24/07/24

For the exam remember to know what the effect of the hyperparameters etc.

Introduction

Machine learning in the literature can only be used in tabular form, but not everything can be represented in this way. This because sometimes you have relation between the elements. For example in text you can see this fenomenon. What is able to operate on this problems is **deep learning**.

Generative AI is able to create new data from previous knowledge

- - - -

Understanding is about **causality**

Example:

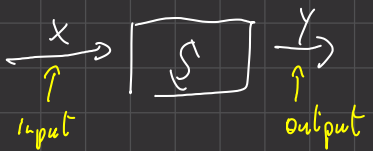
It's hot then people go to the beach and eat ice cream.
Then I hide this fenomen and give this info to a machine it will tell me that there is a correlation between hotness and ice cream consumption.

The data is very important because a machine learns correlation from it.

Machine learning

What we want to do needs to be agnostic. So we must be able to learn independently from the source.

We have a system:



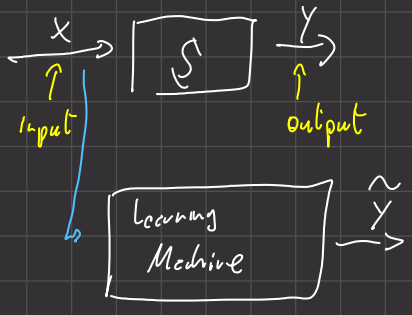
I don't know what's inside S.

Input in machine learning can be easily measured.

The output is not easy to measure or I don't know it now. (For example a plant will breed in one month)

The objective of ML is to predict the output.

Future can be predicted when it's similar to the past.



26/09/24

We want to learn from a completely unknown system S.

Inputs (x) are easy to measure, outputs (y) no.

Usually there is no causality between inputs and outputs. It may happen that y generates x

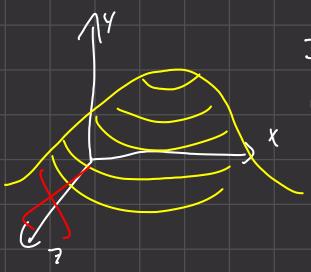
We want to create $\hat{y}$ as close as possible to y.

Even with an oracle there is no possibility knowing x to know y. This because of noise, the measurement will be wrong even with the best tool in the world.

The only way to have a very exact measure is taking more than one measurement. This noise not only takes into account the measurement noise, but also the numerical error due to the fact that we need to save data on digital hardware.

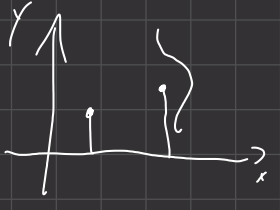
The last "source of noise" could be that we missed a variable and so we are not able to compute correctly the output

Example:



If I omit a dimension then I will lose relevant information. For the same x it seems that there could be two different results.

We cannot distinguish between all these sources of noise:



Two sources of noise:

- Measure
- Partial Knowledge of x

> Input x: it can be $\mathbb{R}$ (number), $\mathbb{R}^d$ (vector), $\{G, Y, B\}$ (category)
> Output y) (graph)

We like real numbers because we have many different ways to operate on them. Other types of functions they could be more challenging.

Note that by doing: $\{Y, G, B\} \to \{1, 2, 3\}$ you are expressing that yellow is closer to green than Blue

the way of transforming them is to transform them in a vector like this:

Y -> [1 | 0 | 0]
G -> [0 | 1 | 0]
B -> [0 | 0 | 1]

this approach is called One Hot coding

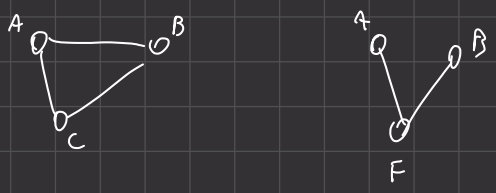
Here the notion of distance is not in play. they are all distant in the same way.

> What if we have ordinal variables?

$x \in \{1, 2, 3, 4\}$ $\quad \hookrightarrow \mathbb{R}$

We can leave it like they are, no need for one hot coding because there is order.

