UNIT 2 : Non Linear Classification, Linear Regression and Colaborative Filtering

LECTURE 5 : Linear regression

Wooooooooo !!!!!!! We are back baby !!!!! Yeahhhhhh !!!!!

After the successful completion of Fundamentals of Statistics , your pal **Scooby** is back with a brand new set of handwritten lecture notes. I found that many of you liked them and appriciated the work by thanking me personally on my email. Which by the way is :

Scooby's email : **peepeepoopoo873@gmail.com**

Feel free to point out mistakes and message me anyways

For those who do not message me...
I don't care man!!! Whataever man !!!! Screw you guys !!!! Whatever!!!

(I do care, please message me. I was just pretending to be indifferent in the above line so that you can't hurt me emotionally.) Anyways.

I tried to make these notes without any mistakes and cover most of the video material.

But as Yoda has rightly said : ' *Is human, To err* '

I however **DO NOT TAKE** any responsibilty if you fail your exams following my notes. So please check the actual lecture slides and videos if you suspect something is wrong, and mail me too.

The notes **DO NOT CONTAIN** solutions to any homework problems or in-lecture exercises. Only the problems and exercises solved in the videos are included in these notes.

Here is an interesting photo I googled

CROOKED FOREST

I ran out of made up quotes so I will write very short stories instead. These may or may not resemble true events, but will definitely have a message that will help you along the journey, that is life.

STORY TIME : The ripple of good deed

Once upon a time there was a musician really passionate about her craft. She for the most part of her life played her guitar and sang in clubs and parties in order to continue her career and pay the bills. One evening as she was returning from one of her gigs at a club, she saw a boy weeping in the alley.

It was dark so she hesitated going there, but finally gathered up the courage to check out what wa happening. She saw a little boy around the age of 13 was sitting there crying non stop. She asked her what the matter was, but the boy won't stop crying. She asked repeatedly but he wouldn't stop.
Out of nowhere she got an idea. She took her guitar out of the bag and started singing a song her mother used to sing to her when she was a child. The song was about being brave and facing challenges head on. To fight like a worrior and and never give up.

The boy stopped sobbing. She continued. The boy looked into her eyes. The song continued. For another minute or two the boy continued looking into her eyes as the song continued.
Finishing the song she again asked the boy what the matter was. The boy said he was sad but couldn't tell her the reason. He thanked her for cheering him up, and said ' I hope I can hear you sing again some day' . With that said the boy ran away without letting her reply.

The chick went back to her apartment where here boyfriend waited for her at the dinner table. They both talked and she told him about the incident. The boyfriend found it pretty weird too.
Days went by, weeks went by, a few years went by. She got married to her boyfriend. But she never saw the kid again until.

One day, she heard a knock on her door. No one was home. Hesitatingly, she opened it to see a rugged looking youngster. The young man told her that he was the same boy she sang to, at the alley that night. He never had been able to forget her So he requested her to play that song again. She did that.

After a little chit chat, she asked him if he remembered why he was crying that night. He said he clearly remembered the reason. “There was a kid in my class who used to bully me continuosly. I,often used to cry in that alley, so that my parents wouldn't know about it. That day you made me happy with your beautiful music and inspired me to be brave, and I thank you for that.”He said as he got up and to take her leave.

As he stood at the doorway to to bid goodbye
The chick asked him “ What happened to the kid that bullied you?”

“Oh I killed him, the day after you cheered me up. Today is the day I got out the Juvenile center, completimg my scentence. I made a vow that day that to meet the person who inspired me, as soon as I got out of there. Thanks again, you are an inspiration” The guy replied and went off.

THE END

MORAL : Never stop for a creepy cry baby in a dark alley. Its scary as f**k

I hope these notes will help you

Let's get started

▷ Linear Regression : Motivation

In the case of classification we were provided with n points.

$$S_n = \{ x^{(t)}, y^{(t)} : t = 1, 2, \ldots n \}$$ was our training set.

$x^{(t)}$: was the t'th feature vector

$y^{(t)}$: was the label of t'th example

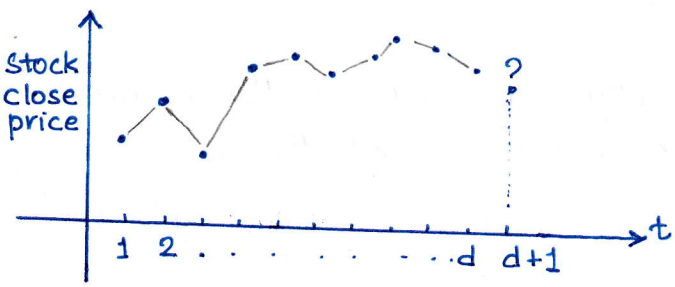
For some reason Prof. B uses t instead of i

The goal of the classifier was to find a mapping denoted by $h()$ that took as input the feature vector x and gave the output, the correct label y

Generally feature vectors $x^{(t)} \in \mathbb{R}^d$, ie feature vectors were d dimensional and each element of $x^{(t)}$ was some real value $\mathbb{R}$

The label $y^{(t)} \in \{+1, -1\}$ for the example we saw till now.

Lets say we have a problem of classification for a stock. We want our model to decide whether we should buy $+1$ or sell -1 the stock today. For our feature vector, we have the closing price of the previous d days. we choose the d as we wish



we may frame this problem as above but we want to know more than just whether to buy or sell.

We want to somehow predict how up or down the stock will go say 1 week or 2 weeks or 1 month Later

Framing this problem as buy/sell problem is an over simplification of the problem. And this is where regression comes in.

The setup of regression is very similiar. We are still doing supervised learning. Ie we will be provided with properly labelled feature vectors

S_n will still be $\{ (x^{(t)}, y^{(t)}) : t = 1, 2, 3 \ldots n \}$

But the labels $y^{(t)}$ will now be real values ie $y^{(t)} \in \mathbb{R}$

And we plan to predict unknown $y \in \mathbb{R}$ for the test data set

$y^{(t)}$ can have different meaning according to the problem

→ Stock prediction : Price of stock tommorow
→ Weather prediction : Temperature tommorow
→ Housing prediction : Price of house based of different features

Now we move on to discuss linear regression's setup

▷ Linear Regression : The setup

we will take the function $f(x, \theta, \theta_0)$ to predict y. Just a reminder

$x \equiv$ feature vector $= [x_1, x_2, x_3 \ldots\ldots x_d]^T \in \mathbb{R}^d$

$\theta =$ parameter $= [\theta_1, \theta_2, \theta_3 \ldots\ldots \theta_d]^T \in \mathbb{R}^d$

$\theta_0 \in \mathbb{R}$ called the intercept term

$f(x, \theta, \theta_0)$ will take in a d dimensional vector and in this case give out a real value $\mathbb{R}$, that will be the predicted value y

For linear regression, we take the regression function $f()$ of the form

$$f(x, \theta, \theta_0) = \sum_{i=1}^{d} \theta_i x_i + \theta_0$$

$$= \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d + \theta_0$$

In vector notation we can write the above equation as

$$\boxed{f(x, \theta, \theta_0) = \theta \cdot x + \theta_0}$$

Now if we look at the equation carefully, $\theta \cdot x + \theta_0$ is the equation of a hyperplane with normal vector θ and intercept θ_0.

