CS446: Machine Learning **Spring 2017**

Problem Set 4

Kaizhao Liang(kl2) *Handed In: March 9, 2017*

1. a. **Algorithm:**
First compute distance from every data point to the center and sort them by the distance from small to large. Find the positive data point with smallest distance and set r_1 to its distance. Then find the data point with largest distance and set r_2 to its distance. If there is no positive point, then find two points which have largest difference in distances to the center and set r_1 to the smaller one added by a samll positive constant, r_2 to the larger one subtracted by a small positive constant.

Pesudo Code:

Initialize an empty priority queue Q
For each positive point in m:
    Q.push(pair(point,distance(point)))
If Q is empty:
    (p_1, p_2) = Minimize(abs(distance(p_1)-distance(p_2))) for p_1 and p_2 in m
    r_1=min(distance(p_1),distance(p_2))
    r_2=max(distance(p_1),distance(p_2))
temp=Q.pop()
r_1=temp.distance
While Q is not empty:
    temp=Q.pop()
r_2= temp.distance

b. i. Because the hypothesis is a tightest area encompassing all the positive points in the training set. While there could exist positive points from m that it has never seen lie outside of this area, all the negative points must lie outside of this area, therefore always classified correctly.

ii. Considering drawing samples from D each time is independent, the probability of that none of the point lie in the areas $r_1^* \leq |x|_2 \leq r_1$ *or* $r_2 < |x|_2 \leq r_2^*$ is $(1-\epsilon)^m$.

c. From b, we have the $Pr_{x \in D}(f(x) = y \; for \; m \; times) = (1-\epsilon)^m < \delta$
Given that $(1-x) < e^{-x}$,

$$e^{-\epsilon m} < \delta$$
$$m > \frac{1}{\epsilon} log(\frac{1}{\delta})$$

d. The VC dimension of the H_{2cc} is 2, since for 3 points, none of the function in the hypothesis space could shatter them, if we assign the labels to points in the order,

"+,-,+", from closest to furthest.
For infinite hypothesis space, the bound of the number of training samples is,

$$m > \frac{1}{\epsilon}(VC(H)log\frac{13}{\epsilon} + 4log\frac{2}{\delta})$$

$$VC(H) = 2$$

So,

$$m > \frac{2}{\epsilon}log\frac{13}{\epsilon} + \frac{4}{\epsilon}log\frac{2}{\delta}$$

The bound we derived in previous question is tighter than the this one, since both ϵ and δ are less or equal than 1, which means that $log(\frac{1}{\epsilon})$ and $log(\frac{1}{\delta})$ are positive, and the number derived in this question is larger than the previous, thus a tighter bound. The result makes sense, because the previous bound is derived for a specific algorithm and the the one in this question is more general.

2. The VC dimension of this hypothesis space is 3. The hypothesis can only divide the axis into three different intervals maximally no matter what the combination of a,b,c is. For the set of four points, there is no way it can shatter them if the points are labelled alternatively, like "+,-,+,-" or "-,+,-,+". However, any combination of labels of three can be shattered, because there could be two solutions, intercepts with the axis for paraboloid and the paraboloid could be convex or concave.

3. a. The dual representation:

$$w = \Sigma_{1,m} r\alpha_i y_i x^i$$

$$f(x) = sgn(w \cdot x) = sgn((\Sigma_{1,m} r\alpha_i y_i x^i) \cdot x) = sgn(\Sigma_{1,m} r\alpha_i y_i (x^i \cdot x))$$

where α_i is the number of mistakes made on the same sample.
Then, rewrite the hypothesis function:

$$f(x) = sgn(\Sigma_{z \in M} S(z) K(x, z))$$

where M is a set of samples that the algorithm makes mistakes on, and $K(x, z) = z \cdot x$ and $S(z = \pm r)$, depending whether the mistakes is a positive or negative example.

b. $K(x, z) = (x^T z)^3 + 49(x^T z + 4)^2 + 64x^T z$
$K(x, z) = (x^T z)^3 + 49(x^T z)^2 + 456x^T z + 784$
Since $x^T z$ is a valid kernel, the $Polynomial(x^T z)$ is also a valid kernel.

c. $K(x, z) = \Sigma_{c \in C} c(x)c(z) = C_m^k$
where m is the number of features that are positive in both x and z.
If m is less than k, then $K(x, z) = 0$.
Since the conjunction does not tolerate any negative features,only the k-conjunctions that have all positive features can survive in the multiplication. Finding features that are positive both in x and z takes O(n), which is linearly time.

