

24/07/24

For the exam remember to know about 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 phenomenon. 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:

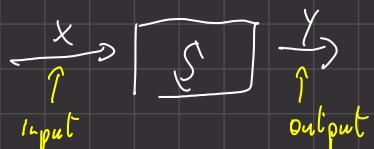
It's hot then people go to the beach and eat ice cream. Then I hide this question and give this info to a machine. It will tell me that there is a correlation between hotness and ice cream consumption.

The data is very important because machine learning conclusion 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:



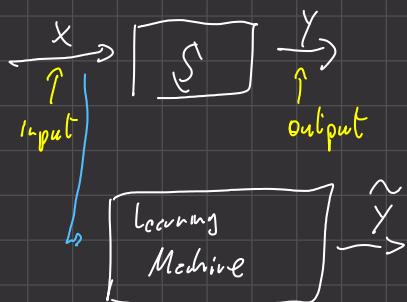
I don't know what's inside S.

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 break in one month)

The objective of ML is to predict the output.

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



26/03/24

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 .

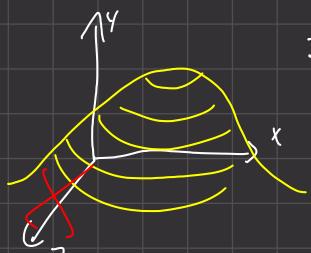
We want to create \hat{y} as close as possible to y .

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.

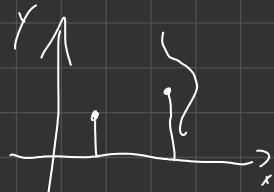
The last "source of noise" could be that we missed a variable and so we are not able to compute correctly the output.

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.

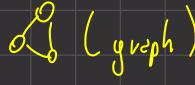
We cannot distinguish between all these sources of noise:



Two sources of noise:

- Measure

- Partial Knowledge of x

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

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

Note that by doing: $\{Y, G, \beta\} \rightarrow \{1, 2, 3\}$ 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:

$$\begin{aligned} Y &\rightarrow [1 | 0 | 0] \\ G &\rightarrow [0 | 1 | 0] \\ \beta &\rightarrow [0 | 0 | 1] \end{aligned} \quad \left. \begin{array}{l} \text{this approach is called} \\ \text{One Hot coding} \end{array} \right\}$$

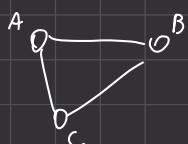
Now the notion of distance is not in play. They are all distinct in the same way.

> 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

$$\begin{matrix} & A & B & C \\ A & 0 & 1 & 1 \\ B & 1 & 0 & 1 \\ C & 1 & 1 & 0 \end{matrix}$$

$$\begin{matrix} & A & B & F \\ A & 0 & 0 & 1 \\ B & 0 & 0 & 1 \\ F & 1 & 1 & 0 \end{matrix}$$

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

Disclose is THE parameter needed for machine learning.

One trick could be to define vectors.

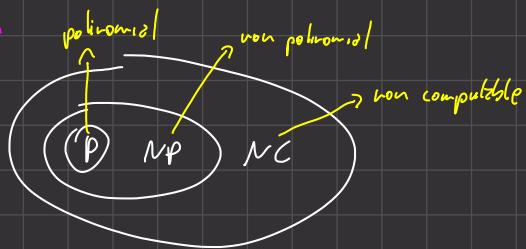
Structures of dimension 1

A	B	C	F	A+B	B+C	A.C	.
1	1	1	1	1	1	1	
D=1				D=2			
1	1	0	1	0	1	1	
				A-B	B-F	F-D	
				($\frac{n}{2}$)		($\frac{n}{3}$)	($\frac{n}{p}$)

(n)

number of nodes

problems:



$$\text{note } O \rightarrow O(\ln(n)) \rightarrow \text{This is better.} \\ O(n^3) \quad \quad \quad$$

$$2^{30} \ln(n) \quad 2^{30} n^3 \\ \text{This is better.}$$

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

A sequence is a unidirectional 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.

Let $M \Rightarrow$ $y = f(x)$ $y_i = f_i(x)$ Element by element.
Learning machine $f: \mathbb{R}^p \rightarrow \mathbb{R}^d$

The most general problem is:

$$y = f(x)$$

The good approximation is called less function that usually occurs e real number.

$$x \in \mathbb{N}$$

Taylor und Fourier zählen zu Approximationen, für example, und in
Mehrdeutigkeitsschichtung ist die Schichtung unterschiedlich.

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

there is a theorem called no free lunch that tells us which the courses or MC if we choose the optimal for one problem will be the worse for another.

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

Let's consider p (grade of the polynomial)

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

The easiest way of computing goodness of fit is:

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

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

$$\left[\begin{bmatrix} 1 & x \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ y \end{bmatrix} \right]^2 = \begin{bmatrix} \|R^{n \times (p+1)}\| \\ \|R^{(p+1)}\| \\ \|R^n\| \end{bmatrix}^2$$

$R^{n \times 1}$

$$\min_{\Sigma} \|x_c - y\|^2 \quad \text{This I want to do}$$

↳ paraboloid

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

$$\min_{\Sigma} \|x_c - y\|^2 =$$

$$\nabla_c \left(c^T x^T x - 2c^T x^T y + y^T y \right) = \phi$$

$$\not c^T x^T x - 2c^T x^T y = \phi$$

$$c^T x^T x = c^T y \quad Ax = b$$

$c = (x^T x)^* x^T y \xrightarrow{\text{pseudoinverse}}$

$\mathcal{O}(n^2 - n^3) \rightarrow \text{computational cost.}$

Example:

$$Ax = b$$

$$\left[\begin{array}{ccc} 1 & * & * \\ 0 & 1 & * \\ \vdots & \vdots & \vdots \\ 0 & \vdots & \vdots \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

To do Gauss Jordan $\mathcal{O}(n^2)$

$$x = A^{-1} b = \mathcal{O}(n^2)$$

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

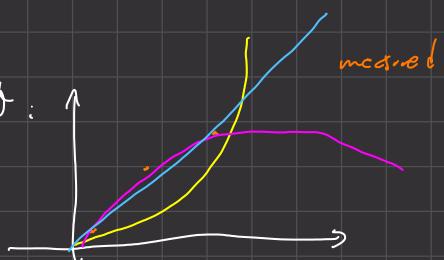


On have fit my function. $\hat{y} = f(x) = \sum_{i=0}^p c_i x^i$

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 parabola.



Suppose that $p=n-1$

$$\min_s \|x - y\|^2 = 0 \quad \forall D_n$$

Examples.

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

Revert on statistics.

$$x \rightarrow x_1, \dots, x_n$$

$$\mu = E_x\{x\} = \int_p x p(x) dx$$

independent and identical distributed.

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

$$E_{\bar{x}}\{\bar{x}\} = E_{x_1, x_n} \left\{ \frac{1}{n} \sum_{i=1}^n x_i \right\} = \frac{1}{n} \sum_{i=1}^n E_{x_i}\{x_i\} = \mu$$

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

- Estimation

- Accuracy

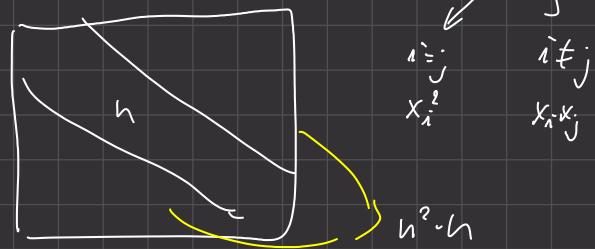
- Confidence.

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 variance the worse my estimator \bar{x} will be close to μ

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

$$\sigma_{\bar{x}}^2 = |\mathbb{E}_{\bar{x}} \{ \bar{x}^2 \} - (\mathbb{E}_{\bar{x}} \{ \bar{x} \})^2| = |\mathbb{E}_{\bar{x} \sim \text{uniform}} \left\{ \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n x_i x_j \right\} - \mu^2|$$



$$= \frac{1}{n} \sum_{x=1}^n \sum_{j=1}^n |\mathbb{E}_{x \sim \bar{x}} (x_i x_j) - \mu^2| + \frac{1}{n^2} \sum_{i=1}^n |\mathbb{E}_x (x_i^2) - \mu^2|$$

↑
 \bar{x}_{ij}
↓
 μ^2

\downarrow
 $\mathbb{E}_x(x)$

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

$$= \frac{1}{n} \left(|\mathbb{E}_x \{ x^2 \} - \mu^2| \right) = \frac{1}{n} \partial_x^2 \quad \text{Proof of Law of Large Number.}$$

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

→ Confidence

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

This is because positive function.