This means we are using the hyper plane to predict the label $y^{(i)}$ for a given vector $x^{(i)}$

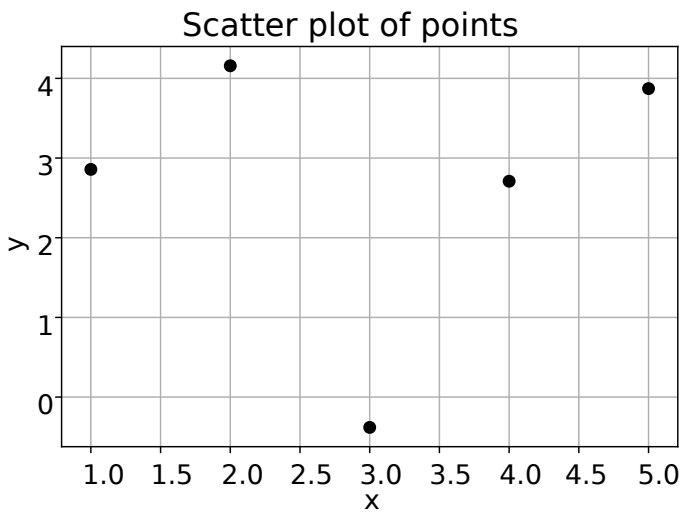
Now one might be concerned with the dependency of y on x vector that we are considering here. We are assuming that y depends linearly on the components of x vector. Why would such a simple assumption hold for different complex problems in the world.

The real power of linear regression holds because although we have this very neat form for the regression funtion $\theta \cdot x + \theta_0$. The feature vectors we choose can be transformed in a number of ways to form a neat Linear relation ship between inputs and output.

For example suppose I have the case where $x \in \mathbb{R}$ (simple 1d case) and I collect the following five data point $(x^{(i)}, y^{(i)})$

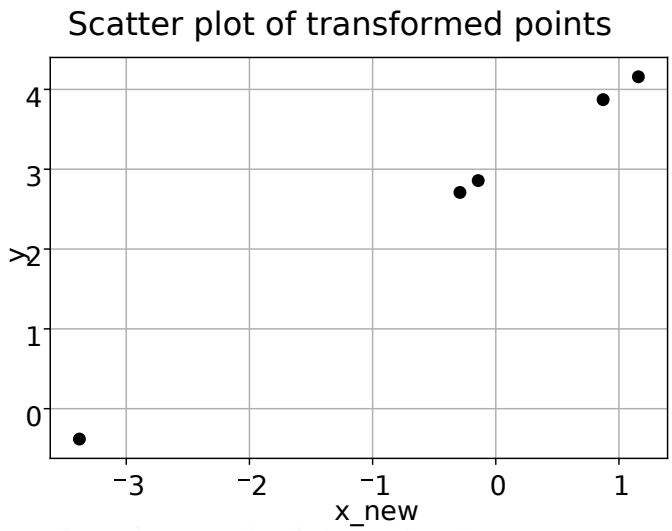
$x^{(i)}$	1	2	3	4	5
$y^{(i)}$	2.857	4.157	-0.38	2.708	3.871

Let's try to scatter plot the 5 points to see if there seems to be a linear relationship bw x and y points. The scatter plot looks like this.



We clearly see that it's a bad idea to fit a line through the data because no linear relationship can be seen.

But how about the next figure here. Is it a good idea to fit a line between the points drawn here?



Absolutely it is. It seems that a line perfectly passes through these 5 pairs drawn here

What if I told you that both the plots are plotting the same data (ie information) won't you be amazed? won't you die with suprise.

Please control your emotions.

The reason that is so, is because

x_new is just $\tan(x+2)$ where x was 1, 2, 3, 4, 5 as earlier

And how do I know this

It's because I made the dataset you stupid!!!

The crux of this discussion is simple.

And we will discuss it in the next page because, the space on which this line is written is just big enough to contain this line but not big enough to contain our discussion. So Please turn over.

The crux is simple

x :	1	2	3	4	5	
x_new :	-0.143	1.157	-3.38	-0.291	0.871	$\equiv \tan(x+2)$
y :	2.857	4.157	-0.38	2.708	3.871	

Although it's a horrible idea to fit the line

$$y = \theta_1 x + \theta_0$$

But it's a great idea to fit

$$y = \theta_1 x_new + \theta_0$$

In fact I generated y using the equation

$$y = 1 * (\tan(x+2)) + 3$$

Now the underlying transformations that will give us a new linear relationship, may not be clear in cases that we study, but those can be investigated. That's what feature engineering is.

The point is, fitting a hyperplane $\theta \cdot x + \theta_0$ is not a bad idea if we have correctly transformed x's. Lets proceed.

The plan of this lecture is to understand 3 issues

1) <u>Objective</u> : How do we know the optimal θ and θ_0? The objective will help us quantify the extent of our mistakes. Minimising it will yield the optimal θ and θ_0

2) <u>Learning Algorithms</u> : We will utilize data to find θ and θ_0 as before. For this we will see two cases

→ Gradient Based

→ Closed Form Solution

Fortunately linear regression has an exact solution in order to find θ and θ_0 according to given data.

3) <u>Regularization</u> : This is the part that will help us the problem of overfitting vs underfitting. We will see how our method generalizes for the test data

With that being said let's begin our journey

Before starting, a simple note θ_0 will be zero from hereon. All the maths remain the same with θ_0 present. This is just for convinience.

