Chapter 2

Machine Learning

Given that machine learning is the basis for the development and research in this thesis, it is important to understand its characteristics and differences from other approaches in computer science. Nilsson [18] defines learning as gaining knowledge, understanding of or skill in, by study, instruction or experience. We may say that a machine learns whenever it modifies data structures, variable values or programs in order to improve its performance against a certain problem.

2.1. Machine Learning Overview

We define machine learning as an area of artificial intelligence which deals with the design of programs and systems which taking advantage of data improve the accuracy or performance of a task or a set of tasks. [16] The main difference between machine learning and other areas of computer science is the approach taken to solve the problem. As indicated by Mitchell, the general approach in computer science is based on assuming that there exists a well defined problem and the challenge resides in finding an algorithmic solution which efficiently makes use of resources and an implementation which best fits the problem. Machine learning takes a different approach; instead of

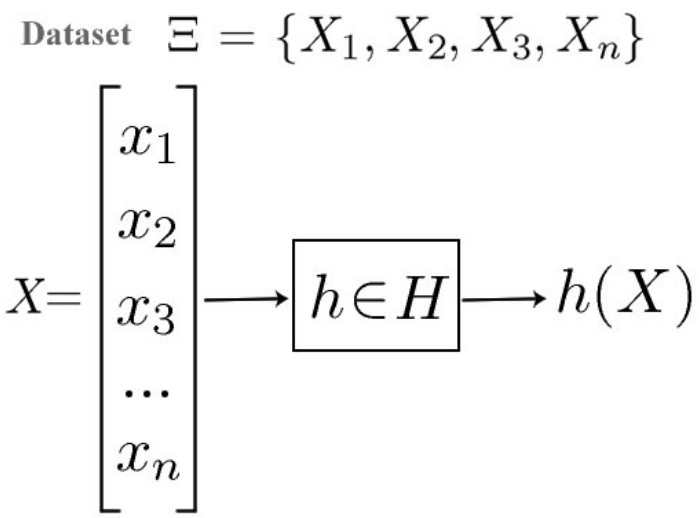

Figure 2.1: Machine Learning Function H

finding the exact solution of the problem, the focus is to design an algorithm (or learning machine) which with the combination of a set of examples of the problem with its corresponding solutions (dataset) is able to solve the problem (figure 2.1). As Mitchell stands out, there exist three levels of generalization in a machine learning solution:

- A set of problems which the machine is able to solve.
- A specific problem which the machine can solve.
- An specific instance of a problem which the machine correctly solves.

One may wonder why taking the machine learning indirect approach instead of the path exposed by algorithm design. Nilsson mentions several reasons [18] :

- Several problems may not be defined but by example input and output. This is the case of problems whose complexity or unknown nature makes it difficult to be modeled. A machine is then expected to be able to learn how to obtain the correct output from the available data and to represent the function that describes the relationship between input and output in its internal data.

- Problems may change over time and machines are able to adapt to changes.
- Several problem domains contain too much data for a human to be able to extract and devise an algorithm from it. A machine that is able to gain this knowledge gradually may be able to extract the important information more efficiently than a human.
- A machine is able to obtain an approximate solution to a problem even when not enough data is available to model the problem or an exact solution cannot be obtained.

Machine learning is specially useful in the problems that are the focus of this research work, which are classification and prediction. In most problems of this kind, multiple variables are present, making it complex to create a single model able to always make the correct decision. These problems also often contain too much data, factor that presents a performance barrier for many algorithmic approaches.

2.2. Concepts in Machine Learning

We assume there is an unknown function f which we wish to learn. In our learning process, we make a hypothesis h of function f. Both functions f and h receive an input $X = \{x_1, x_2, x_3, ..., x_n\}$ of n components and have an associated output $f(X)$ and $h(X)$. The hypothesis h belongs to a family of functions H to which f may also belong. Through training, based on the input data set $\Xi = \{X_1, X_2, X_3, ..., X_n\}$ the function h is chosen such that it represents the best hypothesis of f. [18]

A data set Ξ may be presented in two different ways, each of which present a different learning problem. If $\forall m \in \Xi$, $f(m)$ is a known (or approximated) value, then we have the case of **supervised learning** in which we select h which best adjusts

to f. In the case where there is no value associated with an input X, the learning process is focused on splitting the dataset Ξ into smaller subgroups $\Xi_1, \Xi_2, ..., \Xi_n$ such that each subgroups represents certain information. This kind of learning is known as **unsupervised learning**.

In the literature there are different ways in which an input X, $X \in \Xi$ may be named. Input vector, sample and instance are used interchangeably. Each component of an input vector $x_i \in X$ is known as a feature or attribute. The nature of the output value can be a floating point value or a category. In classification problems, the output is also called label, while in prediction problems simply prediction.

2.3. Factors in Learning

In machine learning, a dataset Ξ is usually partitioned into two different sets: A training set Ξ_{train} and a test set Ξ_{test}. The purpose of this division is to be able to evaluate the algorithm's performance on unseen data. Hence, Ξ_{train} is used on training and Ξ_{test} is used on evaluation. The error of h on Ξ_{train} is known as **training error**, while the error on Ξ_{test} is known as **generalization error**. The generalization error is relevant as it is a measure of the performance of the algorithm in solving the problem at hand.