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

I have proved that:

Variance less than the size of the above

$$P \{ |\bar{x} - \mu| \geq \varepsilon \} \leq \frac{\partial_x^2}{\varepsilon^2} = \frac{\partial_x^2 x}{n \varepsilon^2} \leq \underbrace{\frac{1}{n \varepsilon^2}}_{\text{confidence}} = \delta \quad \varepsilon = \sqrt{\frac{\sigma}{n \delta}}$$

YODO what's about?

confidence

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

$$|\bar{x} - \mu| \leq \sqrt{\frac{1}{n\sigma}} \quad (1-\sigma)$$

$$\bar{x} \leq \bar{x} + \sqrt{\frac{1}{n\sigma}} \quad x \in \{0, 1\} \\ (1-\sigma) \quad x_1, \dots, x_n \Rightarrow i.i.d.$$

If I want 99% accuracy σ is 0.01

$x \in \mathbb{R}$

$y \in \mathbb{R}$

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

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

$$\ell = (f(x), y) = (y - f(x))^2$$

$$\min_c \sum_{i=1}^n (y_i - f(x_i))^2 \quad \hookrightarrow \quad E_{x,y} (y - f(x))^2$$

$\underbrace{\qquad}_{x_i} \qquad \qquad \qquad \underbrace{\qquad}_{M}$

This is the learning part. $\ell \in [0, \infty]$

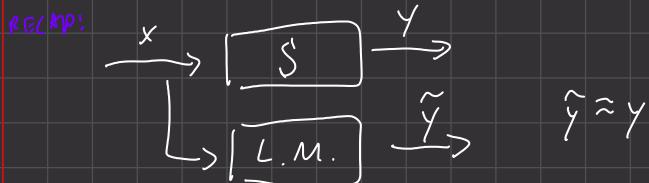
independent and identically distributed
shows the large number theorem (\rightarrow same independent)

[70'32] ?? what if we remove the learning part.

Once we choose the minimum we lose the independence. (Difference between statistics and MLE)

FFD must be there in order to do ML.

7/10/24



$$\text{Dataset} = \left\{ (x_1, y_1) \dots (x_n, y_n) \right\} \quad x, y \in \mathbb{R}$$

Only one input and output.

$\hat{y} = f(x) = \sum_{i=1}^n c_i x^i \rightarrow$ for a polynomial of high enough degree we can build any function.
 ↳ I need a function to map y from x .

There is no optimal choice because of the no-free-lunch.

Loss function

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

We want the coefficient that minimizes the my error on my data:

$$\min_c \underbrace{\frac{1}{n} \sum_{i=1}^n \left(y_i - \sum_{j=0}^p c_j x_i^j \right)}_{\text{emp. avg}} \rightarrow \min_c \| X c - y \|_2^2$$

$$\text{Just one l. system } (X^T X) c = X^T y \quad O(p^2)$$

$$\text{Solve all l. systems. } c = (X^T X)^{-1} X^T y \quad O(p^3)$$

Remember the problem with overfitting

Having a distribution: (statistics)

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

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

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

$$\mu = \mathbb{E}_x \{x\}$$

mean value

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

The more measures that I do the better the \bar{x} will be.

upper bound, so I want the worst case

What I really want to minimize is:

$$\mathbb{E}_{(x,y)} \underbrace{l(f(x), y)}_{X}$$

To have iid losses I need iid data. Data iid means that the system is not changing in time.

By doing the minimizing the errors won't be independent anymore and so we cannot use statistics.

We now need to bring back statistical.

We have selected a function $f \in F$ based on a dataset: $D_n = \{(x_i, y_i) \dots (x_n, y_n)\}$

Since they did not see each other (p.37) they are independent. i.i.d.

Note, the independence is between the log of functions and the dataset.

We suppose finite functions even though we have real coefficients so they could be infinite.

The functions are selected independently from the data. N.B.

I can put the independency hyp on the log because one of them will fit the data probably.

$$P \left\{ \left| \underbrace{\mathbb{E}_{(x,y)} l(f(x), y)}_{\text{expected value}} - \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i) \right| \geq \varepsilon \right\} \leq \frac{1}{n \varepsilon^2}$$

Empirical error

The errors are iid since I apply something independent to iid data.

The expected error of each function is also to the empirical error on the data because fns are independent from the data.

I don't know the prior for the selection on all the functions, but I have nice prior for every single function. So I will need to consider all the functions, so I consider the case where everything will go wrong.

To contemplate all the bad cases I will compute:

$$\frac{n \delta}{n \varepsilon^2}$$

$\forall f \in F$ based on D_n

any f

$$P \left\{ \left| \mathbb{E}_{(x,y)} l(f(x), y) - \frac{1}{n} \sum_{i=1}^n l(f(x_i), y_i) \right| \geq \varepsilon \right\} \leq \frac{n \delta}{n \varepsilon^2} = \underline{\underline{\delta}}$$

$$\begin{aligned} \mathbb{E}_{(x,y)} l(f^*(x), y) &\leq \frac{1}{n} \sum_{i=1}^n l(f^*(x_i), y_i) + \sqrt{\frac{n \delta}{n \varepsilon^2}} \\ &\stackrel{\mu}{\leq} \frac{1}{n} \sum_{i=1}^n l(\bar{f}(x_i), y_i) + \sqrt{\frac{n \delta}{n \varepsilon^2}} + \sqrt{\frac{1}{n \delta}} \end{aligned}$$

\bar{f}

The true error is bounded by a term that is based upon how much data I have.

Not only that but I am also learning and this is from this term.

This formula tells also that one function will fit the model very good for full (Chaitin theorem). Statistical learning theory.

$$\min_{\mathbf{c}} \|\mathbf{x}_s - \mathbf{y}\|^2$$

$\frac{C}{n}$
Lemmas
that C
depends on p

$$\int_0^1 (\hat{f}^{(k)}(x))^2 dx \rightarrow \text{This measures the complexity of a function.}$$

↓
interval of my data.

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

$$f'(x) = \sum_{i=1}^p i c_i x^{(i-1)}$$

$$f''(x) = (f'(x))^2 = \sum_{i=1}^p i(i-1) c_i x^{(i-2)}$$

$$\int_0^1 (\hat{f}''(x))^2 dx = \int_0^1 \sum_{i=1}^p \sum_{j=1}^p c_i c_j (i-1) i(j-1) j x^{i+j-4} dx$$

$$\int_0^1 x^{i+j-4} dx = \frac{x^{i+j-3}}{i+j-3} \Big|_0^1 = \frac{1}{i+j-3}$$

↳ this because what's before is linear operation.

$$\int_0^1 (\hat{f}''(x))^2 dx = \sum_{i=1}^p \sum_{j=1}^p c_i c_j \frac{i(j-1)(j-1)}{i+j-3} = \mathbf{c}^T \mathbf{M} \mathbf{c}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

so At the end:

This regulates over and underfitting.
If $\lambda \rightarrow 0$ overfit, if $\lambda \rightarrow \infty$ underfit.

$$\min_{\mathbf{c}} \|\mathbf{x}_s - \mathbf{y}\|^2 + \lambda \mathbf{c}^T \mathbf{M} \mathbf{c}$$

RIDGE REGRESSION

Going back M is symmetric and semidefinite positive ($M \geq 0$)

$$\min_{\underline{c}} \underbrace{\|\underline{X}\underline{c} - \underline{y}\|_2^2}_{\text{problem}} + \lambda \underline{c}^T M \underline{c} \Rightarrow \text{problem at the end}$$

$M \geq 0$ and symmetric gradient = ϕ

\Downarrow
Ridge sol.

$$\nabla_{\underline{c}} (\underline{c}^T \underline{X}^T \underline{X} \underline{c} - 2 \underline{c}^T \underline{X} \underline{y} + \underline{y}^T \underline{y} + \lambda \underline{c}^T M \underline{c}) = \phi$$

$$2 \underline{X}^T \underline{x} - 2 \underline{X}^T \underline{y} + \phi + \lambda \lambda M \underline{c} = \phi$$

$$O(p^2) \in (\underline{X}^T \underline{X} + \lambda M) \underline{c} = \underline{X}^T \underline{y}$$

$$\underline{c} = (\underline{X}^T \underline{X} + \lambda M)^{\#} \underline{X}^T \underline{y}$$

depends on the λ does not depend on the data

$$(\underline{X}^T \underline{X} + \lambda M)$$

- ↳ $\lambda = 0 \Rightarrow$ min error
- ↳ $\lambda = \infty \Rightarrow$ with complexity