▷ Objective : Empirical Risk

Lets first talk about how we will penalize a point. This will be done with the Loss function : Loss (z_i). Now we can take many kinds of loss But for our case of linear regression, we take the square loss

$$\text{Loss}(z^{(t)}) = \frac{z^{(t)2}}{2} \quad \text{where} \quad z^{(t)} = y^{(t)} - \theta \cdot x^{(t)}$$

First let's understand what this quantity : $y^{(t)} - \theta \cdot x^{(t)}$ is

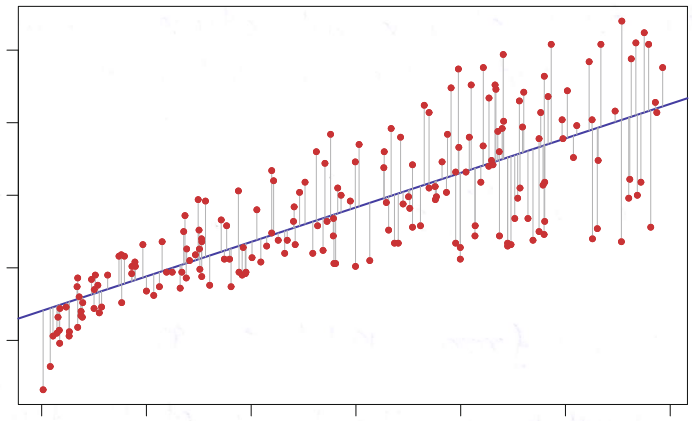
$y^{(t)}$ is the label $\in \mathbb{R}$ of the t'th example in our training data

$\theta \cdot x^{(t)}$ is the predicted value of label for the t'th example according to our function $f(\theta)$, ie regression function

So what we are interested in is

$$\frac{1}{2}(\text{Correct Label} - \text{Predicted label})^2$$

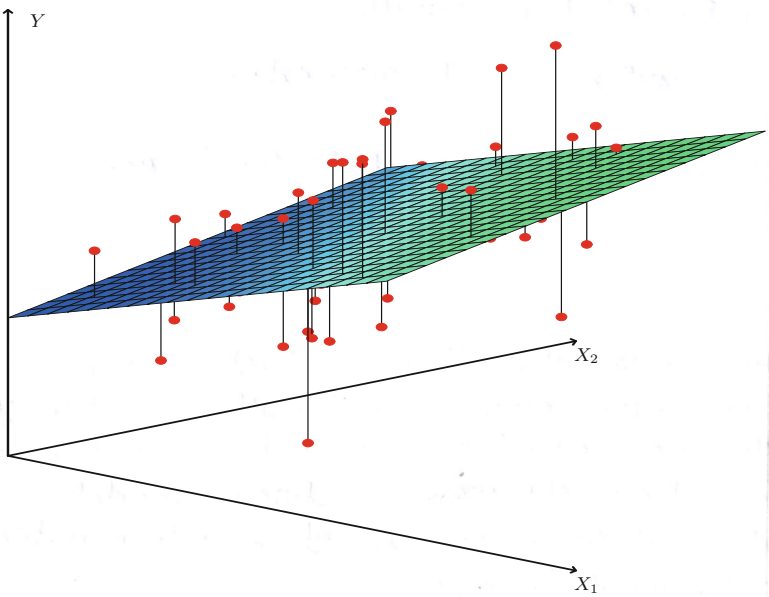
This looks visually as follows



The vertical lines you see from the plane $\theta \cdot x = 0$ to y are the values of $z^{(t)}$ for different points in dataset.

Here we first see a one dimensional case when $x \in \mathbb{R}$. $\theta \cdot x = 0$ is just a line here.

(I have copied this figure from ISLR so it's not passing throug origin)



Next we see the case when $x \in \mathbb{R}^2$ $\theta \cdot x = 0$ is now a plane through origin $y^{(t)} - \theta \cdot x^{(t)}$ is the vertical distance b/w the points and the plane

We may not be able to visualize what $y^{(t)} - \theta \cdot x^{(t)}$ means for more than 2 dimensions, but the math remains the same

Check out : An Introduction to Statistical Learning

the idea of squaring this quantity $y^{(t)} - \theta \cdot x^{(t)}$ is simple. When a point is far away from the hyperplane $\theta \cdot x = 0$ it will be penalized heavily due to square dependence. The points that are not that far away will be penalized less

With this in mind we can make our objective function as average of all these losses

$$\boxed{R_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} (y^{(t)} - \theta \cdot x^{(t)})^2 / 2}$$

This objective is given a special name. It's called the Empirical Risk
Our objective is to minimise the above objective.
Do you get the pun? Actually objective in English language means something like "aim", so what I did there was said that our aim was to minimise the objective function, which is a general term in Machine Learning literature. so I used the same word twice, but with different meanings at different places. This resulted in a mild chuckle from your side, which confirmed the pun was indeed wise. Let's proceed.

But we are also concerned with the issue of generalization. We want our model to perform well on the test data. we can define the empirical risk for the test data

$$R_{n'}(\theta) = \frac{1}{n'} \sum_{t=n+1}^{n+n'} (y^{(t)} - \theta \cdot x^{(t)})^2 / 2$$

where we assume test data is indexed from $n+1$ to $n+n'$

But we can't find this because we don't know the Labels of the test data set. But we want the quantity $R_{n'}(\theta)$ to be small.

Later we will see that the regularization will help us to generalise

But there are two type of mistakes that can occur. Lets see them

> Two mistakes

|1> STRUCTURAL MISTAKES

It may so happen that the linear function is not sufficient to tackle the problem which is to say that the mapping between input vectors x and output vectors y maybe highly non linear. So no matter how much we try, ie how much data we have, we wont be able to predict properly with our linear model. This leads us to choose more broader category or class of function instead of just linear functions, like non linear functions.

12> Estimation MISTAKES

It may also happen that the mapping between input vectors x and output vectors y is linear exactly but we do not have enough data to find that mapping correctly.

These two mistakes counter each other. Look at it this way, suppose we are dealing with the case when $x \in \mathbb{R}$. Instead of using the linear model $y = \theta_1 x + \theta_0$ we say that in order to estimate the y properly, we will take a 3rd order polynomial like $y = \theta_1 x^3 + \theta_2 x^2 + \theta_3 x + \theta_0$. This may decrease structural risk, but we will now need to estimate two additional parameters as compared to the linear case. So we need more data. If we don't have enough data, estimation mistake will be more prominent. So there's always this trade off between structural and estimation mistakes and a machine learning scientist chooses model accordingly.

However here we are all sure that we will use a linear model so lets proceed with that.

▷ Gradient Based Approach

What we want to minimise is our objective

$$R_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} \left(y^{(t)} - \theta \cdot x^{(t)} \right)^2 / 2$$

We already discussed in Lecture 4 that the minimization problem of the form

$$\underset{\theta}{\text{minimise}} \ \frac{1}{n} \sum J_i(\theta)$$

can be minimise using Stochastic Gradient Descent

So we will apply S.G.D

→ initialise $\theta = 0$ {any value will do}

→ Select i randomly from $\{1, 2, 3, \ldots\ldots n\}$ for kth iteration

→ update $\theta_{k+1} = \theta_k - \eta_k * \nabla_\theta \dfrac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2}$

Ie, we start with any initial value of θ. The using a random example from the train dataset we update θ using the specific gradient $\nabla_\theta \dfrac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2}$. Then we repeat till θ_{min} is found.

