***The Learning problem***

In this chapter we will give a clear definition on what is considered as a learning problem. I will start by giving an example about machine learning, then go explore some components of machine learning. I will introduce a simple algorithm in a conceptual way to define the various mechanics and components of a machine learning model, then move on to define the types of learning.

# The essence of machine learning

Learning from data is used in situations where we don't have an analytic solution, but we do have data that we can use to construct an empirical solution. This premise covers a lot of territory, and indeed learning from data is one of the most widely used techniques in science, engineering, and economics, among other fields. In consequence there are 3 indispensable things that needs to exist within the problem for it to be a learning problem.

1. Existence of a pattern.
2. Cannot pin it down mathematically i.e. cannot write a formula that explains the underling system.
3. We need GOOD DATA.

# Example: if a client would buy a product or not

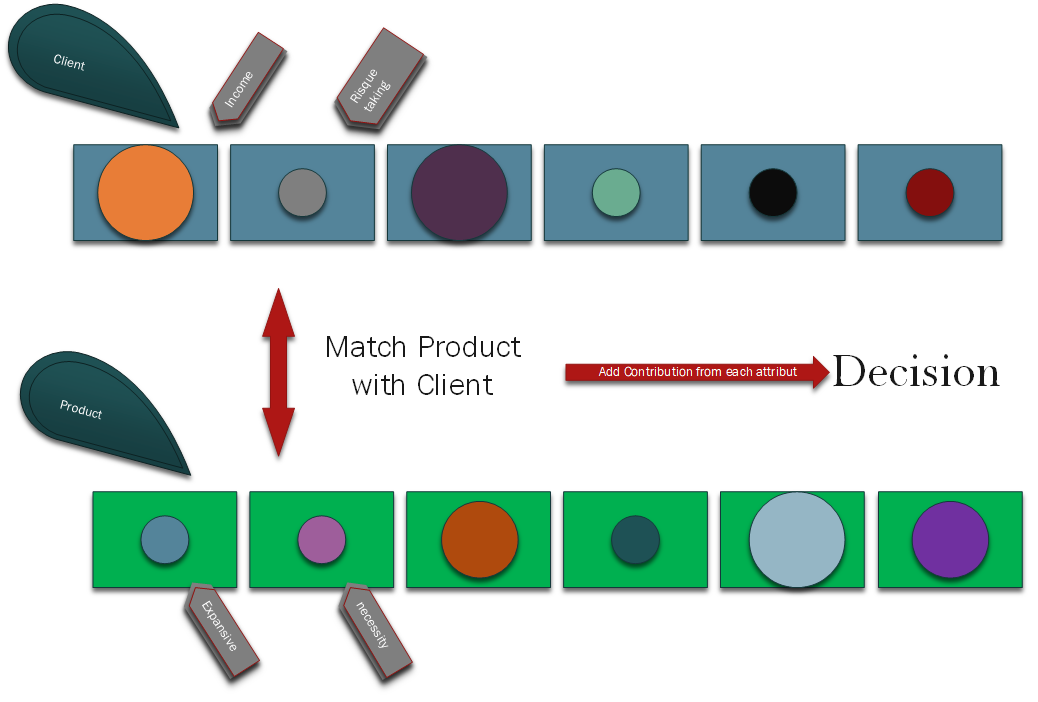
The main difficulty in this problem is that the criteria that client uses to buy products are quite complex. Trying to model those explicitly is no easy task, so it may not be possible to come up with an analytic solution. But what we know that the historical transaction data reveal a lot about how people perform their shopping, so we may be able to construct a good empirical solution. There is a great deal of data available to marketing companies, since they often, well, buy it from data-trackers and data providers, like amazon or Facebook.

Figure 1 A model how to predict if a client would buy a pro or not

The first figure represent a valid and logical approach to the problem at hand except that it’s not a machine learning approach. In machine learning you do not need to match the fields (attributes), or add their contribution to the decision manually or by any algorithm, the machine learning problem is a **Learning From Data Problem** i.e. from the data we reverse engineer and capture the essence that distinguish every decision (look at figure 2 below).

