Chapter - 2
Theory of Learning

Aviral Janveja

1 Can We Learn?

Revisiting the machine learning process described in chapter 1 : We have an unknown target function f, which represents the underlying pattern that we would like to uncover. Next, we have a set of observations that will be used to approximate the unknown target function. Finally, our approximation of the target function, called the hypothesis function g, based on the sample of data that we have.

Now, let us say our hypothesis function performs well on the available data points. However, remember that the goal in machine learning is not for g to perform well in-sample but for g to approximate f well, that is $g \approx f$. So, the question remains, does our hypothesis function perform well out of sample?

2 Answer : Probably, Approximately.

First, let us formalize the performance of our hypothesis function g in terms of its agreement or disagreement with f on the available data points.

For each point x in our data sample. If $g(\mathbf{x}) \neq f(\mathbf{x})$, that is, if our hypothesis disagrees with the given correct output, then it constitutes an **error**. Now, the sum-total of errors that our hypothesis function g makes in-sample, would then be called the in-sample error or E_{in}. Naturally, we try to minimize E_{in} as much as possible. But does a small in-sample error E_{in} imply a small out-of-sample error E_{out} as well, which is what actually matters for learning?

2.1 Possible Vs Probable

In order to answer the above question, first let us consider a simple example in probability. Imagine a bin filled with 100 marbles, 90 green and 10 red. Suppose you randomly pick 10 marbles out of the bin. Is it possible that all of them or most of them turn out red? Well yes, it is possible, but is it probable? Not really. Computing the exact values, shows that the probability of getting 5 or more reds from the above bin is :

$$P[Red \geq 5] \approx 0.000672$$

Therefore, getting a mostly red sample from a mostly green bin is highly unlikely.

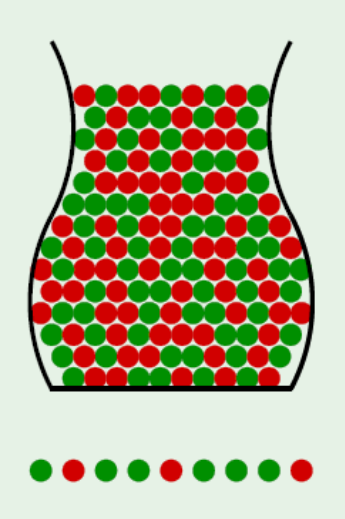

Figure 1: Bin Example

We can similarly compute the probabilities of getting 1 red marble, 2 red marbles and so on in our sample of 10. We can also compute the above results for a different sample size of 20 marbles instead. Computing these results from the bin example above, clearly show us that the bin frequency and the sample frequency are likely to track each other, not perfectly but closely, especially as the sample size increases.

2.2 Connection to Learning

Connecting the above example to our learning scenario. The bin represents the general population and the sample represents our data points that come from the very same population. The green marble is when our hypothesis function gets it right and the red marble represents an error. Therefore, answering the question raised above : we can safely claim that, as per the laws of probability, the in-sample error E_{in} is likely to track the out-of-sample error E_{out}, especially when the sample-size is sufficiently large.

Formally, E_{in} is related to E_{out} through the **Hoeffding Inequality**. It is one of the laws of large numbers from the field of statistics. We use it here in a slightly modified form, adapted as per our use-case in machine learning :

$$P[\ |E_{in}(g) - E_{out}(g)| > \epsilon\ ] \leq 2Me^{-2\epsilon^2 N}$$

Let us clarify the different terms mentioned in the above formula. $E_{in}(g)$ is the in-sample error, $E_{out}(g)$ is the out-of-sample error, ϵ is the tolerance value, that is the difference between E_{in} and E_{out} that is acceptable to us. N is the sample size, that is the number of data points available to us. Hence, the above equation can be put into simple words as follows :

> The probability of $E_{in}(g)$ and $E_{out}(g)$ diverging beyond our tolerance value ϵ will be low, if we have a reasonable ϵ and a lot of data points N.

It can be seen in the above inequality that, having a lot of data points reduces the value on the right-hand-side, thus giving us a stronger probability bound. Whereas, a very small tolerance value tends to increase the right-hand-side value, hence weakening our probability bound.

Finally, M is the number of candidate hypothesis functions out of which g was chosen. Think of it in terms of arriving at the final perceptron weight vector $\mathbf{w}$ after iterating through several possible candidates. Now, this might look surprising as there could be infinitely many choices of real-valued weights in a perceptron. Indeed, the number of possible hypothesis is actually infinite for most relevant learning models. A large M, let alone an infinite one, will make our probability bound completely meaningless. However, this is not our final result in the theory of learning. We will deal with this M term going forward.

3 Error Measures

4 References

1. CalTech Machine Learning Course - CS156, Lecture 2.