Analysis
Homework 4: Learnability by Rohan Chandra

Answer 1:

The rademacher complexity is lower bounded by the growth function as:

$$\mathbb{R}_m(H) \leq \sqrt{\frac{2\log \Pi_G(m)}{m}}$$

and the growth fuction is lower bounded as:

$$\Pi_H(m) \leq \mathrm{O}\left(m^d\right)$$

where d is the VC dimension. The VC dimensions for the axis aligned rectangles, general hyperplanes, and original hyperplanes is ranked as $4 > 3 > 2$ respectively. Therefore, going by the inequalities shown above, the axis aligned rectangles would have the highest complexity (approx. 0.90) followed by general hyperplanes, and lastly by origin hyperplanes (approx. 0.70).

According to the relation:

$$\mathbb{E}[h(z)] \leq \frac{1}{m}\sum_{i=1}^{m} h(z_i) + 2\mathbb{R}_m(H) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

Rewritten as

$$\mathbb{E}[h(z)] - \frac{1}{m}\sum_{i=1}^{m} h(z_i) \leq 2\mathbb{R}_m(H) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

The L.H.S can be intuitively thought of as the generalization error or overfitting error. The R.H.S upper bounds the overfitting error. The better the correlation with sample noise (high rademacher complexity), the larger the upper bound will be leading to a higher chance for overfitting. Conversely, poorer the correlation with noise (lower rademacher complexity), tighter is the upper bound thus lessening the chance for over fitting.

This would lead us to believe that the origin hyperplane better outpeforms the general hyperplane, followed by the axis aligned rectangles in terms of generalization error. However, this cannot be assumed blindly. If the Rademacher complexity of the class of models is low, **and we nonetheless get a good fit to the training data**, that is actually evidence that we will continue to fit well on new data.

In conclusion, **ASSUMING a good fit (low training error) to the training data was achieved in all models**, a lower rademacher complexity would rank our models as follows: origin hyperplanes, general hyperplanes, and then axis aligned rectangles.

Answer 2:

We choose our data as $d_i = 2^{-x_i}$, for i $= 1 \ldots m$. Our classifier model is $\sin(\omega d_i)$, where ω is the frequency. We derive ω as follows:

$$\omega d_i = \left\{\begin{cases} [0,\pi] & : y_i = +1 \\ [\pi, 2\pi] & : y_i = -1 \end{cases}\right.$$

Replacing the value of d_i, and rewriting we get

$$\omega d_i = \left\{\begin{cases} \dfrac{\pi}{2^{x_i}} & : y_i = +1 \\ \pi + \dfrac{\pi}{2^{x_i}} & : y_i = -1 \end{cases}\right.$$

We are able to write this in this form because if we observe $x_i \in \mathbb{Z}_+$, then $\frac{\pi}{2^{x_i}} \in (0, \pi)$, and similarly, $\pi + \frac{\pi}{2^{x_i}} \in (\pi, 2\pi)$.

Summing over all y_i, and writing in closed form, we can derive ω as:

$$\omega = \pi\left(1 + \sum_{i=1}^{m} \frac{1-y_i}{2} 2^{x_i}\right)$$

Now if we consider positive labelled points, then the temr containing the sum goes to zero and

$$\omega d_i = \frac{\pi}{2^{x_i}}$$

which always lies between $(0, \frac{\pi}{2})$. This is basically from the derivation above. Now if you take any negatively labelled point, we get

$$\omega = \left(\frac{\pi}{2^{x_i}} + \frac{\pi}{2^{x_i}} \sum_{i=1}^{m} \frac{1-y_i}{2} 2^{x_i}\right)$$

making a similar argument for the next term, in the summation, the negative label being evaluated contributes a phase of π. Other negative labels with greater values of x_i, that is, $x_j > x_i$ for $j \neq i$ will only contribute phase shifts of 2π and can be ignored in the total phase calculation. Negative labels with values of x_i less than the x_i we're evaluating, that is, $x_j < x_i$ for $j \neq i$ will contribute a phase bounded by $(0, \frac{\pi}{2})$ as we showed in the previous part. So the summation term will be bounded by $(\pi, \pi + \frac{\pi}{2})$. Then we showed that the first term in that phase term is bounded by $(0, \frac{\pi}{2}))$ from our previous result so the total phase is always bounded by $(\pi, 2\pi)$ so that label will always return negative values!

Answer 3:

Now if we're classifying real numbers, some sets cannot be shattered. For example, if the set is given by:

$$S = [x, 2x, 3x, 4x]$$

For some real x, this set cannot be shattered for a set of labels:

Labels = True, True, False, True

From the bounds above, this would suggest that:

$$0 \leq \omega x \bmod 2\pi \leq \pi \quad (1)$$
$$0 \leq 2\omega x \bmod 2\pi \leq \pi \quad (2)$$
$$\pi \leq 3\omega x \bmod 2\pi \leq 2\pi \quad (3)$$
$$0 \leq 4\omega x \bmod 2\pi \leq \pi \quad (4)$$

Assuming $\omega x = \phi + 2n\pi$ we can see this reduces to:

$$0 \leq \phi \leq \pi \quad (5)$$
$$0 \leq 2\phi \leq \pi \implies 0 \leq \phi \leq \frac{\pi}{2} \quad (6)$$
$$\pi \leq 3\phi \leq 2\pi \quad (7)$$
$$0 \leq 4\phi \leq \pi \quad (8)$$

Looking at (8) more closely,

$$0 \leq \phi \leq \frac{\pi}{4} \quad (9)$$
$$\implies 0 \leq 3\phi \leq \frac{3\pi}{4} \quad (10)$$

If (5) alone was true, then (9) could be ambiguous, but because (6) is also assumed to be true it suggests this result is still consistent. However: $3\phi \geq \pi$ from (7) and $3\phi \leq \frac{3\pi}{4}$ from (10) which presents a contradiction. So given this set and a sin classifier, we cannot shatter the 4 points provided.

As per the definition of VC dimension, for VC dimension d there exist a set of d points that can be shattered and there is no set of $d + 1$ points that can be shattered. But, there might exist a set of d points that cannot be shattered and same goes for $m < d$. This means for the case of sine classifier it is true for the case of all $m < \infty$. So all we have to do is to find a case where 4 points cannot be shattered for a general omega.

Extra Credit

I utilize the concept of a separating hyperplane separating two convex hulls. Specifically, I cluster the given data points into two **disjoint** convex hulls C1 and C2, and find the closest pair of points from these hulls. The perpendicular bisector of the line joining these two points is the separating hyperplane.

When computing convex hulls of all possible combinations of hypotheses, it's possible that the hulls intersect. In such a case, it is not possible to draw a separating hyperplane. I thus add code to recognize when two hulls are colliding. This was done using a binary search search tree where one hull was fed into the tree to figure out the maximum and minimum coordinates, while an iterator was passed over the second hull to find out if there was a point that lay in the range of the maxima and minima.

Convex hulls can only be constructed for $n \geq 3$ points. Thus in cases where we had combinations of point and line, line and line, hull and point, hull and line, I used simple algera and geometry to find out the closest point from the line/hull to the next geometric object and the separating hyperplane became the perpendicular bisector of the line joiing the closest point to the object.