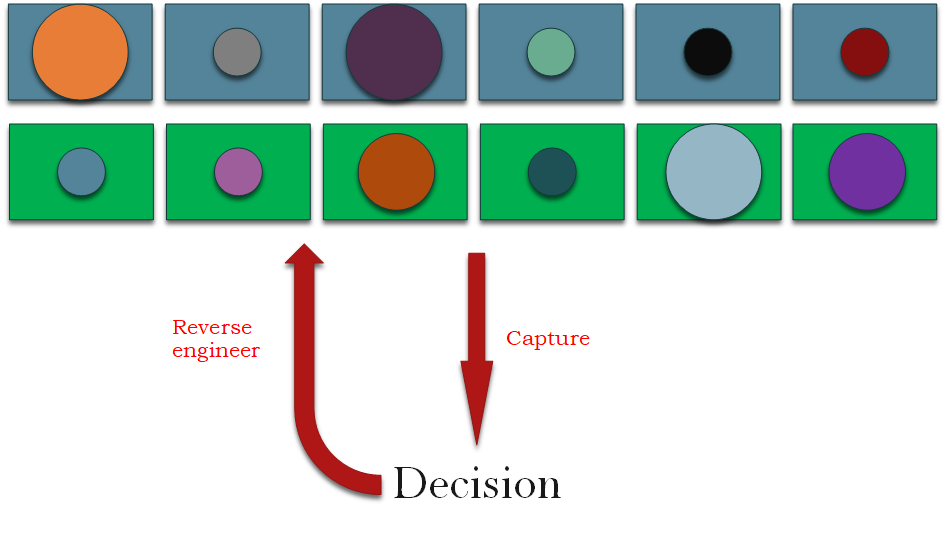


Figure 2 The Machine Learning Model

The power of learning from data is that this entire process (the process of figuring out the decision) can be automated, without any need for analyzing either the client profile or the product attributes.

## The Mathematical component of the learning process

Let us give names and symbols to the main components of this learning problem.

1. Input : x in X (example: Clients information)
2. Output: y in Y(example: Decisions buy a certain Product or not)
3. Target function f:X🡪Y (this function is unknown and represents the taste of the client if it can be pin point mathematically, this is the function that we trying to find)
4. Data, comes in pairs {(x1 ,y1)…(xN,yN)}
5. The Hypothesis g:X🡪Y (g approximate f)
6. The learning algorithm A that tries to find g in the hypothesis set H (example H could be the set of all linear formulas from which the algorithm would choose the best linear fit (g) to the data)

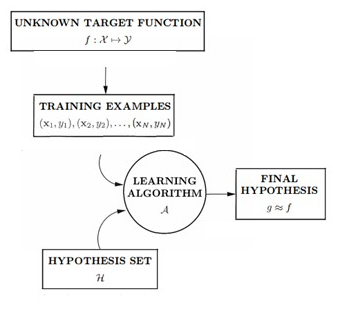


Figure 3 Basic setup of the learning problem

|  |
| --- |
| The 3rd figure represent the various links between these concepts. In general the learning process can be summed as follows:  There is a target to be learned. It is unknown to us. We have a set of examples generated by the target. The learning algorithm uses these examples to look for a hypothesis that approximates the target. |

# Simple Algorithm: the Perceptron:

In this part I will introduce one of the simplest yet useless algorithm in machine learning literature, just to familiarize with the various technical definitions, and terminology. We will go for more details in the next chapters.

But first let us consider the different components of figure 3. Given a specific learning problem, the target function and training examples are dictated by the problem. However, the learning algorithm and hypothesis set are not. These are solution tools that we get to choose. The hypothesis set and learning algorithm are referred to informally as **the learning model**. Here two example of a learning model:

1. H: Neural Network; Learning Algorithm: Backpropagation.

2. H: Support Vector Machine; Learning Algorithm: Quadratic Programing.

## The Perceptron

We will start by defining the different components of figure 3 using the example of the client/product already introduced

1. Input x the client attribute or X =Rd
2. Output y in Y={-1,1} as y=-1 if he didn’t buy the product y=1 if he did buy the product
3. H is the space of all linear function from X🡪Y so **g is a linear separator** (and hence its usefulness). We note H= {h} or h linear combination of the input x and some free parameters that we will define next.

The functional form h(x) that we choose here, gives different weights to the different coordinates of x, reflecting their relative importance in the client decision. So in mathematical form we have

Here we have two parameters to search for the free weights **wi** and the threshold.

|  |
| --- |
| ***Attention :***  *h is not g, h is a family of linear combination of free weights, threshold and the input x or g is just one element of this family, it’s by definition the element that gives the most performance in training and testing.* |

We can rewrite the expression above in a more compact format:

* the *sign* function return -1 if it’s input is negative +1 otherwise.
* (x1,…xd) components of x and w0= threshold

By introducing x0=1 we can write the previous expression in a more compact way

* w=(w0,..wd)
* x=(1,x1,…xd)=(x0,x1,…,xd)

|  |
| --- |
| ***Attention:***  *We did not define a training algorithm yet. But in all cases the h function behaves similarly, it always tries to find a linear line that splits the data into two categories “buy, didn’t buy”. If we look conceptually at the presented model, we will see that in order to make good prediction, the “Learning Algorithm” needs to assign large weights to the attributes of x that contributes the most to the decision, and null or smaller weights to the insignificant attributes, and use the bias term the split between the two decision “buy, didn’t buy”.* |

Now we introduce the “perceptron learning algorithm “ **PLA.** The algorithm will determine what w should be, based on the data. Let us assume that the data set is linearly separable, which means that there is a vector w that makes h(x) achieve the correct decision, or for all n data point we have h(xn)=yn. Now we have a system of linear equations and we need to define the optimum solution in term of misclassification.