Let's see how the $\nabla_\theta \frac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2}$ is calculated

Remember for the general case in d dimensions

$$y^{(i)} \in \mathbb{R} \; ; \; \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}_{d\times 1} \in \mathbb{R}^d \; ; \; x^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_d^{(i)} \end{bmatrix}_{d\times 1} \; ; \; \nabla_\theta \equiv \begin{bmatrix} \frac{\partial}{\partial \theta_1}(\,) \\ \frac{\partial}{\partial \theta_2}(\,) \\ \vdots \\ \frac{\partial}{\partial \theta_d}(\,) \end{bmatrix}$$

So we have

$$y^{(i)} - \theta \cdot x^{(i)} = y^{(i)} - \theta_1 x_1^{(i)} - \theta_2 x_2^{(i)} \cdots\cdots - \theta_d x_d^{(i)}$$

$$(y^{(i)} - \theta \cdot x^{(i)})^2 = (y^{(i)} - \theta_1 x_1^{(i)} - \theta_2 x_2^{(i)} \cdots\cdots - \theta_d x_d^{(i)})^2$$

Suppose i want to find the partial derivative of above thingy wrt the jth coordinate θ_j

$$\frac{\partial}{\partial \theta_j}(y^{(i)} - \theta \cdot x^{(i)})^2 = 2(y^{(i)} - \theta_1 x_1^{(i)} - \theta_2 x_2^{(i)} \cdots\cdots - \theta_d x_d^{(i)}) * (-x_j^{(i)})$$

$$= 2(y^{(i)} - \theta \cdot x^{(i)})(-x_j^{(i)})$$

So $\nabla_\theta (y^{(i)} - \theta \cdot x^{(i)})^2$ is just all these partial derivatives stacked in a column vector

$$\nabla_\theta \frac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2} = (y^{(i)} - \theta \cdot x^{(i)}) \begin{bmatrix} -x_1^{(i)} \\ -x_2^{(i)} \\ \vdots \\ -x_d^{(i)} \end{bmatrix} = -\underbrace{(y^{(i)} - \theta \cdot x^{(i)})}_{\text{This is a real value}} \underbrace{x^{(i)}}_{\text{This is a } d\times 1 \text{ vector}}$$

So we have

$$\nabla_\theta \frac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2} = -(y^{(i)} - \theta \cdot x^{(i)})\, x^{(i)}$$

Lets plug this expression into our algorithm

SGD Algorithm

→ Initialise $\theta = 0$

→ Randomly select an integer from $\{1, 2, 3 \ldots\ldots n\}$

→ Update θ : For kth iteration

$$\theta_{k+1} = \theta_k + \eta_k (y^{(i)} - \theta \cdot x^{(i)}) * x^{(i)} \quad \text{——— (1)}$$

Now we know the whole deal about η_k : it depends on the kth iteration

We want $\sum_{k=1}^{\infty} \eta_k = \infty$ But we also want $\sum_{k=1}^{\infty} \eta_k^2 < \infty$

$\eta_k = \frac{1}{1+k}$ is a good choice satisfying above conditions

Also note in above equation (1)

> when $y^{(i)} - \theta \cdot x^{(i)}$ is large, ie the point deviates far from our plane (hyperplane) and its contribution to θ update is high

> Also unlike perceptron, each iteration results in update for θ

> Lastly if $y^{(i)} > \theta \cdot x^{(i)}$ ie predicted value is smaller than the actual Label then θ_{k+1} is given a +ive contribution towards $x^{(t)}$ ie it increased θ, so it in some way wants to bridge the gap between y and $\theta \cdot x$ by making $\theta \cdot x$ larger.

Now in the case of linear regression we are able to find a closed form solution to the problem

▷ Closed Form Solution

Fortunately Linear regression is one of the machine Learning topics that has a closed form solution. Lets see how we reach there.

We have our empirical risk

$$R_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} \frac{(y^{(t)} - \theta \cdot x^{(t)})^2}{2}$$

Now this is a function of $\theta \in \mathbb{R}^d$ ie its a d dimensional function that gives as output a real value in $\mathbb{R}$

To find the minima, we will find $\nabla_\theta R_n(\theta)$ and put it to zero

That's the way calculus people do it.

In the previous discussions of SGD we already calculated

$$\nabla_\theta \frac{(y^{(t)} - \theta \cdot x^{(t)})^2}{2} = -(y^{(t)} - \theta \cdot x^{(i)})\, x^{(i)}$$

This will help us in calculating

$$\nabla_\theta R_n(\theta) = \nabla_\theta \frac{1}{n} \sum_{t=1}^{n} \frac{(y^{(t)} - \theta \cdot x^{(t)})^2}{2} = \frac{1}{n} \sum_{t=1}^{n} \nabla_\theta \frac{(y^{(t)} - \theta \cdot x^{(t)})^2}{2}$$

∇_θ goes inside the summation, & we have, using previous expression

$$\nabla_\theta R_n(\theta) = -\frac{1}{n} \sum_{t=1}^{n} (y^{(t)} - \theta \cdot x^{(t)})\, x^{(t)}$$

Now before we proceed, let me remind you one thing (or few things)

$y^{(t)} \in \mathbb{R}$ ie its a real valued label

$x^{(t)} \in \mathbb{R}^d$ its a d dimensional vector, looks like $\begin{bmatrix} x_1^{(t)} \\ x_2^{(t)} \\ \vdots \\ x_d^{(t)} \end{bmatrix}_{d\times 1}$

$\theta \in \mathbb{R}^d$ its a d dimensional vector

It looks like $\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}_{d\times 1}$

So $\theta \cdot x^{(t)}$ is the inner product of θ vector with $x^{(t)}$ therefore its a real value

Also $y^{(t)} - \theta \cdot x^{(t)}$ will be a real value

Now $R_n(\theta)$ is a real valued function $R_n(\theta) : \mathbb{R}^d \to \mathbb{R}$ so

$\nabla_\theta R_n(\theta)$ will be a d dimensional vector like $\begin{bmatrix} \frac{\partial}{\partial \theta_1} R_n(\theta) \\ \frac{\partial}{\partial \theta_2} R_n(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_d} R_n(\theta) \end{bmatrix}_{d\times 1}$

so when we will put $\nabla_\theta R_n(\theta) = 0$ to find the minimizer θ_{min} we are putting d equations in θ to be equal to zero ie we have to solve a system of d linear equations

Let's denote our solution by $\hat{\theta}$, ie an estimator found from our data we finally have

$$\nabla_\theta R_n(\theta) = -\frac{1}{n} \sum_{t=1}^{n} (y^{(t)} - \hat{\theta} \cdot x^{(t)})\, x^{(t)} = 0$$

