Introduction

1 What is Machine Learning?

Two definitions of Machine Learning are offered. Arthur Samuel described it as: ”the field of study that gives computers the ability to learn without being explicitly programmed.” This is an older, informal definition.

Tom Mitchell provides a more modern definition: ”A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Example: playing checkers.

- $E =$ the experience of playing many games of checkers
- $T =$ the task of playing checkers.
- $P =$ the probability that the program will win the next game.

2 Supervised Learning

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

Supervised learning problems are categorized into ”**regression**” and ”**classification**” problems. In a regression problem, we are trying to predict results within a **continuous** output, meaning that we are trying to map input variables to some **continuous** function. In a **classification** problem, we are instead trying to predict results in a **discrete** output. In other words, we are trying to map input variables into **discrete** categories.

Example:

Given data about the size of houses on the real estate market, try to predict their price. Price as a function of size is a *continuous* output, so this is a regression problem.

We could turn this example into a classification problem by instead making our output about whether the house ”sells for more or less than the asking price.” Here we are classifying the houses based on price into two *discrete* categories.

3 Unsupervised Learning

Unsupervised learning, on the other hand, allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don’t necessarily know the effect of the variables.

We can derive this structure by clustering the data based on relationships among the variables in the data.

With unsupervised learning there is no feedback based on the prediction results, i.e., there is no teacher to correct you. It is not just about clustering. For example, associative memory is unsupervised learning.

Example:

Clustering: Take a collection of 1000 essays written on the US Economy, and find a way to automatically group these essays into a small number that are somehow similar or related by different variables, such as word frequency, sentence length, page count, and so on.

Associative: Suppose a doctor over years of experience forms associations in his mind between patient characteristics and illnesses that they have. If a new patient shows up then based on this patients characteristics such as symptoms, family medical history, physical attributes, mental outlook, etc the doctor associates possible illness or illnesses based on what the doctor has seen before with similar patients. This is not the same as rule based reasoning as in expert systems. In this case we would like to estimate a mapping function from patient characteristics into illnesses.

4 Training Set

For Y^1, a dependent variable of the independent variables $x_1^1, x_2^1, ..., x_n^1$, which shape the vector X^1, we define the Training Set T as the group of elements $T = (X^1, Y^1, X^2, Y^2, ..., X^m, Y^m)$, where $m, n \in \mathbb{N}$. m is the number of **Training examples**, and n is the number of independent variables associated with each dependent variable, or the number of **Features**. hereinafter we will call the X^j like **Inputs**, and the Y^j like **Outputs**.

Example: Suppose the price of a house depends on several features like it is shown in the following table:

Size($feet^2$)	Number of bedrooms	Number of floors	Age($years$)	Price($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

For this example we have $m = 4$(Features) and $n = 4$(Training examples). And
$X^1 = \begin{bmatrix} 2104, & 5, & 1, & 45 \end{bmatrix}, Y^1 = \begin{bmatrix} 460 \end{bmatrix}$, and $X_3^2 = 2$

5 Lineal Regression

For this regression we suppose that it is possible to get an output value $h_\theta(x)$, fitting the input values in a lineal function that is call **Hypothesis**:

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_2 + \theta_2 x_2 + ... + \theta_n x_n \tag{1}$$

Where $\theta_0, \theta_1, \theta_2, ..., \theta_n$ are the **Parameters** of the *hypothesis.*

5.1 Cost function

If we will find some function to fit the input parameters and obtain some output, it is useful to compare the results with the training Y values, to do that we can use the cost function:

$$J(\theta_0, \theta_1, \theta_2, ..., \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{2}$$

The matrix version of this formula is:

$$J(\theta) = \frac{1}{2m} (X\theta - Y)^T (X\theta - Y) \tag{3}$$

The purpose is obtain a function so that the value of the cost function is the lowest possible, to find these value there are different techniques, one of them is the gradient descent.

5.2 Gradient Descent

The idea is to start at some initial value for the hypothesis function, find the direction in which the function decreases faster(partial derivative), and take a small step in that direction(multiply by a small constant), and then repeat the process until find the local minimum value of the function. One step is:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{4}$$

5.3 Feature Normalization

When features differ by orders of magnitude, first performing feature scaling can make gradient descent converge much more quickly. The most general formula to normalize the feature values is:

$$x_j^i =: \frac{x_j^i - \mu}{s} \tag{5}$$

Where μ is the average value, and μ is the standard deviation of $X^{(i)}$.

5.4 Normal Equation

There is a closed-from solution of to linear regresion:

$$\theta = (X^T X)^{-1} X^T Y \tag{6}$$