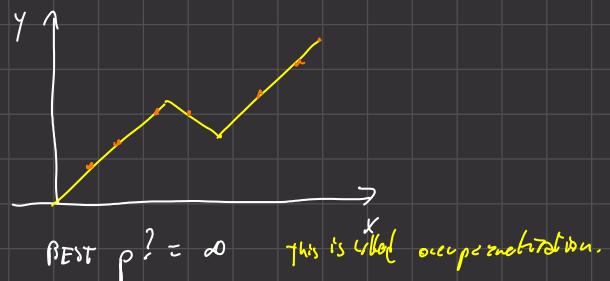
Suppose that M is an identity matrix:

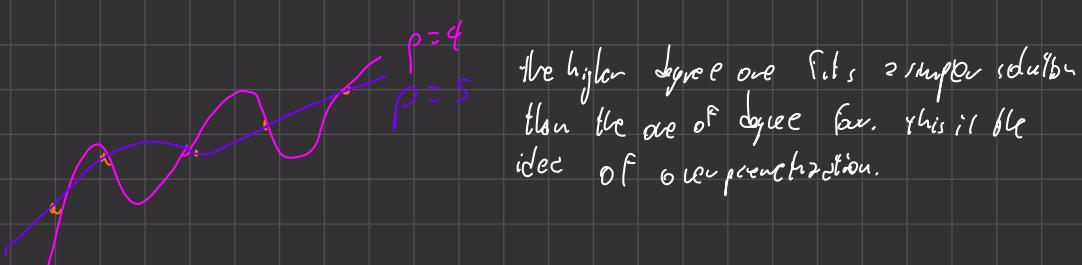
$$\underline{X}^T \underline{X} + \lambda I$$

↳ adding this you are adding to eigenvalues an eigenvalue.

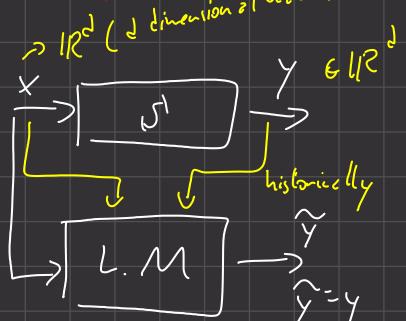
By adding λI it force the matrix to react in a strict way not depending on the data but only on my lambda. This is > Filter.

I weight small perturbations but "keep" only big ones





03/10/24 (dimensional vector)



We considered $\begin{cases} x \in \mathbb{R} \\ y \in \mathbb{R} \end{cases}$

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

We need to define the tool to transform x into y .
There is no optimal way of doing that.

We have mathematical constraints, but also computer constraint.

$$y = f(x) = \sum_{i=0}^p c_i x^i \quad (\text{we need a polynomial of degree } p.)$$

We choose the loss function:

$$l(f(x), y) = \underbrace{(y - f(x))^2}_{\hookrightarrow \text{distance}}$$

Once defined all those things we needed to defined the optimal value for this parameter.

Now we ask what is the best const of f given c knowing the points.

The best thing is to minimize the error on my data.

$$\min_c \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

The problem here is that it assumes that we know what we are doing. In some cases this approach does not work (degree of polynomial equal to number of samples because error always \Rightarrow which is impossible)

We want to be good on previously unseen data NOT historical data.
 We used statistics and discovered that if the dataset is IID.

The more data \rightarrow the system does not change in time
 The more info I have

and choose f from a set of function independently from F .

$f \in F$ independently F

(Error of the function on the data)

we know that the true error is bounded by the empirical error + something which goes like

$$R(f) \leq \widehat{R}(f) + \sqrt{\frac{1}{n\sigma}} \quad R(f) = \left| E_{(x,y)} f(x, y) \right|$$

Expected value

$$\widehat{R}(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$$

empirical risk ↓
(error that I measure on my data) empirical average.

For now we are just doing statistics, and if n (number of samples) goes to infinity then the true error is equal to the empirical error.

Since we are in statistics we have probability of getting something right at this probability is the **confidence**.

The higher the confidence the higher the risk, so I could be far away from the data

If instead f is independent from the dataset, but f is chosen in F using the dataset:

Find D_n , $f \in F$ using D_n

Let's say that we have two bags: one of data and one of functions the bag of functions is chosen before the one of data. Then I look at the data and I choose the best function between the ones that I have

(statistical learning theory)

Now we have

$$R(f) \leq \widehat{R}(f) + \sqrt{\frac{n_f - 1}{n\sigma}} + \sqrt{\frac{1}{n\sigma}}$$

(1 - α)
probability

When I learn something from the data I don't just wish that my data are not enough or do not represent the population but I also wish that the function that I select works well on those data just by luck. (charleson problem)

(H. P. = If I check enough functions I will find one that works well
on that data probabilistically speaking)

The problem of just minimizing the error on my data is that I am not taking into account that I cannot choose whatever function I want, but I have to select a function that is simple, because it means that I trade off error on my data with simplicity. The simpler, the smaller the chance of over fitting.

$$\min_{\mathcal{S}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \underbrace{\mathcal{C}(f)}_{\text{Complexity of the function}} \quad \text{(trade off error and simplicity)}$$

In matrix form:

$$\min_{\mathcal{S}} \|X\mathbf{c} - \mathbf{y}\|^2 + \lambda \mathcal{C}(\mathbf{M})$$

Solution: $\underbrace{(X^\top X + \lambda \mathbf{I})}_{(p \times p)(p \times p)} \mathbf{c} = X^\top \mathbf{y}$ $\mathcal{O}(p^2)$

When I measure my system we know that there are some noises that affect my measurement.

Noise \rightarrow measure

\rightarrow partial knowledge on X

In order to predict y we may need many x but we cannot have them for many reasons. (money, many approach)

The combination of not minimizing the error but compromising simplicity and accuracy and noise we discover first even if we knew something about the system it is useless because the noise could make everything wrong.

Since we are adding the complexity term the logical rule that the more we increase p the more the model becomes complex is no more true. Here the complexity is described by the second derivative.

- - - - -

- Function approximation works by defining a positive error and below a certain value. If we increase the elements for example in tailor the error decrease.

- In ML first of all we take a sample and we say that this is a distribution which is IID.

The difference between sampling and Tailor is that in Taylor you "look at all the function while sampling only the sampled part is relevant."

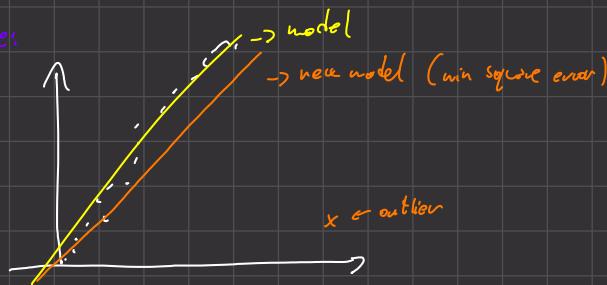
We always talk about expected error or empirical error because if x any

$$\mathbb{E}_{(x,y)} l(f(x), y) \quad \sum_{i=1}^n l(f(x_i), y_i)$$

Does not appear it doesn't care.

In ML I don't make the model work well point-wise, but I make it work well w.r.t. the data generating distribution

Example:



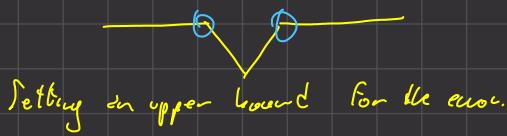
In order to remove the problem, just remove the outlier. But that's a dog following its tail.

In order to mitigate the problem we could change the measure of the error:

$$(y - f)^2 \rightarrow |y - f|$$



What I would want is



The good of the quadratic is that is differentiable and convex.

The modulus is convex but not diff.

The third one is non convex because there are concave parts

The problem of optimization with concave functions is that we don't have efficient solutions. With convex we have polynomial solutions. Non convex \Rightarrow not polynomial.