Now whenever we have a lot of linear equation, we resort to our God : Linear Algebra

So we want a neat algebra form for our d equations rather that the clumsy expression that is

$$\underbrace{-\sum_{t=1}^{n} y^{(t)} x^{(t)}}_{\text{I}} + \underbrace{\sum_{t=1}^{n} \hat{\theta} \cdot x^{(t)} \, x^{(t)}}_{\text{II}} = 0$$

Look at the first part $y^{(t)} \in \mathbb{R}$; $x^{(t)} \in \mathbb{R}^d$ therefore $y^{(t)} x^{(t)} \in \mathbb{R}^d$

we will call

$\sum_{t=1}^{n} y^{(t)} x^{(t)}$ as the vector b

So

$$\boxed{b = \sum_{t=1}^{n} y^{(t)} x^{(t)} \quad \in \mathbb{R}^d}$$

Let's see the second part $\sum_{t=1}^{n} \hat{\theta} \cdot x^{(t)} \, x^{(t)}$

Instead of using this dot '.' notation, lets properly write how inner product is defined. So

$$\hat{\theta} \cdot x^{(t)} \equiv [\hat{\theta}]^T x^{(t)} \equiv \hat{\theta} \, [x^{(t)}]^T \equiv [x^{(t)}]^T \hat{\theta}$$

where $[\ ]^T$ means transpose of the vector inside

Now $\hat{\theta} \cdot x^{(t)} \, x^{(t)} \equiv x^{(t)} \, \hat{\theta} \cdot x^{(t)}$ because $\hat{\theta} \cdot x^{(t)}$ is just a real number

Also $x^{(t)} \, \hat{\theta} \cdot x^{(t)} \equiv x^{(t)} [x^{(t)}]^T \hat{\theta}$ using result from above

So $\sum_{t=1}^{n} \hat{\theta} \cdot x^{(t)} \, x^{(t)} \equiv \sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T \hat{\theta}$

Now nothing in $\hat{\theta}$ is indexed by t, it's constant for all t's, so we can take it out of summation

$$\sum_{t=1}^{n} \hat{\theta} \cdot x^{(t)} \, x^{(t)} \equiv \sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T \hat{\theta} \equiv \left(\sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T \right) \hat{\theta}$$

Now look

$x^{(t)}$ is $d \times 1$ vector
$[x^{(t)}]^T$ is $1 \times d$ vector

$\Big\rangle$ $x^{(t)} [x^{(t)}]^T$ is a $d \times d$ matrix

& so is $\sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T$

Lets call this matrix A

$$\boxed{A = \sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T}$$

And therefore we have the equation as

$$-\sum_{t=1}^{n} y^{(t)} x^{(t)} + \sum_{t=1}^{n} \hat{\theta} \cdot x^{(t)} x^{(t)} = 0$$

$$-b + A\hat{\theta} = 0$$

Ie to find the closed form solution we need to solve :-

$$A\hat{\theta} = b$$

where

$$A = \sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T \qquad b = \sum_{t=1}^{n} y^{(t)} x^{(t)}$$

And we know the solution

$$\boxed{\hat{\theta} = A^{-1} b}$$

Now when will this solution be found ? It's simple: when A is invertible
Now A is a dxd matrix, so linear Algebra tells us something about when this happens..

If you don't like trivial proofs to some common results now's your chance to skip to the end of Page 16

Here we go <u>start</u>

The proof will consist of several parts

Part 1.> I will show what the matrix $A = \sum_{t=1}^{n} x^{(t)}[x^{(t)}]^T$ means

For this lets first say we have $n=3$ examples with each vector containng $d=2$ features

So the matrix A looks like :

$$A = \sum_{t=1}^{3} x^{(t)} [x^{(t)}]^T = x^{(1)}[x^{(1)}]^T + x^{(2)}[x^{(2)}]^T + x^{(3)}[x^{(3)}]^T$$

$n = 3$

$d = 2$

$$A = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \end{bmatrix} + \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} \begin{bmatrix} x_1^{(2)} & x_2^{(2)} \end{bmatrix} + \begin{bmatrix} x_1^{(3)} \\ x_2^{(3)} \end{bmatrix} \begin{bmatrix} x_1^{(3)} & x_2^{(3)} \end{bmatrix}$$

$$= \begin{bmatrix} x_1^{(1)2} & x_1^{(1)} x_2^{(1)} \\ x_2^{(1)} x_1^{(1)} & x_2^{(1)2} \end{bmatrix} + \begin{bmatrix} x_1^{(2)2} & x_1^{(2)} x_2^{(2)} \\ x_2^{(2)} x_1^{(2)} & x_2^{(2)2} \end{bmatrix} + \begin{bmatrix} x_1^{(3)2} & x_1^{(3)} x_2^{(3)} \\ x_2^{(3)} x_1^{(3)} & x_2^{(3)2} \end{bmatrix}$$

$$= \begin{bmatrix} x_1^{(1)2} + x_1^{(2)2} + x_1^{(3)2} & x_1^{(1)} x_2^{(1)} + x_1^{(2)} x_2^{(2)} + x_1^{(3)} x_2^{(3)} \\ x_2^{(1)} x_1^{(1)} + x_2^{(2)} x_1^{(2)} + x_2^{(3)} x_1^{(3)} & x_2^{(1)2} + x_2^{(2)2} + x_2^{(3)2} \end{bmatrix}_{2\times 2}$$

Now if I make a matrix called X as follows

$$X = \begin{bmatrix} \text{——} & x^{(1)} & \text{——} \\ \text{——} & x^{(2)} & \text{——} \\ \text{——} & x^{(3)} & \text{——} \end{bmatrix}_{3\times 2}$$

ie X is all feature vectors stacked one over other after taking the transpose

$$X^T = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} \end{bmatrix}$$

Again reminder
$n = 3$
$d = 2$

Now what if I told you that

A is just $X^T X$

Let me show you how

$$X^TX = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} \end{bmatrix} \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ x_1^{(3)} & x_2^{(3)} \end{bmatrix}$$

You can check the elements of this matrix and they are exactly equal to $\sum_{t=1}^{3} x^{(t)} [x^{(t)}]^T$ earlier calculated

And this is true for any general matrix with n & d

$$X = \begin{bmatrix} \text{———} & x^{(1)} & \text{———} \\ \text{———} & x^{(2)} & \text{———} \\ & \vdots & \\ \text{———} & x^{(n)} & \text{———} \end{bmatrix}_{n \times d}$$

Then X^TX is $\sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T$

Part I proves $\sum_{t=1}^{n} x^{(t)} [x^{(t)}]^T = A = X^TX$

where X is matrix with feature transpose vectors one over the other

---

|Part 2> If we have a matrix M with m independent columns then $Mv = 0$ cannot be true for any non zero vector v

Suppose we say that $M = \begin{bmatrix} | & | & & | \\ col_1 & col_2 & \cdots & col_m \\ | & | & & | \end{bmatrix}$

and we claim these columns are independent.