> Graphs:



For them I can use an adjacency matrix

	A	B	C
A	0	1	1
B	1	0	1
C	1	1	0

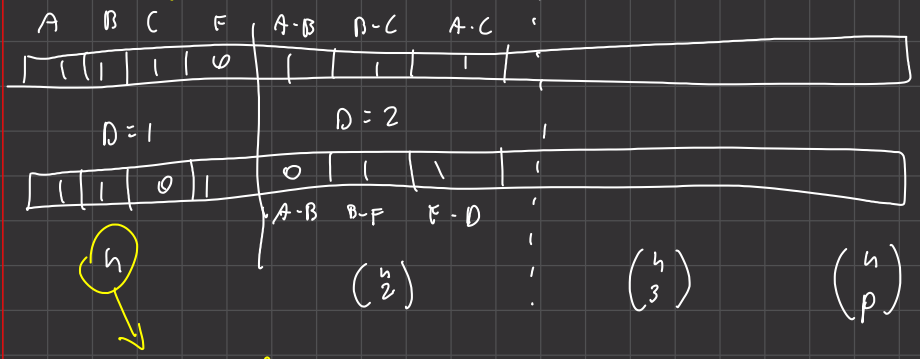
	A	B	F
A	0	0	1
B	0	0	1
F	1	1	0

The problem here is that there is no way of defining distance between matrices easily.

Distance is THE parameter needed for machine learning.

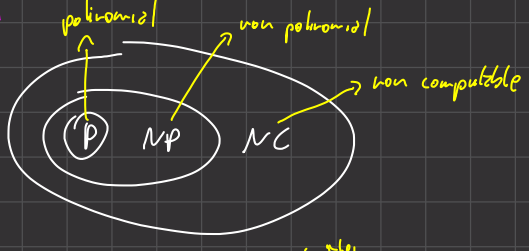
One trick could be to define vectors.

Structures of dimension 1



number of nodes

Reminder:



Note O → This is better.

$O(\ln(n))$ $O(n^7)$

$10^{70} \ln(n)$ $10^{-30} n^7$ ← This is better.

Machine learning is almost polynomial problems. Deep learning is non polynomial } approximation of non computable problem of learning from data.

A sequence is a unidirected graph.

Whatever we have in input we can map in a series of real numbers.

- - - - - - .

y can be the same as what x can be.

It can be transformed in a vector of elements.

L.M => $y = f(x)$ $y_i = f_i(x)$ Element by element.

Learning machine

$y \in \mathbb{R}^p$ $x \in \mathbb{R}^d$

The most general problem is:

$$y = f(x)$$

$x \in \mathbb{R}^d$, $y \in \mathbb{R}$

The good approximation is called loss function that usually returns a real number.

Taylor and Fourier are an approximation for example, but in Machine learning everything is different.

First of all we should define $f(x)$

There is a theorem called no free lunch that tells us which the pecurses of ML if we choose the optimal for one problem will be the worse for another.

$$f(x) = \sum_{i=0}^{p} c_i x^i$$ with enough variables I can fit any function.

loss function — distance — I want to find these.

$$l = (f(x), y) = (y - f(x))^2$$

$D_n = \{(x_1, y_1) \dots (x_n, y_n)\}$ approximated with min square error.

Let's consider p (grade of the polinomial)

Finding it for a computer is hard but finding c is easy

The actual way of computing goodness of c_i is:

$$\min_c \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{p} c_j x_i^j\right) = \|Xc - y\|^2$$

$$c = \begin{bmatrix} c_0 \\ \vdots \\ c_p \end{bmatrix} \in \mathbb{R}^{p+1} \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n \qquad X = \begin{bmatrix} x_1^0 & \dots & x_1^p \\ \vdots & & \\ x_n^0 & \dots & x_n^p \end{bmatrix} \in \mathbb{R}^{n \times (p+1)}$$

$$\left[\begin{array}{|c|} \hline \\ \\ \\ \\ \hline \end{array} \begin{array}{|c|} \hline \\ \\ \\ \\ \hline \end{array} - \begin{array}{|c|} \hline \\ \\ \\ \\ \hline \end{array} \right]^2 = \left\| \quad \right\|^2$$

X c y

$\mathbb{R}^{n \times (p+1)}$ $\mathbb{R}^{(p+1)}$ $\mathbb{R}^n$

$\mathbb{R}^{n \times 1}$

$\min \|Xc - y\|^2$ This I want to do

↳ paraboloid

To find the minimum I will need to find the gradient of the function.