Our learning algorithm will find this w using a simple iterative method.

|  |
| --- |
| 1. ***As long as there is a misclassification:***    1. ***Pick a misclassified pair form the data as (x(t),y(t))***    2. ***Update the weights by:***       1. ***W(t+1)=w(t)+y(t)x(t)***    3. ***Test the new weight on all the data*** |

Although the update rule in (1.a) considers only one training example at a time and may “mess up” the classification of the other examples that are not involved in the current iteration, it turns out that the algorithm is guaranteed to arrive at the right solution in the end.

|  |
| --- |
| ***Remarque:***  *Within the infinite space of all weight vectors, the perceptron algorithm manages to find a weight vector that works, using a simple iterative process. This illustrates how a learning algorithm can effectively search an infinite hypothesis set using a finite number of simple steps. This feature is a characteristic of many techniques that are used in learning, some of which are far more sophisticated than the perceptron learning algorithm.* |

The goal of this part is to give an example and to create a close up look on the different abstraction of machine learning models, and also to show the different mechanics of these concepts, and the interaction between them. Now we move to the most import part of this chapter and that is Design vs learning.

# Design Vs Learning

The goal is to distinguish between learning and a related approach that is used for similar problems. While learning is based on data, this other approach does not use data. It is a “design” approach based on specifications (example: the dynamics of systems like Partial Differential Equations, Dynamic Programing and Optimal Control, Modeling by Mathematical Programing, inferential Statistics, descriptive Statistics, …) , and is often discussed alongside the learning approach in pattern recognition literature.

Let’s consider the problem of classification different type of rocks, based only on their chemical component:

## The Learning Approach:

We are given samples from each classes of rocks and we use them as our data set. We take the percentages of chemicals as our input vector x and the class number as output y. there are some variation in the values of the chemical composites, but each samples of a given class tends to cluster around a focal point. Then the learning algorithm define the boundaries of each class.

## The Design Approach:

We call a specialist to give us information about the geothermal dynamics on how chemical component form various crystals, and how different combination of crystals cluster to form rocks. We also ask him about the number of different classes of rocks are in any given geographical place to figure out the relative frequency of each class of rocks. Finally, we make a physical model of the variations of the chemical component due to the geothermal forces and weather conditions based on every geolocation. We put all of this information together and compute the full joint probability distribution of the chemical components. Once we have that joint distribution, we can construct the optimal decision rule to classify the rocks.

The main difference between the learning approach and the design approach is the role that data plays. In the design approach, the problem is well specified and one can analytically derive f without the need to see any data. In the learning approach, the problem is much less specified, and one needs data to pin down what f is. Both approaches may be viable in some applications, but only the learning approach is possible in many applications where the target function is unknown.

# Types of Learning:

The basic premise of learning from data is the use of a set of observations to uncover an underlying process. It is a very broad premise, and difficult to fit into a single framework. As a result, different learning paradigms have arisen to deal with different situations and different assumptions. In this section, we introduce some of these paradigms.

In this courses we only focus on one paradigms “Supervised Learning”, but it’s important to take a look at some of the other paradigms. Notably some of the other paradigms are the Stats of the Art in AI, and now revolutionizing a lot of industries.

## Supervised Learning:

When the training data contains explicit examples of what the correct output should be for a given inputs, then we are within the supervised learning setting. The most notable example is the classification of images using captions. Regression is another notorious example of supervised machine learning, we will go on more details later on the lectures.

## Unsupervised Learning:

The training data does not contain any output information at all. We are just given input example, and the learning algorithm will try to find the correct pattern that flows through the data. The most notable example is K-Nearest-Neighbor often used in classification problems.

## Reinforced Learning:

This method of learning is most similar on how humans learn, here we are given the input and some output but not all, and also we grade the output, and the learning algorithm tries to figure a pattern by trial and error, in case of a classification problem a reinforced learning algorithm will get good grade if he classified the input correctly and bad grades if he didn’t this dynamic will push it to better itself after each data treatment. And it’s similar on how humans learn, if we are correct on a subject our brains will reward us with dopamine, and if we are not correct we feel a little bit of unease and discomfort.

# Conclusion:

In this chapter the most important thing to take is the difference between a problem that could be solved by a machine learning approach and a problem that could not. Solving a problem using ML requires that the problem satisfied some criteria (we defined those criteria in “The essence of machine learning”). Also it is vigorously important to know the difference between a design approach and a learning one, even in some cases they both use data to detect patterns, but their essence is very different ( I cannot stress you enough on how BIG these points are ).