Suppose we say $Mv = 0$ is true which means

$$\begin{bmatrix} | & | & & | \\ col_1 & col_2 & \cdots & col_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = v_1 * col_1 + v_2 * col_2 + \cdots + v_m * col_m = 0$$

for some non zero v

But that can't be true because the column's of M are independent

$v_1 * col_1 + v_2 * col_2 + \cdots + v_m * col_m = 0$ would mean

all v_i's are zero for $i = 1, 2, 3, \ldots, m$

There fore part 2 proves

For a matrix M with m independent columns, $Mv \neq 0$ for <u>any</u> non zero vector v

---

|Part 3⟩ A square matrix S is invertible iff $Sw \neq 0$ for any non zero vector w

Suppose that's not true, S is invertible and there exists a non zero vector w st: $Sw = 0$ then I can multiply both sides by S^{-1} as it exists

$$w = S^{-1}0 = 0$$

& thats a contradiction as we assumed w is non zero.

Now we are ready for our question

When does A^{-1} exists ?

|Part 1⟩ says $A = X^T X$

So to explore the inverse of A we need to explore the inverse of $X^T X$. Suppose the inverse of $X^T X$ exist

|Part 3⟩ will tell us $X^T X$ a $d \times d$ square martrix inverse only exist if there is no non zero vector w st

$$X^T X w = 0$$

which means A^{-1} exists if

$$X^T X w \neq 0 \quad \text{for any non zero vector } w$$

I can premultiply with w^T both sides

$$w^T X^T X w \neq w^T 0 \equiv 0$$

$$(AB)^T = B^T A^T$$

$$(Xw)^T Xw \neq 0$$

$$\| Xw \|^2 \neq 0$$

A vectors norm² can only be non zero when the vector is itself non zero because norm² is the sum of squares of vectors components

So if A^{-1} exists then

$$\|Xw\|^2 \neq 0 \quad \text{for any non zero vector } w$$

$$\Longrightarrow \quad Xw \neq 0 \quad \text{And now}$$

|Part 2> kicks in to say that if

$$Xw \neq 0 \quad \text{for any non zero vector } w \text{ then}$$

Columns of X must be independent

We are there, but not yet just a little bit more
Now there is a famous theorem in Algebra that for a matrix X

$$\text{Rank}(X) = \text{Rank}(X^T)$$

Rank is essentially the number of <u>independent</u> columns in a matrix
We already proved that X (above that we created) has all columns independent if A^{-1} exist So

X has d columns which are all independent

This means X^T also has the same rank d
also has d independent columns
& what is X^T, its : All featur vectors (columns) stacked side by side

$$X^T = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(n)} \\ | & | & & | \end{bmatrix}_{d \times n}$$

should have d independent columns for $A = X^T X$ to be invertible

and what does d independent columns do?
They can make up any d dimensional vector in $\mathbb{R}^d$
Ie

> A is invertible if feature vectors $x^{(1)}\ x^{(2)}, \ldots, x^{(n)}$ have d independent vectors OR equivalently
> They span the d dimensional space $\mathbb{R}^d$

Now there are two issues that are to be noted

1. we need our n feature vectors to span a d dimesional space. Usually n: the number of observations will be greater than d: the number of features for each case.

   But this is not always the case. Long Long ago I did a course called Fundamentals of Statistics on EdX. There my professor R. told us about how he studied the effect of different genes on cancer. Now there could be 100's of thousands of genes as features for the study. But the train cases n would not be that large.

   In those cases $n < d$. So we cant find A^{-1} because n vectors even if they all are independent can't make <u>any</u> vector in $\mathbb{R}^d$, which is to say that when $n < d$, n vectors can't span $\mathbb{R}^d$

   So in order to find A^{-1} we need $n \gg d$, and we hope that out of these n vectors we can find d independent vectors

2. The computation of $\hat{\theta} = A^{-1}b$ is of order $O(d^3)$ so even if we have say 1000 features for each vector we will have computational time 1000000000, and that's a lot

Lets now move on to the topic of regularization

In order to understand the need and role of regularization we will first need to undestand a very important topic that reccuringly occurs through our study. It is the :

▷ Bias - Variance Tradeoff

Let me tell you a little about what perspective I learned in the statistics course. We have data and we want to model it as a statistical problem. All our problems & questions will be answered if we know what is the distribution that produces our data. It may not be the case that the data comes from a distribution, but if we want to study our problem mathematically, we will have to assume that that's true.

So what is our data exactly. We have $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. So look at it this way. Each time we sample a data point we get (or realise) a pair $(x^{(t)}, y^{(t)})$ according to the parent distribution (which we do not know)

As $x^{(t)} \in \mathbb{R}^d$ & $y^{(t)} \in \mathbb{R}$, we could have said that each time we get (realise) $d+1$ real numbers from our distribution. But there is a reason we said that we realise the pair $(x^{(t)}, y^{(t)})$. That because we want to tackle a different problem. we want to predict y when we have a given x

The idea is this X is a random vector, which is just d random variable stacked in a column and Y is a random variable. We want to predict Y given that X has realized some value denoted by x