In order to solve:

$$\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{y}\|^2 + \lambda \mathbf{c}^\top \mathbf{M} \mathbf{x} \quad \text{we need } O(p^2) \text{ lin. system}$$

Since its squared we cannot really increase p how much we want.

Remember that we want to find a minimum in a concave function.

GRADIENT DESCENT

$$\min_x f(x) \quad x_0 = \emptyset$$

$$x_{i+1} = x_i - \gamma \nabla f(x)|_{x_i}$$

↓ step size or LEARNING RATE

GRADIENT
DESCENT

This approach does not depend on the number of parameters that I need to optimize. It does not depend on γ .

This algorithm depends on the space that it has to travel, but this depends on the number of iterations and on the learning rate.

There is a theorem that states that:

Given a function and using gradient descent if the gradient of the function has a norm that behaves like: $\|\nabla f(x) - \nabla f(y)\| \leq L \|x-y\|^2$ (the gradient is Lipschitz which means that γ does not change too much). I am also stating that there are no singular points.

If $\gamma \leq \frac{1}{L}$ I am sure that I will never go too far away from the point because at maximum it will be dividing the gradient by L .

The problem is that L is a worse case approach, so even if the function becomes very flat you will go slower if L very large.

$$|\nabla f(x^*) - \nabla f(x_n)| \leq \frac{\|x_0 - x^*\|^2}{2LK} \approx \underline{\underline{\Omega(\frac{1}{n})}}$$

optimal point ↓
 1th point (iterations)

The accuracy goes down linearly.

Let's take Ridge regression in mono space:

$$\min_{c_0 \dots c_p} \frac{1}{n} \sum_{i=1}^n \left(y_i - \sum_{j=0}^p c_j x_i^j \right)^2 + \lambda c^T M c$$

We start from $c = \emptyset$

$$c_{i+1} = c_i + \gamma \nabla_c \left(\dots \right) |_{c_i}$$

$$= c_i + \gamma \left(\frac{1}{n} \sum_{i=1}^n \nabla_c \left(y_i - \sum_{j=0}^p c_j x_i^j \right)^2 + 2\lambda M c \right)$$

this can be computed in parallel on different PCs on different sets of jobs, then I can just send a vector of dimension n .

Since these are just matrix-vector sums you can use the GPU.

$$\min_{c \in C_0} \frac{1}{n} \sum_{i=1}^n \left(y_i - \sum_{j \neq i} c_j x_i^j \right)^2 + \lambda c^T M c$$

This could be substituted by another with different complexity.
The simplest is the identity, $\Rightarrow c^T c = \|c\|^2$

$$\min_{C} \|X_C - y\|^2 + \lambda \|c\|^2 \quad \underset{\text{---}}{\substack{\min \\ \leq}} \quad \frac{\min \|x_C - y\|^2}{\|c\|^2 \leq A} \quad (R)$$

they are the same

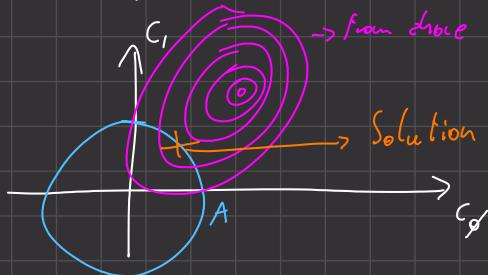
Suppose that the solution to (L) is c^* and its norm is $\|c^*\| = A$
Now I put A in (R) and the solution is c^* because c can go up to A and we know the complexity increases if we diminish the error.

In (R) we have to minimize the error so search for c with maximum complexity, so it reaches A .

So if I find $\|c\| = A$ with smaller error than the one ^{reached} it means that before I didn't minimize the error. (PROOF BY ABSURDO)

The second problem can be repeated in 2D:

The upper is a pedobold contained somewhere

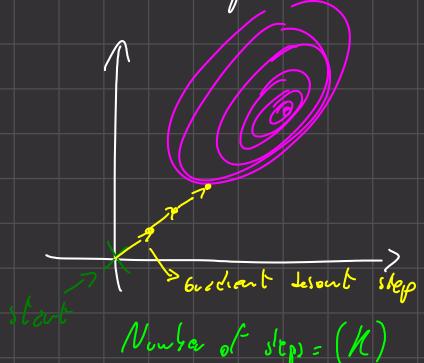


Now I remove completely the complexity, but in order to compute this I use the Gradient Descent

$$\min_{C} \|X_C - y\|^2$$

$$c_0 = \emptyset$$

$$c_{in} = c_i - \gamma \nabla_{c_i} \|X_C - y\|^2 \Big|_{c_i}$$



K is the same as A or λ . It's just another way of regulating the complexity of the result.

This is called early stopping.

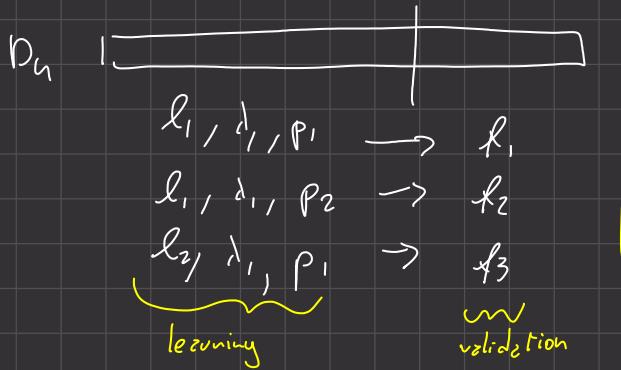
this because the more steps \hat{c} do the more I look at the data \rightarrow step = look
this is an indicator of how many times you look at the Y-axis

γ is the learning rate or how much time you spend on each page of the book

Now the problem is how to set the hyperparameters. Well we don't have a clue. We can for example search the other in $O(p^2)$

$D_n \rightarrow \text{RiDiGf} \rightarrow \subseteq, P, \wedge \Rightarrow f \rightarrow$ I need to test this function.

So I take my dataset and I split it in two parts, and I can do this because it's IID



I am using always the same validation set because I don't have so much data.

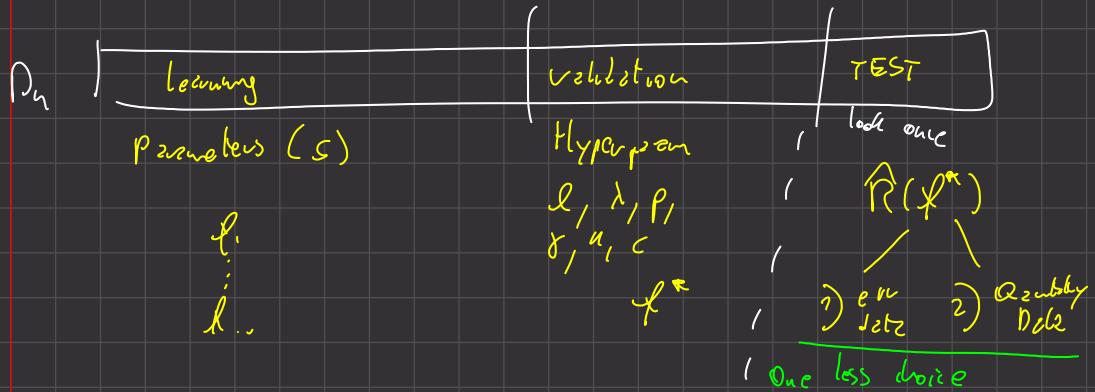
Both on the left and on the right I have a needle or awl on the table, limited tools, and matches to do.

On the left I will securitizing and on the right overvaluation.

Cross validation is what helps with overfitting, so repeating calculation on different splits.

{ 70% of job for learning
30% for validation.

To see the performance of the model we do:



$$R(\phi) \leq \hat{R} + \sqrt{\epsilon} + \sqrt{\kappa}$$

$$R(\phi) \leq \hat{R}(\phi) + \sqrt{\frac{1}{n\delta}}$$

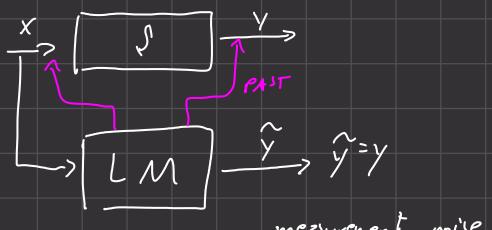
8/10/24

$x \in \mathbb{R}$

$y \in \mathbb{R}$

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

i i d
the more
date the better
system
not changing in time



$\hat{y} \neq y$ because of
partial knowledge of x

$$\hat{y} = f(x)$$

$$R(f) = \mathbb{E}_{(x,y)} \ell(f(x), y) \text{ real error}$$

$$\ell(f(x), y) \quad \hat{R}(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) \text{ empirical error}$$

↑ complexity of the function.

