CIS 519: Homework 3

{Ian MacDonald}

11/11/20

Although the solutions are my own, I consulted with the following people while working on this homework: {Sally Hu, So Han}

Short Questions

(a) (1) (b)

(2) (d)

(b) (1) (A)

(2) Since we have N possibilities for a, and N possibilities for b, we have a concept class C of size N^2. If we want to learn a function in C, our hypothesis space would be all concepts in C, and therefore our hypothesis space would be of size N^2 as well. Given that $log(|H|) >= VC(H)$, $log(N^2) >= VC(H)$, meaning that the order of magnitude of the VC dimension of C is O(log N)

(3) (B)

(c) (1) (B)

(2) Since the training error is bounded by $e^{-2\gamma^2 T}$, as T increases the bound on the training error decreases as well, and will eventually reach 0 after a certain amount of iterations. Since we know that there were e^{14} examples in this data set, and the absolute smallest number of incorrect examples we can have without a training error of 0 is 1, then the smallest training error that we can achieve before a training error of 0 is e^{-14}. Knowing this, we can set the training error bound equal to this, and then solve for T to determine the number of iterations that would bound the error at e^-14.
$e^{-14} <= e^{-2(1/4)^2 T}$
$e^{-14} <= e^{-(1/8)T}$
$-14 <= -(1/8)T$
$T >= 112$
After 112 iterations the maximum bound on the error would be e^{-14},

so the best numeric bound I can give would be 113 iterations, since at about that amount of iterations the maximum error bound would be approximately 0 and our algorithm should halt.

Boosting

Fill in the boosting table below.

i	Label	Hypothesis 1				Hypothesis 2			
		D_0	$f_1 \equiv [x > 5\,]$ $\epsilon = .3$	$f_2 \equiv [y > 0\,]$ $\epsilon = .5$	$h_1 \equiv [\,f_1\,]$	D_1	$f_1 \equiv [x > 1\,]$ $\epsilon = .381$	$f_2 \equiv [y > 0\,]$ $\epsilon = .4525$	$h_2 \equiv [\,f_1\,]$
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	+	.1;	+;	+;	+;	.0715;	+;	+;	+;
2	+	.1;	−;	+;	−;	.1665;	+;	+;	+;
3	+	.1;	−;	+;	−;	.1665;	+;	+;	+;
4	+	.1;	+;	+;	+;	.0715;	+;	+;	+;
5	+	.1;	+;	+;	+;	.0715;	+;	+;	−;
6	−	.1;	−;	+;	−;	.0715;	−;	+;	+;
7	−	.1;	−;	+;	−;	.0715;	+;	+;	+;
8	−	.1;	−;	+;	−;	.0715;	+;	+;	+;
9	−	.1;	+;	+;	+;	.1665;	+;	+;	+;
10	−	.1;	−;	+;	−;	.0715;	+;	+;	+;

The final hypothesis is

$$H_{\text{final}} = .61h_1 + .35h_2 \tag{1}$$

Subset of work done to arrive at above answers:

$D_0 = .1$ for all

ϵ calculated by finding the number of misclassified examples and then weighting them according to D_0, which is .1 for all cases initially, so all examples are weighted equally.

for x=0, $\epsilon = .5$
for x=1, $\epsilon = .4$
for x=2, $\epsilon = .4$ ...

for y=0, $\epsilon = .5$
for y=1, $\epsilon = .6$
for y=2, $\epsilon = .7$ ...

The lowest error was found in f_1, with multiple ties at $\epsilon = .3$, so x=5 was chosen arbitrarily

$a_0 = \frac{1}{2} log_2((1-\epsilon_t)/\epsilon_t)$
$a_0 = \frac{1}{2} log_2((1-.3)/.3)$
$a_0 = \frac{1}{2} log_2(.7/.3)$
.61

$z_0 = \sum_i D_0(i) * 2^{-(.61 y_i h_t(x_i))}$
$\frac{7}{10} 2^{-.61} + \frac{3}{10} 2^{.61}$
.4586 + .4579
.9165

Finding $D_1(i)$ for all i:
Since i=1 is predicted correctly, $D_1(1) = D_0(i) * 2^{-a_0} = .0715$
Since i=2 is not predicted correctly, $D_1(2) = D_0(i) * 2^{a_0} = .1665$
Above repeated for all i = 1,2,3...

Second round done identical to first round, but using D_1 instead of D_0

for x=0, $\epsilon = .4525$
for x=1, $\epsilon = .381$
for x=2, $\epsilon = .476$ ...

for y=0, $\epsilon = .4525$
for y=1, $\epsilon = .619$
for y=2, $\epsilon = .7855$ ...
The lowest error was found in f_1, with $\epsilon = .381$ at x=1

$a_1 = \frac{1}{2} log_2((1-\epsilon_t)/\epsilon_t)$
$a_1 = \frac{1}{2} log_2((1-.381)/.381)$
$a_1 = \frac{1}{2} log_2(1.632)$
.353

SVMs

(a) 1. $\mathbf{w} = [-1, -1]$, $\theta = 0$

2. $\mathbf{w} = [\frac{-1}{2}, \frac{-1}{2}]$, $\theta = 0$ With these values for w and theta, the hyperplane $w^T x + \theta = 0$ splits the difference between points 2 and 4 evenly, which are the 2 closest points to our linear classifier and thus would be used as the support vectors. Additionally, this is the smallest weight vector that still satisfies the constraints of SVMs, namely that $y(w^T x + \theta) >= 1$.

(b) 1. I = {2,4}

2. $\alpha_1 = \frac{1}{4}, \alpha_2 = \frac{1}{4}$

$[\frac{-1}{2}, \frac{-1}{2}] = \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2$
$[\frac{-1}{2}, \frac{-1}{2}] = \alpha_1 * 1 * [-2, 0] + \alpha_2 * -1 * [0, -2]$
$[\frac{-1}{2}, \frac{-1}{2}] = [-2\alpha_1, -2\alpha_2]$
$\alpha_1 = \frac{1}{4}, \alpha_2 = \frac{1}{4}$

3. Objective function value: $\frac{1}{4}$
$\frac{1}{2}||w||^2 = \frac{1}{2} w^T w$
$\frac{1}{2}||w||^2 = \frac{1}{2}(\frac{-1}{2}^2 + \frac{-1}{2}^2)$
$\frac{1}{2}||w||^2 = \frac{1}{2}(\frac{1}{4} + \frac{1}{4})$
$\frac{1}{2}||w||^2 = \frac{1}{2}(\frac{1}{2})$
$\frac{1}{2}||w||^2 = \frac{1}{4}$

(c) For large values of C, we want a small slack error, which allows us to focus on achieving a small training error but at the expense of an optimal margin. The support vectors chosen with a large C may be based on data points that are outliers, which leads to a hyperplane that separates the training data well, but has a higher chance to overfit. At C=infinity, this optimization problem is identical to the previous one and will give us the same hyperplane we found before, as an infinite C value will not allow any points to be misclassified, essentially giving us a hard-margin SVM. C=0 is the reverse of this, allowing us to misclassify any point with no penalty. C=1 allows us to meet in the middle, allowing for some miscalssification of points but still penalizing them, which should allow for us to find better support vectors and thus a good separating hyperplane.