$\min \|Xc - y\|^2 =$

$\nabla_c (c^T X^T X c - 2 c^T X^T y + y^T y) = \emptyset$

$2 X^T X c - 2 X^T y = \emptyset$

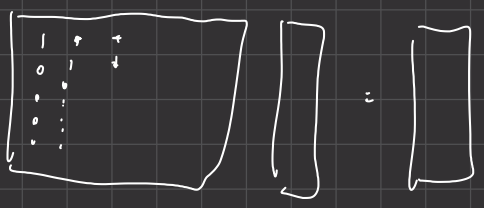
$X^T X c = X^T y \qquad Ax = b$

$c = (X^T X)^{\#} X^T y \longrightarrow$ pseudoinverse.

$O(n^2 \cdot n^3)$ → computational cost.

Remark:

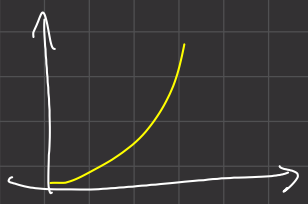
$Ax = b$



To do Gauss Jordan $O(n^2)$

$x = A^{\#} b : O(n^{2.3})$

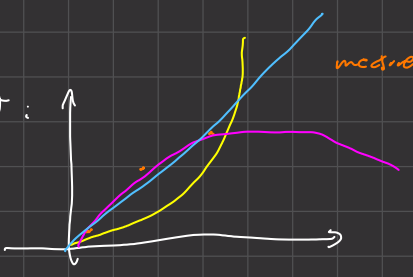
Suppose now that our data has been generated by something like $y = x^2$



On here I fit my function. $\hat{y} = f(x) = \sum_{i=0}^{p} c_i x^i$

Even though it may seem that the best $p = 2$ it can be either $\{0, 1, 2\}$.

It may happen due to noise that:

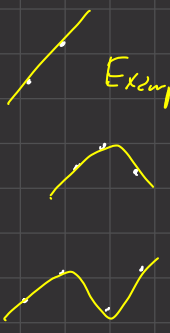


the line fits better the original data than the predicted.

Suppose that $p = n-1$

$$\min_{c} \| X c - y \|^2 = 0 \quad \forall D_n$$

Examples.



this means that minimizing over data is not smart otherwise we would minimize over noise.

remark on statistics.

$x \to x_1, \dots, x_n$ $\qquad \mu = \mathbb{E}_x\{x\} = \int_D x\, p(x)\, dx$

independent and identical distributed.

μ can be estimated with empirical average $= \frac{1}{n} \sum_{i=1}^{n} x_i$

$$\mathbb{E}_{\tilde{x}}\{\tilde{x}\} = \mathbb{E}_{x_1, x_n}\left\{ \frac{1}{n} \sum_{i=1}^{n} x_i \right\} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{x_i}\{x_i\} = \mu$$

This is not enough to get the average, we need.

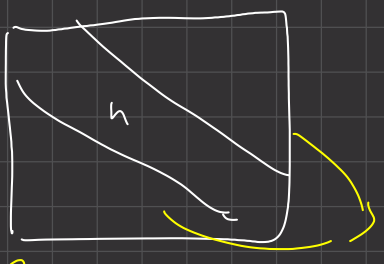
- Estimator
- Accuracy
- Confidence.