(f)

Now we can build the algorithm:

$$\min_f \hat{R}(f) + \lambda C(f)$$

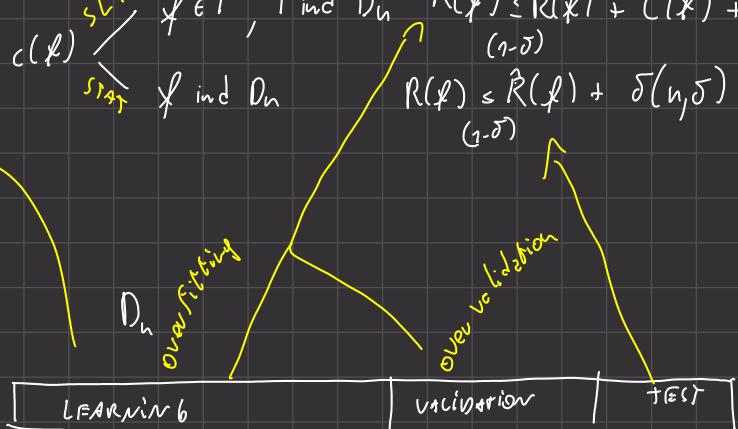
(\rightarrow this needs to be tuned to have the trade-off between accuracy and complexity)

This approach is theoretically grounded because when f is chosen inside a set where f is independent from a dataset is bounded by the empirical risk κ + a complexity term + something that depends on the number of samples + probability statement.

$$\min_f \hat{R}(f) + \lambda C(f) \quad \begin{array}{l} \text{SCT} \\ \text{SVM} \end{array} \quad f \in F, \text{Find } D_n \quad R(f) \leq \hat{R}(f) + C(f) + \delta(n, \delta)$$

$$R(f) \leq \hat{R}(f) + \delta(n, \delta)$$

$f?$
 $\ell?$
 $\lambda?$
 $C?$
 \dots



SCT
choice
 $f \in F$

SCT
choice

one
statistics

Suppose now

$$x \in \mathbb{R}^d \quad D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

First thing to do is to write the function

$$\hat{y} = f(x) = c_{0,0} + \underbrace{c_{1,1} x_1 + c_{1,2} x_2 + \dots + c_{1,d} x_d}_{\binom{d}{1}} + \underbrace{\left[c_{2,1} x_1^2 + \dots + c_{1,d} x_d^2 + c_{1,d+1} x_1 x_2 + \dots + c_{1,d-1} x_{d-1} x_d \right]}_{\binom{d}{2}} + \underbrace{\left[c_{3,1} x_1^3 + \dots + c_{3,d} x_d^3 + c_{3,d+1} x_1 x_2^2 + \dots + c_{3,d-1} x_{d-1} x_d^2 + c_{3,d-2} x_1 x_2 x_3 \right]}_{\binom{d}{3}} + \dots + \underbrace{c_{d,1} x_1^d}_{\binom{d}{d}}$$

P

Taylor extended to multi dimension

$$\binom{d}{1}, \binom{d}{2}, \binom{d}{3}, \dots, \binom{d}{\rho}$$

The problem is that the number is enormous:

$$d = 10^{12} \quad \rho = 15 \quad \binom{d}{\rho} \underset{\downarrow}{\sim} d^\rho \underset{\text{Stirling approximation}}{\sim} (10^{12})^{15} = 10^{150}$$

Classical methods cannot be used in this context.

$$y = f(x) = b + \underbrace{\omega x}_{\text{constant}} \rightarrow \text{We linearize the model and remove the}$$

$$\ell(f(x), y) = (y - f(x))^2$$

$$C(f) = \omega^\top \omega = \|\omega\|^2 \quad \text{this is true by analogy}$$

$$\min_w \sum_{i=1}^n (\underline{y}_i - \underline{x}_i^\top \underline{\omega})^2 + \lambda \|\omega\|^2$$

$$\underline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n \times d}$$

$$\min_w \underbrace{\|\underline{x}_w - \underline{y}\|^2}_{\text{nx1}} + \lambda \|\omega\|^2$$

$$\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

$$\min_w \underbrace{\|Xw - y\|^2}_{\text{Paraboloid}} + \underbrace{\lambda \|w\|^2}_{\text{Paraboloid}} = \text{Paraboloid}$$

Ridge-
Regression
Primal
Formulation

$$w^T X^T X w - 2 w^T X^T y + y^T y + \lambda w^T I w$$

$$\nabla_w (*) = \phi$$

$$X^T X w - X^T y + \phi + \lambda I w = \phi$$

$$(X^T X + \lambda I) w^* = X^T y$$

$$\underbrace{O(d^2)}_{d \times d} \quad \underbrace{w^*}_{\substack{d \times n \\ \text{and}}} = \underbrace{(X^T X + \lambda I)^{-1}}_{d \times d} X^T y \quad O(d^3)$$

The solution is quadratic in the number of dimensions

This is good because this algorithm is quadratic in the number of dimensions but not in the number of samples

$$\min_w \|Xw - y\|^2 + \lambda \|w\|^2$$

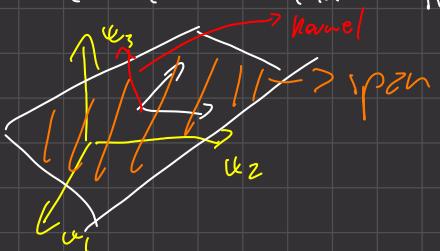
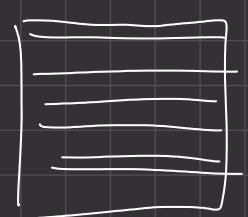
$$(X^T X + \lambda I) w^* = X^T y$$

$$w^* = X^T \alpha = \sum_{i=1}^n \alpha_i x_i$$

Remember

$X^T w \rightarrow \text{parallel}$
 $X^T w \perp \rightarrow \text{perpendicular}$
 $X^T w = \phi \rightarrow \text{Kernel}$

The lines represent subspace and the lines lie in the subspace



Proof by induction

$$w^* = X^T \alpha = \sum_{i=1}^n \alpha_i x_i = w^T I$$

By absurd I assume : $w^* = w_u + w_v$

$$\|x(\omega_{\text{ff}} + \omega_+) - y\|^2 + \lambda \underbrace{\|\omega_{\text{ff}} + \omega_+\|^2}_{\geq 0} \rightarrow \dots$$

$$= \|x\omega_{\text{ff}} + \phi - y\|^2 + \lambda (\|\omega_{\text{ff}}\|^2 + \|\omega_+\|^2)$$

$$= \underbrace{\|x\omega_{\text{ff}} - y\|^2 + \lambda \|\omega_{\text{ff}}\|^2}_{(\geq \text{less than})} + \underbrace{\lambda \|\omega_+\|^2}_{\geq 0} \text{ because I assumed } \omega_+ \text{ to be present}$$

So since ω_{ff} is smaller than ω^* we reached an optimum since ω^* must be the minimum so:

$$\underline{\omega^* = X^\top \alpha}$$

$$\min_{\alpha} \|X\alpha - y\|^2 + \lambda \alpha^\top X^\top X \alpha$$

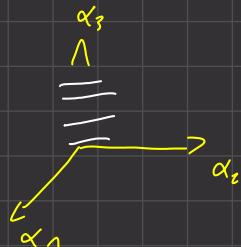
$$Q = X^\top X \stackrel{n}{\underset{d}{\geq}} \begin{bmatrix} x_1^\top \\ \vdots \\ x_d^\top \end{bmatrix} \in \begin{bmatrix} \cdot \\ \vdots \\ \cdot \end{bmatrix} \rightarrow Q_{ij} = x_i^\top x_j$$

