Solutions by **Andrew Lys** `andrewlys(at)u.e.`

1. Online Perceptron and Perceptron Analysis

1.1. Implementing CONSISTENT.

(a) Note, that if $\gamma(S)$, is the supremum over w, then we have:

$$\gamma(S) \geq \min_{(x_i,y_i)\in S} \frac{y_i\langle \hat{w}, \phi(x_i)\rangle}{\|\hat{w}\|}$$

And since $\hat{w}$ realizes S with probability one, we have:

$$y_i\langle \hat{w}, \phi(x_i)\rangle > 0$$

for all i, and since S is finite, we have:

$$\min_{(x_i,y_i)\in S} \frac{y_i\langle \hat{w}, \phi(x_i)\rangle}{\|\hat{w}\|} = m > 0$$

Therefore, $\gamma(S) \geq m > 0$

(b)

$$M_t \leq \frac{1}{\gamma(S)^2}$$

for all t, so

$$\limsup_{t\to\infty} M_t \leq \frac{1}{\gamma(S)^2}$$

Therefore, the possible number of mistakes is bounded by $\frac{1}{\gamma(S)^2}$. Further, the number of iterations is bounded by the number of mistakes, since there is an iteration only if a mistake was made.

(c) We implement the step as follows:

1: **for** $(\phi(x_i), y_i) \in S'$ **do**
2:     **if** $y_i\langle w, \phi(x_i)\rangle \leq 0$ **then**
3:         **return** (x_i, y_i)
4:     **end if**
5: **end for**

Before we invoke this iteration, we store

$S' \leftarrow \{(\phi(x_i), y_i)\}_{i=1}^m$

This operation takes $O(md)$ and takes up $O(m(d+1)) = O(md)$ memory. This simplifies the computation of $\phi(x_i)$ in our iteration.

On step 1, we compute

$$y_i \cdot (w_1\phi_1(x_i) + \ldots + w_d\phi_d(x_i))$$

Which consists of d multiplications inside the parantheses, $d-1$ additions inside the parentheses, and an additional multiplication by y_i. Thus, the total arithmetic is $O(2d) = O(d)$. Additionally, we compare to 0, which is a constant time operation.

We perform this step at most $|S| = m$ times, so the runtime of our iteration is $O(md)$. An iteration occurs only if a mistake happens as well, so the maximum number of times this iteration occurs is at most M_t, which is bounded by $\frac{1}{\gamma(S)^2}$. Thus, the total run time is bounded by

$$O\left(\frac{md}{\gamma(S)^2}\right)$$

(d) Let $\mathcal{X} = \mathbb{R}$ $\mathcal{Y} = \{\pm 1\}$, and $S = ((1,-1),(-1,-1))$. Clearly, this sample is not linearly separable, since for any sign of w, $\text{sign}(w \cdot 1) \neq \text{sign}(w \cdot (-1))$, unless w is zero, in which case the sign is still $+1$, which is still wrong. WLOG, we may assume that $w_0 = 0$.

If $w_t = 0$, then we have:

$$\text{sign}(w_t \cdot 1) = 1 \neq -1$$

Then

$$w_{t+1} = 0 + (-1)(1) = -1$$

If $w_t = -1$, then

$$\text{sign}(w_t \cdot (-1)) = 1 \neq -1$$

and then

$$w_{t+1} = -1 + (-1)(-1) = 0$$

Thus, the algorithm never stops.

1.2. **Statistical Guarantee.**

(a) Recall:

$$E_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(\tilde{\mathcal{A}})] \leq \frac{M}{m+1}$$

therefore, for the learning rule of $\widetilde{\text{PERCEPTRON}}$, we have:

$$E_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(\widetilde{\text{PERCEPTRON}}(S))] \leq \frac{1}{\gamma(S)^2(m+1)} < \varepsilon$$

Thus, the number of samples we need to ensure generalization error of at most ε is

$$\frac{1}{\gamma(S)^2\varepsilon} - 1 < m$$

(b) There is no contradiction because we are not actually learning the class of homogenous linear predictors on $\mathbb{R}^d$. We are learning the class of homogenous linear predictors with feature map ϕ. This has lower VCdim because the possible samples can only come from the image of the feature map, which need not be onto $\mathbb{R}^d$.

2. 0/1 Loss vs Squared Loss vs Hinge Loss

(a) Note that the sample S is finite, so $\Gamma_{\mathcal{H}}(S) < \infty$. Thus, $\exists h^*$, such that

$$\inf_{h\in\mathcal{H}} L_S^{01}(h) = L_S^{01}(h^*) = 0$$

Since,

$$|\{L_S^{01}(h) : h \in \mathcal{H}\}| \leq \Gamma_{\mathcal{H}}(S)$$

so we are minimizing over a finite set.

Additionally, since $L_S^{01}(h^*) = 0$, we have that $h^*(x_i) = y_i$ for all i.

Now we compute the square loss of h^*, which we have determined is indeed in our hypothesis class.

$$L_S^{sq}(h^*) = \frac{1}{m}\sum_{i=1}^{m} \ell^{sq}(h^*(x_i); y_i) = \frac{1}{m}\sum_{i=1}^{m}(y_i - h^*(x_i))^2 = 0$$

Therefore, we cannot have $\hat{h}_{sq}$ have error greater than 0.5, since h^* would have better error than it, and thus be better than the optimal.

(b) It is indeed possible for this to occur. Consider h^* as above, and simply let $h(x_i) = 2h^*(x_i)$. It is clear that h is still a linear predictor, and we have that h minimizes the hinge loss, since

$$\ell^{hinge}(h(x_i); y_i) = [1 - 2y_i h^*(x_i)]_+ = [1 - 2]_+ = 0$$

However, clearly, the 0/1 loss is 1, since every prediction is either $+2$ or -2, which is always wrong.