We need the variance:

$\sigma^2 = \mathbb{E}_x\{(x-\mu)^2\} =$ the variance is important because we can prove the law of large number so the larger the bolic the more my estimator $\bar{x}$ will be close to μ

$= \mathbb{E}_x\{x^2\} - 2\mathbb{E}_x\{x\mu\} + \mathbb{E}_x[\mu^2] = \mathbb{E}_x\{x^2\} - \mu^2$

$\sigma_{\bar{x}}^2 = \mathbb{E}_{\bar{x}}\{\bar{x}^2\} - \left(\mathbb{E}_{\bar{x}}\{\bar{x}\}\right)^2 = \mathbb{E}_{x_1 \dots x_n}\left\{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=i}^{m} x_i x_j\right\} - \mu^2 =$



$i = j$ → x_i^2 ; $i \neq j$ → $x_i x_j$

$= \frac{1}{n}\sum_{x=1}^{n}\sum_{\substack{j=1 \\ i \neq j}}^{n} \mathbb{E}_{x_i x_j}(x_i x_j) + \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}_i(x_i^2) - \mu^2$

TODO ???

$\mathbb{E}_{x_i x_j}(x_i x_j) \to \mu^2$; $\mathbb{E}_i(x_i^2) \to \mathbb{E}_x(x^2)$

$= \frac{n^2-n}{n^2}\mu^2 + \frac{n}{n^2}\mathbb{E}_x\{x^2\} - \mu^2 = \frac{n^2-n-n^2}{n^2}\mu^2 + \frac{1}{n}\mathbb{E}_x\{x^2\} =$

$= \frac{1}{n}\left(\mathbb{E}_x\{x^2\} - \mu^2\right) = \frac{1}{n}\sigma_x^2$ Proof of Large number theorem.

The more data like more I will follow the data that I want to predict.

> Confidence

$\sigma_{\bar{x}}^2 = \int_{-\infty}^{+\infty}(\bar{x}-\mu)^2 p(\bar{x})\,d\bar{x} \geq \int_{|\bar{x}-\mu|>\varepsilon}(\bar{x}-\mu)^2 p(\bar{x})\,d\bar{x} \geq$

this is because positive function.

$\geq \varepsilon^2 \int_{|\bar{x}-\mu|\geq\varepsilon} p(\bar{x})\,d\bar{x} = \varepsilon^2 p\{|\bar{x}-\mu| \geq \varepsilon\}$

I have proved that:

$$P\{|\bar{x}-\mu| \geq \varepsilon\} \leq \frac{\sigma_{\bar{x}}^2}{\varepsilon^2} = \frac{\sigma_x^2}{n\varepsilon^2} \leq \frac{1}{n\varepsilon^2} = \delta \qquad \varepsilon = \sqrt{\frac{1}{n\delta}}$$

Variance is less than the size of the domain

$\frac{1}{n\varepsilon^2}$ confidence

TODO what's what?

$$P\{|\bar{x}-\mu| \geq \varepsilon\} \leq \frac{1}{n\varepsilon^2} = \delta \qquad \varepsilon = \sqrt{\frac{1}{n\delta}}$$

$$|x-\mu| \underset{(1-\delta)}{\leq} \sqrt{\frac{1}{n\delta}}$$

$$\mu \underset{(1-\delta)}{\leq} \bar{x} + \sqrt{\frac{1}{n\delta}}$$

$x \in \{0,1\}$

$x_1 \dots x_n \Rightarrow$ i.i.d.

If I want 0.95 accuracy δ is 0.01

$x \in \mathbb{R}$

$y \in \mathbb{R}$ $\qquad D_n = \{(x_1,y_1) \dots (x_n,y_n)\}$

$$f(x) = \sum_{i=0}^{p} c_i x^i$$

$$l = (f(x),y) = (y - f(x))^2$$

$$\min_c \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \longleftrightarrow \mathbb{E}_{x,y} (y - f(x))^2$$

$\bar{x}$ $\qquad$ μ

This is the learning part.

$l \in [0,1]$

independent and identically distributed

allows the large number theorem

is time independent

20:32 ??? exist if I remove the learning part.

Once I choose the minimum I lose the independence. (Difference between statistics and ML)

IID must be there in order to do ML.