Properties of Q :

$$Q_{ij} = Q_{ji} \quad \text{symmetric}$$

$$Q \geq \phi \quad \text{semi-definite positive}$$

$$\boxed{\min_{\alpha} \|Q\alpha - y\|^2 + \lambda \alpha^\top Q \alpha}$$



$$\nabla_{\alpha} (\alpha^\top Q^\top Q \alpha - 2\alpha^\top Q^\top y + y^\top y + \lambda \alpha^\top Q \alpha) = \phi$$

$$\cancel{2} Q^\top Q \alpha - \cancel{2} Q^\top y + \cancel{2} \lambda Q^\top \alpha = \phi$$

$$\boxed{(Q + \lambda I)\alpha = y} \rightarrow \text{linear system of which we can do the pseudoinverse}$$

$$\boxed{(X^\top X + \lambda I)w^* = X^\top y}$$

Dual

work

work

work

$O(n^2)$

$\frac{\text{pseudoinv}}{\text{dyn work}} \frac{\text{dyn work}}{\text{work}} \frac{\text{work}}{\text{work}}$

$O(d^2)$

w^* can adapt

$$\underline{\omega^* = X^\top \alpha} \quad \text{Reproducing Theorem.}$$

We may be yet the Ridge regression in multi-dimension. The function is that we assumed the model to be linear.

$$\min_{\underline{w}} \|\underline{X}\underline{w} - \underline{y}\|^2 + \lambda \|\underline{w}\|^2 \quad f(\underline{x}) = \underline{w}^* \cdot \underline{x}$$

$$(\underline{X}^T \underline{X} + \lambda I) \underline{w}^* = \underline{X} \underline{y}$$

$$\underline{w}^* = \underline{X}^T \alpha^*$$

$$(\underline{Q} + \lambda I) \alpha^* = \underline{y} \quad f(\underline{x}) = \underline{w}^{*T} \underline{x} = \alpha^{*T} \underline{X} \underline{x} = \sum_{i=1}^n \alpha_i \underline{x}_i^T \underline{x}$$

$\hookrightarrow Q_{ij} = \underline{x}_i^T \underline{x}_j$
 $\alpha^* = \underline{X}^{-1} \underline{y}$

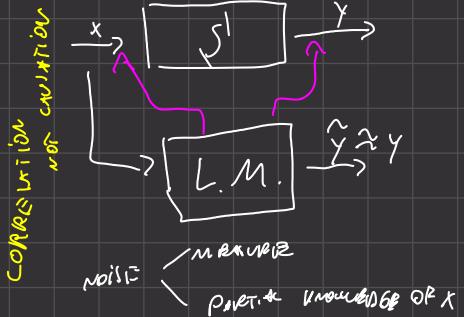
20/10/24

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\} \text{ in d}$$

$$\underline{x} \in \mathbb{R}^d \quad y \in \mathbb{R}$$

$$f(\underline{x}) = \underline{w} \cdot \underline{x}$$

We use a linear model because it's simple and allows us to use all the inputs



$$l(f(\underline{x}), y) = (f(\underline{x}) - y)^2$$

$$C(f) = \underline{w}^T \underline{I} \underline{w} = \|\underline{w}\|^2$$

$$\min_{\underline{w}} \|\underline{X}\underline{w} - \underline{y}\|^2 + \lambda \|\underline{w}\|^2$$

$$\underline{X} = \begin{bmatrix} \underline{x}_1^T \\ \vdots \\ \underline{x}_n^T \end{bmatrix} \in \mathbb{R}^{n \times d}$$

$$\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^{n \times 1}$$

$$\rightarrow (\underline{X}^T \underline{X} + \lambda I) \underline{w}^* = \underline{X}^T \underline{y} \quad O(d^2)$$

$$\underline{w}^* = \underline{X}^T \alpha^*$$

$$O(n^2)$$

$$(\underline{Q} + \lambda I) \alpha^* = \underline{y}$$

$\hookrightarrow Q_{ij} = \underline{x}_i^T \underline{x}_j \rightarrow$ Note, I only used to know the initial parameters.

$$f(\underline{x}) = \underline{w} \cdot \underline{x} = \sum_{i=1}^n \alpha_i \underline{x}_i^T \underline{x}$$

Problems:

- X is not always b-dimensional
- Who told me that $\|w\|^2$ is a good complexity measure
- Is linear a good model (Ridge regression)?
- How to choose f, ℓ, C, λ and algorithm for min

$$(x_i, y_i) \quad C(f) = \|w\|^2$$

$\xrightarrow{\text{Can be stated as}}$

$$R(f) \leq \hat{R}(f) + " \alpha_n " + " \delta_n "$$

$(2-\delta)$

This formula is a bit naive because the number of function is infinite

Now we want to bound the distance: $R(f) - \hat{R}(f)$

$$\mathbb{E}_{D_n} \{ R(f) - \hat{R}(f) \} \leq \dots$$

\hookrightarrow I want to check what happens if I repeat the measure many times

$R(f)$ is random because depends on D_n , but now we chose E_{D_n} now it's deterministic.

$$\mathbb{E}_{D_n} \{ R(f) - \hat{R}(f) \} \leq \mathbb{E}_{(x_1, y_1) \dots (x_n, y_n)} \left\{ \mathbb{E}_{(\hat{x}_i, \hat{y}_i)} \ell(f(x), y) - \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) \right\}$$

$$= \mathbb{E}_{(x_1, y_1) \dots (x_n, y_n)} \left\{ \underbrace{\mathbb{E}_{(\hat{x}_1, \hat{y}_1) \dots (\hat{x}_n, \hat{y}_n)} \left[\frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i) - \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) \right]}_{\text{More simple}} \right\}$$

You assume you have it

$$= \mathbb{E}_{(x_1, y_1) \dots (x_n, y_n)} \left\{ \frac{1}{n} \sum_{i=1}^n \left[\ell(f(x_i), y_i) - \ell(f(\underline{x}_i), \underline{y}_i) \right] \right\}$$

$$\leq \mathbb{E}_{(x_1, y_1) \dots (x_n, y_n)} \left\{ \sup_{f \in F} \frac{1}{n} \sum_{i=1}^n \left[\ell(f(x_i), y_i) - \ell(f(\underline{x}_i), \underline{y}_i) \right] \right\}$$

$$= \mathbb{E}_{(x_1, y_1) \dots (x_n, y_n)} \left\{ \sup_{f \in F} \frac{1}{n} \sum_{i=1}^n \partial_i \left[\ell(f(x_i), y_i) - \ell(f(\underline{x}_i), \underline{y}_i) \right] \right\}$$

Note that adding ∂ does not change anything because all the terms replicated twice because of this

$\partial_1 \dots \partial_n$

∂_i iid

$$\partial_i \in \{-1, +1\} \quad P\{\partial_i = 1\} = P\{\partial_i = -1\} = \frac{1}{2} \quad \text{Redundant random variables}$$

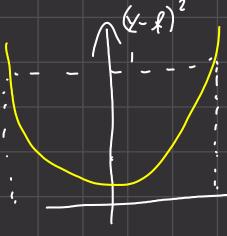
$$\leq \mathbb{E}_{\substack{(x_1, y_1), \dots, (x_n, y_n) \\ (x'_1, y'_1), \dots, (x'_n, y'_n)}} \left\{ \sup_{f \in F} \frac{1}{n} \sum_{i=1}^n \partial_i [l(f(x_i), y_i) + l(f(x'_i), y'_i)] \right\}$$

$\partial_1 \dots \partial_n$

$$= \mathbb{E}_{\substack{(x_1, y_1), \dots, (x_n, y_n) \\ \partial_1 \dots \partial_n}} \sup_{f \in F} \frac{2}{n} \sum_{i=1}^n \partial_i l(f(y_i), f(x_i))$$

Rademacher Complexity

Let's now consider the square loss:



$$\boxed{\begin{array}{l} \text{It's Lipschitz} \\ |g(x) - g(x')| = L|x - x'| \\ \text{Lip constant} \end{array}}$$

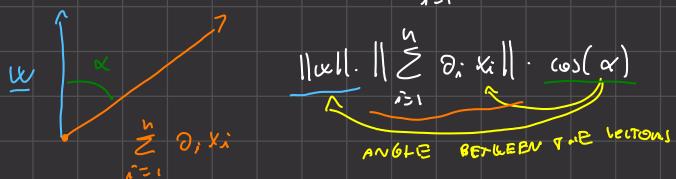
$$|f(x) - f(x')| \leq L|x - x'|$$

$$\mathbb{E}_{\substack{(x_1, y_1), \dots, (x_n, y_n) \\ \partial_1 \dots \partial_n}} \sup_{f \in F} \frac{2}{n} \sum_{i=1}^n \partial_i l(f(y_i), f(x_i)) \leq$$

$$\leq \mathbb{E}_{\substack{(x_1, y_1), \dots, (x_n, y_n) \\ \partial_1 \dots \partial_n}} \sup_{f \in F} \frac{2L}{n} \sum_{i=1}^n \partial_i \underbrace{f(x_i)}_{w \cdot x_i} =$$