4. a. 1. $w = [-1, 0]$
$\theta = -0.3$

2. $w = [-0.5, -0.5]$
   $\theta = 0$
3. The hard SVM formation tries to make the margin as large as possible, i.e, maximizing the distance between the separator and the closest point. For pair of any negative and positive points in the data set, the perpendicular bisector yields the largest margin for those particular two points. However, the true margin could be smaller because other points. So among those perpendicular bisectors, I find the one with the largest margin, which must be the solution for this optimization problem.

b. 1. Support Vectors:$[-2, 0], [0, 2]$
2. $\alpha = \{\frac{1}{4}, -\frac{1}{4}\}$
3. $Objective\ function\ value = \frac{1}{2}\|w\|^2 = \frac{1}{4}$

c. When C=0, the soft SVM becomes hard SVM and thus we get the same optimization result we got from (a)-2. The parameter C determine whether the optimization has small hinge loss or small margin. If C=0, the optimization is going to maximize the margin. If C=1, the optimization is going to evenly optimize the margin and hinge loss. If C=∞, the optimization is going to find the tightest hypothesis and thus minimize both the margin and hinge loss.

5. a. Boosting

i	Label	Hypothesis 1				Hypothesis 2			
		D_0	$f_1 \equiv [x > 2]$	$f_2 \equiv [y > 11]$	$h_1 \equiv [x > 2]$	D_1	$f_1 \equiv [x > 9]$	$f_2 \equiv [y > 11]$	$h_2 \equiv [y > 11]$
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	$-$	$\frac{1}{10}$	$-$	$-$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
2	$-$	$\frac{1}{10}$	$-$	$-$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
3	$+$	$\frac{1}{10}$	$+$	$-$	$+$	$\frac{1}{16}$	$-$	$-$	$-$
4	$-$	$\frac{1}{10}$	$-$	$-$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
5	$-$	$\frac{1}{10}$	$-$	$+$	$-$	$\frac{1}{16}$	$-$	$+$	$+$
6	$-$	$\frac{1}{10}$	$+$	$-$	$+$	$\frac{1}{4}$	$-$	$-$	$-$
7	$+$	$\frac{1}{10}$	$+$	$-$	$+$	$\frac{1}{16}$	$+$	$-$	$-$
8	$-$	$\frac{1}{10}$	$-$	$-$	$-$	$\frac{1}{16}$	$-$	$-$	$-$
9	$+$	$\frac{1}{10}$	$-$	$+$	$-$	$\frac{1}{4}$	$-$	$+$	$+$
10	$+$	$\frac{1}{10}$	$+$	$-$	$+$	$\frac{1}{16}$	$-$	$-$	$-$

Table 1: Table for Boosting results

b. The best hypothesis under the distribution D_0 is $x > 2$

c. The best hypothesis under the distribution D_1 is $y > 11$ Since the first weak learner makes 3 mistakes out of ten samples,

$$\epsilon_0 = \frac{2}{10}$$

$$\alpha_0 = \frac{1}{2} log(\frac{1-\epsilon_0}{\epsilon_0})$$

So,

$$\alpha_0 = \frac{1}{2} log(\frac{8}{2}) = log(2)$$

$$e^{+\alpha_0} = \frac{1}{\sqrt{\frac{8}{2}}} = \frac{1}{2}$$

$$e^{-\alpha_0} = \sqrt{\frac{8}{2}} = 2$$

Since in prediction, there are two misclassifications and eight correct classifications, eight of the values get demoted and two of them get promoted.

$$z_0 = \Sigma_i D_0(i) e^{-\alpha_0 y_i h_0(x_i)}$$

$$D_0(i) = \frac{1}{10} \; for \; i = 1, 2, 3, ...10$$

$$z_0 = \frac{1}{10}(8 \cdot \frac{1}{2} + 2 \cdot 2) = 0.8$$

So for the points that are correctly classified:

$$D_{1+} = \frac{D_0 e^{+\alpha_0}}{z_0} = \frac{\frac{1}{10} \cdot \frac{1}{2}}{0.8} = \frac{1}{16}$$

And for the points that are not correctly classified:

$$D_{1-} = \frac{D_0 e^{-\alpha_0}}{z_0} = \frac{\frac{1}{10} \cdot 2}{0.8} = \frac{1}{4}$$

The second weak learner makes 4 mistakes out of ten samples, but this time the distribution of the points is changed. So,

$$\epsilon_1 = \frac{4}{16} = \frac{1}{4}$$

$$\alpha_1 = \frac{1}{2} log \frac{1-\epsilon_1}{\epsilon_1} = \frac{1}{2} log(\frac{1-\frac{1}{4}}{\frac{1}{4}}) = \frac{1}{2} log(3)$$

d. $H_{final} = sgn\{log(2) \cdot sgn(x-2) + \frac{1}{2} log(3) \cdot sgn(y-11)\}$