Another important factor in machine learning is the bias. The bias refers to the a priori information used when designing the learning algorithm or preparing the data that affects the final result of learning. An important bias is the selection of the family of functions H in which we assume that f belongs. As Nilsson expresses it [18], bias is always present in machine learning problems.

Another important factor is noise. Noise can affect both features and measured outputs and it can bring the machine to match a function f which is incorrect. The

problem of noise is related to the problem of overfitting. Overfitting refers to defining a function h which is too adapted to Ξ_{train}, but fails to perform well on Ξ_{test}. This is a common case when Ξ_{train} is not general enough or the amount of noise is substantial.

2.4. Machine Learning Theory

Given a machine learning algorithm, it is important to be able to define when an algorithm is successful. Furthermore, we need to be able to define an upper bound for the generalization error ϵ given almost any training set. Of course, sets which are corrupt or do not represent a good sample of the problem space will not be able to hold this upper limit. Valiant [27] introduced a theoretic approach for defining this upper limit using PAC (Probably Approximately Correct) learning.

2.4.1. PAC Learning

In order to provide an insight into PAC Learning [9] (Probably Approximately Correct Learning), we will look specifically at the classification problem. Given the set H of functions, we assume the true classification function $h^* \in H$. h is a boolean classifier with possible results 1 (classified as) or -1 (not classified as). We also assume that $\forall X \in \Xi$, X is generated independently according to a probability function $P(X)$ over $\Re^d \times \{-1, 1\}$. We then define the generalization error as:

$$err_P(h) = P\{x : h(x) \neq h^*(x)\} \tag{2.1}$$

Being P over the input space now. Based on the previous definition, Hoffman [13] defines a learning algorithm A as a mapping from training data to a hypothesis $h \in H$:

$$A = \bigcup_{n=1}^{\infty} (X \times \{-1, 1\})^n \rightarrow H, \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \tag{2.2}$$

Given that different datasets produce different generalization errors (hence, the generalization error is a random variable), we may determine an average value for it on a sample of size n:

$$err(A, n) \equiv E_{P_n}[err_P(A(X^n))] \tag{2.3}$$

where E_{P_n} is the expected value for the distribution P of size n. PAC Learning bases its analysis on the previous expression by defining an upper bound on the generalization error of $h(X^n)$:

$$P^n\{X^n : err_p(h(X^n)) > \epsilon\} < \delta \tag{2.4}$$

Hoffman [13] provides the following definition of PAC Learning:

> A hypothesis space H is PAC-learnable if there exists a learning algorithm A that makes use of an oracle to generate random samples in unit time, such that: for every deterministic concept with true model $h^* \in H$ (the so-called learnable case), for every input distribution P and every choice of $\delta, \epsilon > 0$: with probability at least $1 - \delta$ the algorithm returns a hypothesis $h \in H$ with $err_P(h) \leq \epsilon$, in time polynomial in $1/\delta$, $1/\epsilon$, and m, where m is the dimensionality of the feature space (number of attributes).

From this definition, the concept of sample complexity can be developed. For a hypothesis space H, there exists a probability of at least $1 - \delta$ that for a function $h \in H$ consistent with a n-sized training set, $err_p(h) < \epsilon$.

2.4.2. VC Theory

It is possible to define the upperbound for the generalization error. Two different approaches involve considering a finite or infinite set of hypothesis [13]. In the case of

a finite set of hypothesis, given a dataset Ξ of size n, the probability of h to not make any mistake on the training set is $(1-\epsilon)^n$, as the probability of not making a mistake on a single sample is $(1-\epsilon)$. Hoffman [13] indicates that a binomial tail bound may be used as follows:

$$err_p(h) > \epsilon \Rightarrow P^n\{h \in V(X^n)\} = (1-\epsilon)^n \leq e^{-\epsilon n} \tag{2.5}$$

where $V(X^n)$ is the version space, and $P^n\{h \in V(X^n)\}$ is the probability that h will be in the version space given a random sample X^n. The version space is defined as the set of all hypothesis consistent with the training data. If we now extend the definition from one single hypothesis to all possible hypothesis, we obtain:

$$P\{\exists h \in V(X^n)\ and\ err_p(h) > \epsilon\} \leq \|H\| e^{-\epsilon n} \tag{2.6}$$

The previous expression indicates an upper bound for the probability of having one consistent hypothesis whose error is greater that ϵ. Now using the PAC form:

$$P^n\{X^n : err_P(A(X^n)) \leq \frac{1}{n}\log\frac{\|H\|}{\delta}\} \geq 1-\delta \tag{2.7}$$

We may interpret this expression as follows: Given a machine learning algorithm A, the probability of its generalization error being less than $\frac{1}{n}\log\frac{\|H\|}{\delta}$ is $1-\delta$. A possible observation from the previous deduction is the inverse relationship between δ and the lower bound $1-\delta$. Proofs for the previous expressions may be found in [27] and [13].