$w^* = (x^T x, 1) x^T y$

$$= \mathbb{E}_{\substack{(x_1, y_1), \dots, (x_n, y_n) \\ \partial_1 \dots \partial_n}} \sup_{\|w\|^2 \leq \|w^*\|^2} \frac{L}{n} \sum_{i=1}^n \partial_i w^T x_i =$$



Note in order to have the supremum $\cos(\alpha) = 1$, and $w = w^*$ with w^* the maximum

This is a constant, but I didn't know it.

$$= \mathbb{E}_{\substack{(x_1, y_1), \dots, (x_n, y_n) \\ \partial_1 \dots \partial_n}} \sup_{\|w\|^2 \leq \|w^*\|^2} \frac{2L}{n} \|w^*\| \left\| \sum_{i=1}^n \partial_i x_i \right\|$$

proportional

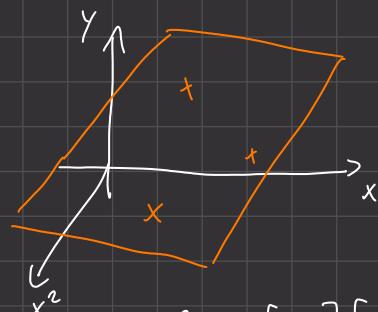
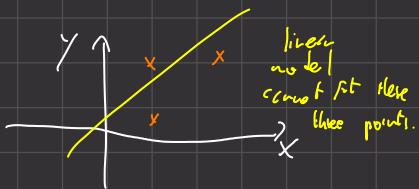
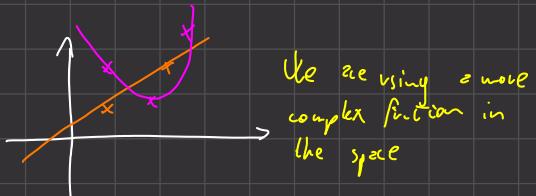
$$\propto \boxed{\|w^*\|}$$

$R(f)$ risk on debt
 $\hat{R}(f)$ risk on population

But we've proved that the distance between $R(f)$ and $\hat{R}(f)$ is proportional to itself so it does make sense that the complexity is measured with $\|w\|^2$

Generalize to non linear model

$$x \in \mathbb{R} \quad f(x) = \sum_{i=0}^p c_i x^i$$



$$f(x) = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \end{bmatrix} = w \phi(x)$$

One could see the measure of complexity of x as the inverse of space.

$$f(x) = \sum_{i=0}^p c_i x^i = w' \phi(x) = \begin{bmatrix} w_0 \\ \vdots \\ w_p \end{bmatrix} \begin{bmatrix} \phi_0(x) \\ \vdots \\ \phi_p(x) \end{bmatrix} = \begin{bmatrix} w_0 \\ \vdots \\ w_p \end{bmatrix} \begin{bmatrix} x^0 \\ \vdots \\ x^p \end{bmatrix}$$

? Model for the ridge regression

$$f(x) = w' x \rightarrow \underline{w' \phi(x)}$$

Non of its
non linear

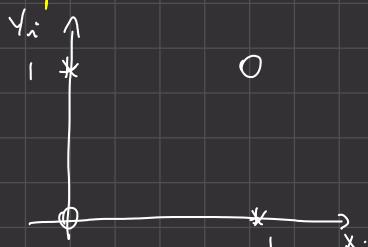
Non linear blocks to ϕ

$$= \sum_{i=1}^n y_i^T x_i x \rightarrow \sum_{i=1}^n \alpha_i \underline{\phi(x_i)^T \phi(x_i)}$$

$$\alpha = (Q + \lambda I)^{-1} y$$

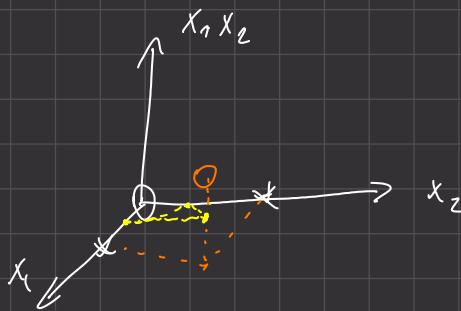
$$\hookrightarrow Q_{ij} = y_i^T x_j \rightarrow \underline{\phi(x_i)^T \phi(x_j)}$$

Example:



In this configuration there is no linear model
to split circles in sets

We apply ϕ to go in higher dimension:



Now we have a place able to split circles and stars.

Using a linear model in ϕ 's equivalent of may be nonlinear in the first space. This just by applying $\phi(x)$ to x .

$$\phi(x) = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \\ x_1^2 \\ \vdots \\ x_d^2 \\ x_1 x_2 \\ \vdots \\ x_d x_d \end{pmatrix} \in \mathbb{R}^d$$

N.P problem if using ϕ so many.

$$\binom{d}{p} \approx d^p$$

To solve this problem we will use Mercer and kernels.

Kernels are a class of function that takes two vectors and return a number:

$$K(u, v) \rightarrow \mathbb{R} \quad K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$$

$$K(u, v) = \phi^T(u) \phi(v) \quad \text{where } \phi \text{ unknown}$$

A kernel function is a kernel when the integral w.r.t u and v :

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K(u, v) g(u) g(v) du dv \geq 0 \quad \text{by, } \int_{\mathbb{R}^d} g(x)^2 dx < \infty$$

Example of kernels:

$$\left\{ \begin{array}{l} K(u, v) = e^{-\gamma \|u-v\|^2} \\ K(u, v) = (u^T v)^p \\ K(u, v) = (v^T u + b)^p \\ K(u, v) = e^{-r \|u-v\|_1} \end{array} \right.$$

The Mercer theorem is very important because we can just put the kernel

$$\sum_{i=1}^n \alpha_i K(x_i, x) + K(x_i, x_j)$$

even if I don't know the ϕ that generated the kernel

$$K(u, v) = e^{-\gamma \|u - v\|^2}$$

Gaussian / RBF Kernel

Let's go one-dimensional with this hyperplane.

$$K(u, v) = e^{-\gamma(u-v)^2} = e^{-\gamma u^2 - \gamma v^2 + 2uv} = *$$

Using the McLaurin series which is the Taylor series centered in ϕ :

$$e^x = \sum_{i=0}^{\infty} \frac{d^i e^x}{dx^i} \Big|_{x=0} \frac{1}{i!} x^i = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

$$* = e^{-\gamma u^2} e^{-\gamma v^2} \sum_{i=0}^{\infty} \frac{(-2\gamma uv)^i}{i!} \rightarrow 1 + 2uv + \frac{2^2}{2!} u^2 v^2 + \frac{2^3}{3!} u^3 v^3 + \dots$$

$$= e^{-\gamma u^2} \begin{bmatrix} 1 \\ \sqrt{2\gamma} u \\ \sqrt{\frac{2\gamma^2}{2!}} u^2 \\ \sqrt{\frac{2\gamma^3}{3!}} u^3 \end{bmatrix}^\top e^{-\gamma v^2} \begin{bmatrix} 1 \\ \sqrt{2\gamma} v \\ \sqrt{\frac{2\gamma^2}{2!}} v^2 \\ \sqrt{\frac{2\gamma^3}{3!}} v^3 \end{bmatrix} = \phi(u)^\top \phi(v)$$

$\phi \in \mathbb{R}^\infty$

OVER-parametrization

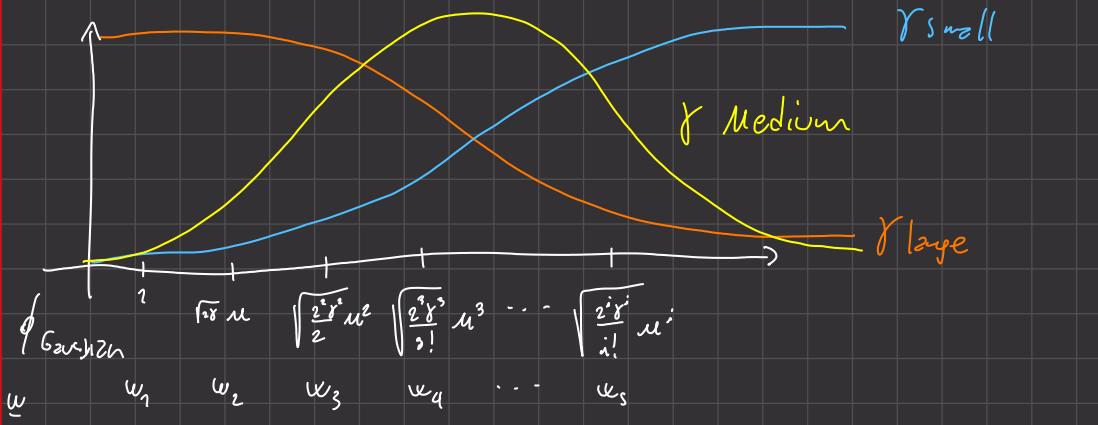
With the Gaussian kernel I projected my data in an infinite dimension space and then I can fit them - an infinite model

The problem here is that I can over-fit my data, but we can apply the L_2 norm to regularize and choose simple functions in this infinite space.