This is written as $Y \mid X = x$; '|' means given

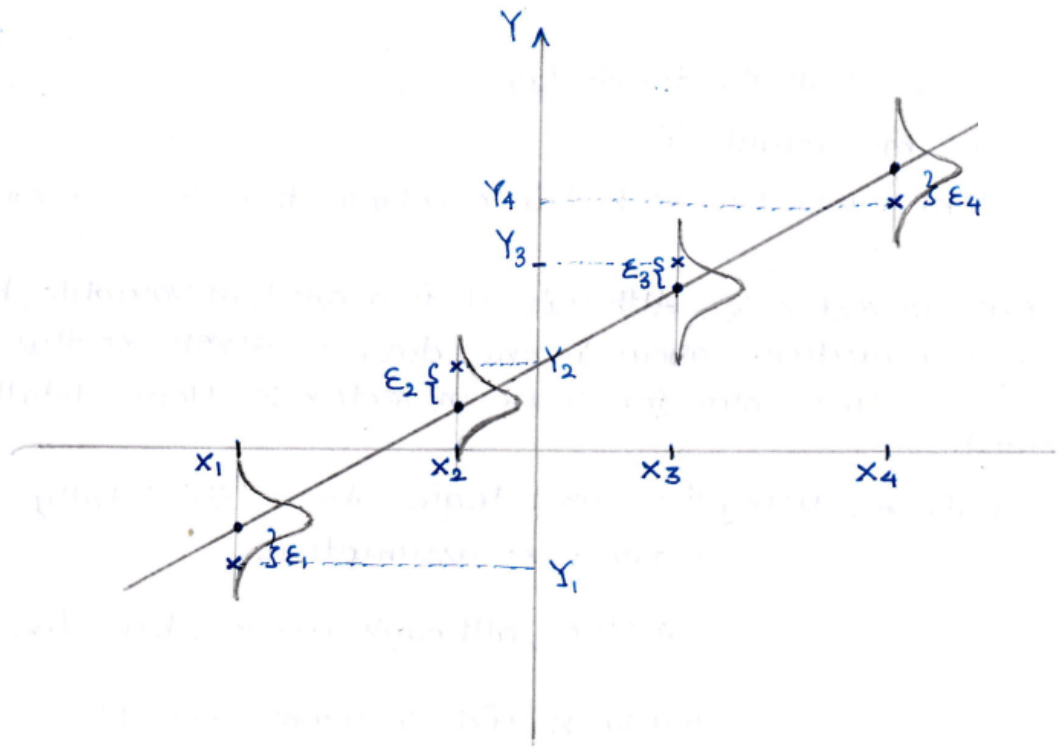
Note that even if we know the value of X vector we can <u>never</u> find the value of Y totally, ie their will always be a random component in Y that will be independent of X

In fact we can model $Y \mid X = x$ as follow

$$Y = \underbrace{f(x)}_{\text{This is the completely deterministic part once } x \text{ is fixed}} + \varepsilon$$

& this is the random part that comes from a distribution of its own Even if x is same, for every time we sample a value of Y, it will be different from previous values due to this random component ε

Now the most general assumption that statisticians like to make is $\varepsilon \sim N(0, \sigma^2)$ ie the ε random variable follows a normal distribution. This helps us proceed with great mathematical convinience. See how this looks graphically when we assume that $f(x)$ is a linear function of x



We will also proceed with the assumption that $\varepsilon \sim N(0, \sigma^2)$ just to see how Bias-Variance plays off

Now when we say that $Y = f(x) + \varepsilon$ {we are talking about Y given $X = x$ but I just use the notation Y} we believe that that's the truth. This is part of our modelling. But we will never be able to find this $f(x)$ unless we are all knowing GOD. What we can instead do is make an estimator of the function $f(x)$ which we will denote by $\hat{f}(x)$

What is an estimator you ask? Well to paraphrase Prof R. from the statistics course : 'Estimator is a function of data that can be computed once data is realised (ie we have numerical values of data)'

Look at it this way. For linear regression (assuming $\theta_0 = 0$) we assumed that the truth was $f(x) = \theta \cdot x = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_d x_d$

Unless we are God we will never know the true $\theta : \theta_1, \theta_2, \theta_3 \ldots \theta_d$

What we can do instead is estimate $f(x)$ by estimating all these θ's using our train dataset : $\{(x^{(t)}, y^{(t)}) ; t = 1, 2, 3 \ldots n\}$

$\hat{f}(x) = \hat{\theta} \cdot x$ where we found $\hat{\theta} = A^{-1} b$ earliear, $\hat{\theta}$ is an estimator & so $\hat{f}(x)$ is an estimator for the function $f(x)$ which is the truth

Now there are certain properties of interest that define how good or bad an estimator is.

1. Bias : $\hat{f}(x)$ is a function made from training data. Each time we have a different set of training data, each time $\hat{f}(x)$ will be a different function. But statisticians wants to see if on average, the $\hat{f}(x)$ is similiar to $f(x)$. Therefore they define

   Bias : $E[\hat{f}(x)] - f(x)$

   And this is called bias of the estimator of $f(x)$: $\hat{f}(x)$

   If the estimator is unbaised then the above quantity is 0. This would mean that although $\hat{f}(x)$ may not be equal to $f(x)$ for one train set. But on average over many many many (infinitely many) datasets, it will be $f(x)$ for every value of x

We must also note that $\hat{f}(x)$ is an estimator of $f(x)$. It is itself a random function. And the random function's random ness comes from the train data. Each time train data is realised, it will be different from previous data. & therefore each time $\hat{f}(x)$ will be different. Hence its a random function. And for a fixed value say $x = x_0$, $\underline{\hat{f}(x_0) \text{ will be a random variable}}$.

That is why we were able to take the expectation of $\hat{f}(x)$ above to calculate the bias

Next we will compute the variance of this random function $\hat{f}(x)$

Just a reminder, if Z is a random variable, then it's variance is defined as

$$Var(Z) = \mathbb{E}\left[(Z - \mathbb{E}[Z])^2\right] \quad \text{this simplifies to}$$

$$= \mathbb{E}[Z^2] - (\mathbb{E}[Z])^2$$

Both definitions are equivalent

2. Variance : similiarly a variance for $\hat{f}(x)$ is defined

$$\mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)]\right)^2\right] = \mathbb{E}[\hat{f}(x)^2] - (\mathbb{E}[\hat{f}(x)])^2$$

Now what this measures is the spread of a distribution. for a fixed value of $x = x_0$; $\hat{f}(x_0)$ will be a random variable. If that is so, then it will have a distribution and that distribution's width will be quantified by the above variance at $x = x_0$

Now lets look at the following quantity of interest

$\mathbb{E}[(y - \hat{f}(x))^2]$: I wanna tell you something. Firstly this is a function of x because $y = f(x) + \varepsilon$. The expectation here is over all the possible (infinite) training sets that will train $\hat{f}(x)$. Hence this quantity is a function (of x) that tells us how much prediction error, on average will we have

For linear regression $\hat{f}(x) = \hat{\theta} \cdot x$

But the following derivation holds for any general $\hat{f}(x)$

We want to see what our theory (of probability) says for this quantity so we take the analysis of

$$\mathbb{E}[(y - \hat{f}(x))^2]$$

Let's expand whats inside but before that remember $y = f(x) + \varepsilon$

$$\mathbb{E}[(f(x) - \hat{f}(x) + \varepsilon)^2]$$

$$\mathbb{E}[(f(x) - \hat{f}(x))^2 + \varepsilon^2 + 2\varepsilon(f(x) - \hat{f}(x))]$$

$$\mathbb{E}[f^2 + \hat{f}^2 - 2f\hat{f} + \varepsilon^2 + 2\varepsilon f - 2\varepsilon\hat{f}]$$

I have skipped '(x)' for a while

Now in order to simplify these expression I must tell you somethings

a) Note that there are only two random entities here, $\hat{f}(x)$ & ε

$\varepsilon \sim N(0, \sigma^2)$ so $\mathbb{E}[\varepsilon] = 0$; $Var[\varepsilon] = \mathbb{E}[\varepsilon^2] - \mathbb{E}[\varepsilon]^2 = \mathbb{E}[\varepsilon^2] = \sigma^2$

