5. Learning Problems

We now aim to combine the concepts which we have thus far encountered, and construct a formal definition for a learning problem. We will begin by defining the necessary inputs, and the desired outputs, before working towards the fundamental definition of PAC learnability, which will allow us to have some notion of the possibility of learning a certain problem.

5.1 Introduction

We begin the chapter by laying out an example which we will reference and build around throughout.

Suppose that you are a loaner, aiming to determine if an arbitrary loanee will or will not default on the loan you give them. We phrase the problem deliberately in this manner, to emphasize the simplistic scenario in which we are working – the value of the loan is predetermined, and the goal is to provide a binary classification of loanees.

What information will be useful in this task? How will we determine if our solution is adequate? How will be provide a useful notation to describe the problem?

5.2 A formal learning model

5.2.1 Learning input

The learner of a statistical learning problem has access to the following,

- Domain set: an arbitrary set which we typically label by $\mathcal{X}$. This set is nothing more than the objects to which we aim to assign labels. In the example outlined in the previous section, this set $\mathcal{X}$ is the set of potential loanees.
- Label set: A set which we typically label by $\mathcal{Y}$. This is the set of possible labels, e.g., { default , no default }. Of course, it is more common to denote the possible labels simply by integers, and in our binary classification example, we will take $\mathcal{Y} = \{\pm 1\}$.
- Training data: a finite sequence of pairs $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, i.e., a sequence of labelled objects. In our case, this will be a sequence of past loaners together with information relating to their repayment. We will tend to denote the training data by $\mathcal{S} = ((x_1, y_1), ..., (x_m, y_m))$, where m is the size of the training data.

> Remark. It is important to note that the training data is described as a sequence rather than a set. This is a rather nuanced point, and not one which we will dedicate much thought to. The reason for such pedanticity is that there exist learning algorithms which are dependent on the order of the training points, and there is no necessity for $\mathcal{S}$ to be duplicate free.
>
> It is however still common to refer to $\mathcal{S}$ as the training set.

5.2.2 Learning output

- Prediction rule: the only output from a statistical learning problem is a function $h : \mathcal{X} \to \mathcal{Y}$. We also refer to the function as a hypothesis, and is a rule which the learner can use to label new elements of the domain space.

If we are considering an algorithm $\mathcal{A}$, then we will denote by $\mathcal{A}(\mathcal{S})$ or $h_{\mathcal{S}}$ the output hypothesis of the algorithm.

5.2.3 A data generating model

5.2.4 Measure of success