i < w^*, x_i >$$

$$= T\eta \sum_{t=1}^{T} y_i < w^*, x_i >$$

$$= T\eta < w^*, w^{(T+1)} >$$

Since $T, \eta > 0$, this scalar multiplication of the final vector of the original perceptron has the same direction.