Matemáticas del aprendizaje de máquina - 2022 II
Ejercicios Statistical Learning Theory

Manuela Leal Peláez
mlealp@unal.edu.co

June 20, 2022

The following corresponds to exercises from Abu-Mostafa, Y. S. (2012). *Learning from Data.*

1. **Exercise 1.2: Suppose that we use a perceptron to detect spam messages. Let's say that each email message is represented by the frequency of occurrence of keywords, and the output is $+1$ if the message is considered as spam.**

   (a) **Can you think of some keywords that will end up with a large positive weight in the perceptron?**

   Miss, giftcard, win, offer, exclusive, promo, chosen, discount, opportunity, inactive...

   (b) **How about keywords that will get a negative weight?**

   Dear, Miss, Mr, greetings, the, I, be, is, ...

   (c) **What parameter in the perceptron directly affects how many borderline messages end up being classified as spam?**

   In the perceptron the parameter is the threshold used to determine positive and negative classes (spam/non.spam), which affects the amount of borderline classified objects.

2. **Exercise 1.3: The weight update rule in (1.3), $w(t+1) = w(t) + y(t)x(t)$, has the nice interpretation that it moves in the direction of classifying $x(t)$ correctly.**

   (a) **Show that $y(t)w^T(t)x(t) < 0$.**

   Notice that if $x(t)$ is misclassified bu $w(t)$ then $y(t)$ differs from the sign of $w^T(t)x(t)$, which means that one of them is $+1$ and the other -1 necessarily. Therefore $y(t)w^T(t)x(t) = (1)(-1) = -1$ or $y(t)w^T(t)x(t) = (-1)(1) = -1$; either case $y(t)w^T(t)x(t) = -1 < 0$.

(b) **Show that $y(t)w^T(t+1)x(t) > y(t)w^T(t)x(t)$.**

The result follows from the following

$$\begin{aligned}
y(t)w^T(t+1)x(t) &= y(t)(w(t)+y(t)x(t))^T x(t)\\
&= y(t)(w(t)^T + y(t)x(t)^T)x(t)\\
&= (y(t)w(t)^T + y(t)^2 x(t)^T)x(t)\\
&= y(t)x(t)w(t)^T + y(t)^2 x(t)^T x(t)\\
&= y(t)w(t)^T x(t) + y(t)^2 ||x(t)||_2^2\\
&> y(t)w(t)^T x(t)
\end{aligned}$$

(c) **As far as classifying $x(t)$ is concerned, argue that the move from $w(t)$ to $w(t+1)$ is a move "in the right direction".**

We just saw that $y(t)w^T(t+1)x(t) > y(t)w^T(t)x(t)$, in other words $y(t)w^T(t)x(t)$ and t have a direct proportionality relationship, as the increase of the later implies increase of the first. Also, from this and the fact that $y(t)$ differs from the sign of $w^T(t)x(t)$, if $y(t) = -1$ then increasing t implies decreasing $w^T(t)x(t)$ towards negative region, and if $y(t) = +1$ then increasing t implies increasing $w^T(t)x(t)$ towards positive region. In either case, referring to the classification of $x(t)$, the movements implied by increasing t (i.e the move from $w(t)$ to $w(t+1)$) follows the right direction.

3. **Exercise 1.11: We are given a data set D of 25 training examples from an unknown target function $f : X \to Y$, where $X = \mathbb{R}$ and $\mathcal{Y} = \{1, +1\}$. To learn f, we use a simple hypothesis set $\mathcal{H} = \{h_1, h_2\}$ where h_1 is the constant $+1$ function and h_2 is the constant 1. We consider two learning algorithms, S (smart) and C (crazy). S chooses the hypothesis that agrees the most with D and C chooses the other hypothesis deliberately. Let us see how these algorithms perform out of sample from the deterministic and probabilistic view that there is a probability distribution on $\mathcal{X}$, and let $\mathbb{P}[f(x) = +1] = p$.**

(a) **Can S produce a hypothesis that is guaranteed to perform better than random on any point outside D?**

No, this can't be guaranteed. Consider that for a random function we would have a probability of labeling the points as $+1$ or -1 with a $50\% - 50\%$ probability, which implies having at least one match with h_1 outside $\mathcal{D}$ for example. However, if we had an hypothesis that classifies as positives every sample within $\mathcal{D}$ and negative in other case, then the learning algorithm S will produce the hypothesis h_1 but this will not match $\S - \mathcal{D}$ at any point. Therefore the random function would do better.

(b) **Assume for the rest of the exercise that all the examples in $\mathcal{D}$ have $yn = +1$. Is it possible that the hypothesis that C produces turns out to be better**

than the hypothesis that S produces?

Indeed, as exemplified previously, $50\% - 50\%$ probability can classify better than the hypothesis given by S when every example in $\mathcal{D}$ is positive class.

(c) **If $p = 0.9$, what is the probability that S will produce a better hypothesis than C?**

As the hypothesis that agrees the most is h_1 we will have that S produces h_1 and C produces h_2. If $p = 0.9$ we will have that h_1 matches f outside $\mathcal{D}$ with a 90% chance while h_2 does it with a 10% chance, therefore h_1 produces by S being better that C.

(d) **Is there any value of p for which it is more likely than not that C will produce a better hypothesis than S?**

Similarly, ss the hypothesis that agrees the most is h_1 we will have that S produces h_1 and C produces h_2. If $p < 0.5$ we will have that h_1 matches f outside $\mathcal{D}$ with a chance lower than 50% while h_2 does it with greater chance, therefore h_2 produces by C being better that S.

4. **Exercise 1.12: A friend comes to you with a learning problem. She says the target function f is completely unknown, but she has 4000 data points. She is willing to pay to you to solve her problem and produce for her a g which approximates f. What is the best that you can promise her among the following:**

   (a) **After learning you will provide her with a g that you will guarantee approximates f well out of sample.**

   (b) **After learning you will provide her with a g, and with high probability the g which you produce will approximate f well out of sample.**

   (c) **One of two things will happen.**

      i. **You will produce a hypothesis g;**

      ii. **You will declare that you failed.**

   If you do return a hypothesis g, then with high probability the g which you produce will approximate f well out of sample.

As f is completely unknown then there is nothing I could assure regarding $E_{out(g)}$. However, recalling Hoeffding's inequality, under which $E_{in}(g) \approx E_{out}(g)$, if I can ascertain something about $E_{in}(g)$ then I could as well say something about $E_{out}(g)$. In this case, having a large enough amount of data, chances are that g will give a good approximation of f based on examples, and therefore $E_{in}(g) \approx 0$, but if it is not the case that the function can be learned under these conditions, then this cannot be assured. Therefore, either $E_{in}(g) \approx E_{out}(g) \approx 0$, or failure must be declared. The best that can then be promised is c).