Using Gaussian kernel (Kernel Ridge Regression) It includes two additional hyperparameters, the hyperparameter of the kernel and the kernel itself.

1:39:35

Let's understand the effect of gamma on the solution



I cannot choose w too big because limits complexity,

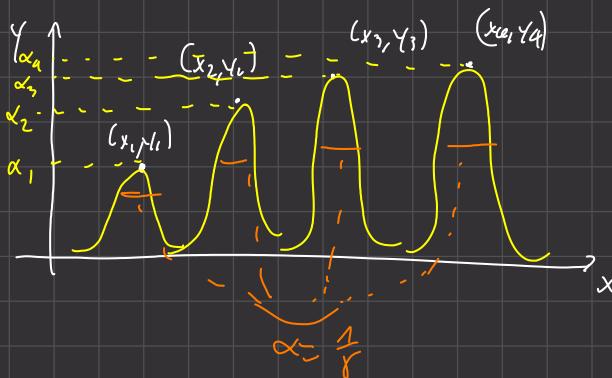
$\gamma_{\text{small}} \rightarrow \text{linearity}$
 $\gamma_{\text{large}} \rightarrow \text{non-linearity}$

This because w needs to be the same power of the solution to have an effect, but the bigger, the more complex \rightarrow non-linearity.

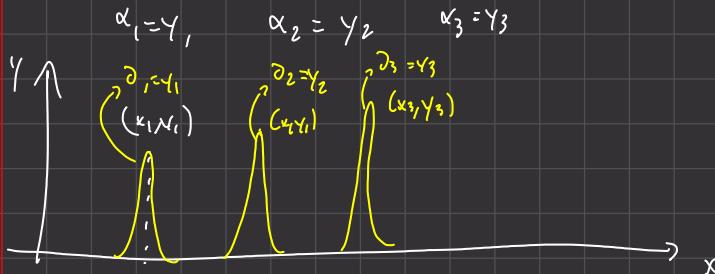
Another perspective

$$f(x) = w^* \phi(x) = \sum_{i=1}^n \alpha_i K(x_i, x) = \sum_{i=1}^n \alpha_i e^{-\gamma \|x_i - x\|^2}$$

Gaussian centered in x_i
 with variance $\sigma^2 = \frac{1}{\gamma}$



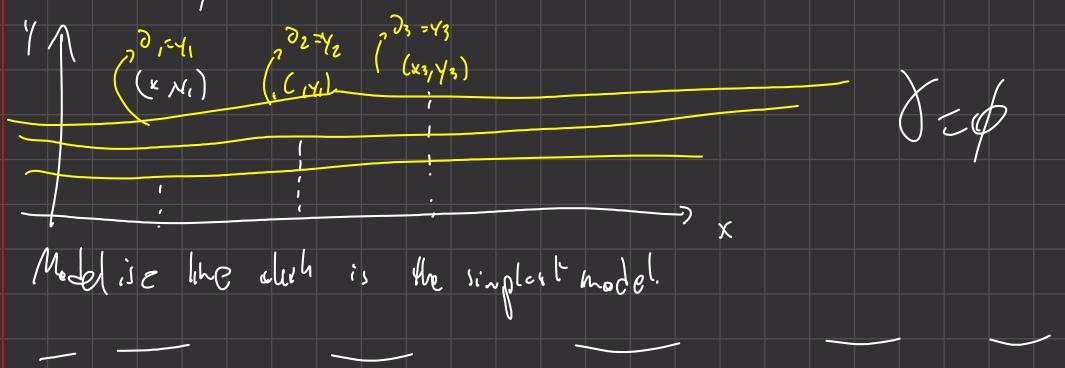
Suppose now that γ is very large, then the sum of these Gaussians



$$y_i \in f(x_i) \quad \forall i \in \{1, \dots, n\}$$

This solution is over fitting when γ very large

If γ is very small

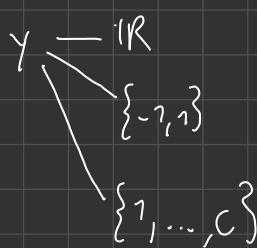


Model is like which is the simplest model.

$$f(x) = \omega^* x = \sum_{i=1}^n \alpha_i^* k(x_i, x)$$

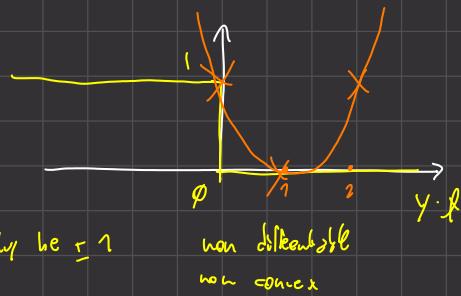
$$\mathcal{Q}_{ij} = k(x_i, x_j)$$

$$k(u, v) = e^{-\gamma \|u - v\|^2}$$



With kernel ridge regression we can deal with all of them

$y \in \{-1, 1\}$ In the binary domain we care about the product between y and f



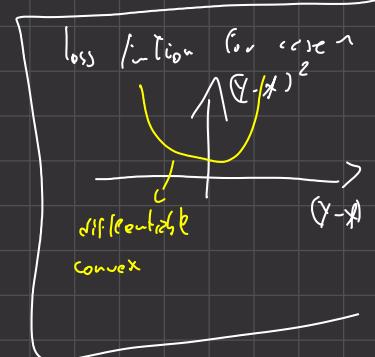
and y can only be ± 1

(let's say with $y \neq 0$) then $y = \pm 1$

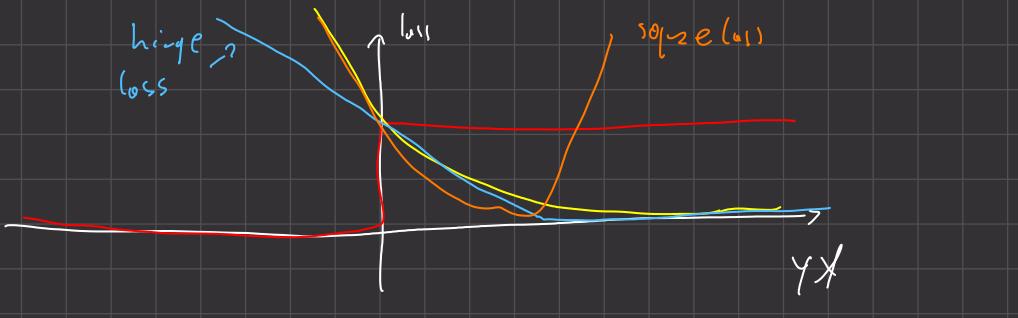
$$\text{since } y \neq 0 \rightarrow f = \phi \rightarrow (y-f)^2 = 1$$

$$y \neq 1 \rightarrow f = y \rightarrow (y-f)^2 = 0$$

$$y \neq -1 \rightarrow f = -y \rightarrow (y-f)^2 = 1$$



K.R.R is used in
binary classification.



Multiclass Classification

$$y \in \{1, 2, 3, \dots, C\}$$

$$(f(x) - y)^2$$

$$f_1 = 6 - 1 \rightarrow 5$$

$$f_2 = 2 - 1 \rightarrow 1$$

many because
there is no concept
of distance in class

For binary case ok because
you had just two classes.