$f(x)$ is not a random entity, it's fixed. We are talking about analysing Y when $X = x$ is given to us so $f(x)$ is fixed

$$\mathbb{E}[f(x)] = f(x)\,\mathbb{E}[1] = f(x)$$

b. $\hat{f}(x)$ is random not because x is random, as stated above $X = x$ is known. The randomness comes from the training data that will change every time we sample it. that will change the <u>form</u> of $\hat{f}(x)$

c. **The estimator function fcap(x) is idependent of epsilon E , because the we think of Y being made of two parts. The first is f(x) that carries within it all the dependence Y has on X and the other is epsilon that is independent of the value f(x) has. This means**

So $\quad \mathbb{E}[\hat{f}\,\varepsilon] = 0$

Now we can find

$$\mathbb{E}[(y - \hat{f}(x))^2] = \mathbb{E}[f^2 + \hat{f}^2 - 2f\hat{f} + \varepsilon^2 + 2\varepsilon f - 2\varepsilon\hat{f}]$$

$$= \mathbb{E}[f^2] + \mathbb{E}[\hat{f}^2] - 2\,\mathbb{E}[f\hat{f}] + \mathbb{E}[\varepsilon^2] + 2\mathbb{E}[\varepsilon f] - 2\mathbb{E}[\varepsilon\hat{f}]$$

$f^2\,\mathbb{E}[1] = f^2$ (a.)

$\mathbb{E}[\hat{f}^2]$

$-2f\,\mathbb{E}[\hat{f}]$

σ^2 (a.)

$2f\,\mathbb{E}[\varepsilon] = 0$ (a.)

$-2\mathbb{E}[\varepsilon\hat{f}] = 0$ (b.)

$$= f^2 + \mathbb{E}[\hat{f}^2] - 2f\,\mathbb{E}[\hat{f}] + \sigma^2$$

Let's add & substract $(\mathbb{E}[\hat{f}])^2$

$$= f^2 + (\mathbb{E}[\hat{f}])^2 - 2f\,\mathbb{E}[\hat{f}] + \mathbb{E}[\hat{f}^2] - (\mathbb{E}[\hat{f}])^2 + \sigma^2$$

$$= (f - \mathbb{E}[\hat{f}])^2 + \mathbb{E}[\hat{f}^2] - (\mathbb{E}[\hat{f}])^2 + \sigma^2$$

So finally we have

$$\mathbb{E}[(y-\hat{f}(x))^2] = \underbrace{(f(x) - \mathbb{E}[\hat{f}(x)])^2}_{\text{This is Bias}^2} + \underbrace{\mathbb{E}[\hat{f}(x)^2] - (\mathbb{E}[\hat{f}(x)])^2}_{\text{This is variance}} + \sigma^2$$

of the estimator function $\hat{f}(x)$

$$= (f(x) - \mathbb{E}[\hat{f}(x)])^2 + \mathbb{E}\left[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2\right] + \sigma^2$$

What we did here is called Bias-Variance decomposition

This quantity $\mathbb{E}[(y-\hat{f}(x))^2]$ what does it denote ?

We know that it is

$$\mathbb{E}\left[(f(x) + \hat{f}(x))^2\right]$$

I want to remind you something, we are doing this for a fixed x. The source of randomness is not x. Allowe analysis is done when we have $X = x$ given. Thats why we were able to take $f(x)$ out of expectation $E[f(x)] = f(x)$. The expectation is over all random training sets for which $\hat{f}(x)$ will be trained. This is avery important point to understand And the next thing we will do from here can be alot confusing. So follow closely

WHAT IS EXPECTATION ?

I ask you? You don't know do you? That's what I expected.

Hello there. Scooby here. I would like to take a small detour here, in order to explain the pun above. Expectation is a mathematical propablistic concept, but in english language 'expectation' means a strong belief towards something happening. I asked you if you understood the mathematical concept, and said that I didn't 'expect' you to understand it. Hence the same word was used with two different meanys resulting in a slight chuckle from your side. End of explanation.

Expectation is a calculation you do on your random variable say Z

if Z is continuous then

$$\mathbb{E}[Z] = \int_{-\infty}^{\infty} z\, p(z)\, dz$$

where $p(z)$ is the probability density function of random variable Z.

& WE CANT DO THIS FOR ANY of THE ABOVE EXPECTATIONS

Calculation

Because if we can do these calculations, then that would mean that we have the joint distribution of (X,Y) & if that is the case then we can find answer to all our questions regarding Y or X or anything, maybe even life. using the distribution.

So we have to estimate all these expectation to observe their behaviour
And that is done by averaging over the "things" over which expectation is taken.

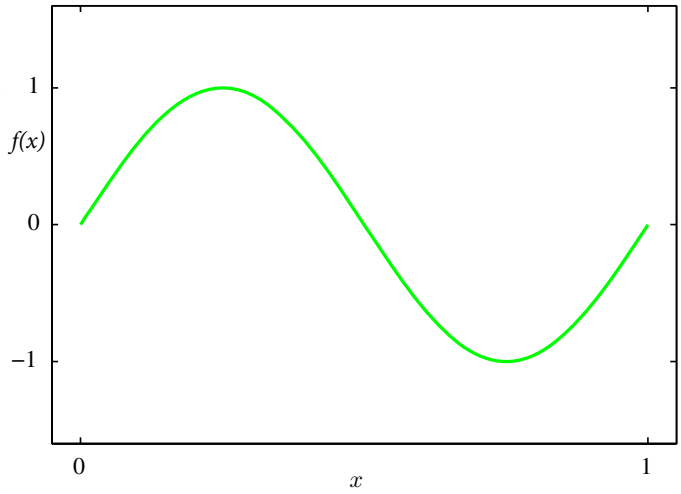
Instead of blabbering about it let me just show you. & in order to understand that you will have to understand the setup first

We wanted to understand the mystery of bias-variance tradeoff. So we turned to GOD for help. GOD being busy with this COVID crap, redirected us to a Bishop. Mr Bishop will help us indeed

1. Mr. Bishop told us that when we do machine Learning we do <u>not</u> have the truth ie $y = f(x)$ known. But what if we started with some known $f(x)$ & sample data from it to see how the bias variance trade off plays off
So he said lets take

$$y = f(x) + \varepsilon = \sin(2\pi x) + \varepsilon$$

and he said that we will sample our pair $(x,y) \in \mathbb{R}^2$ from this function



Because of the ε part, y will still be random for a given x
He drew the truth $f(x) = \sin(2\pi x)$ in the GREEN curve above

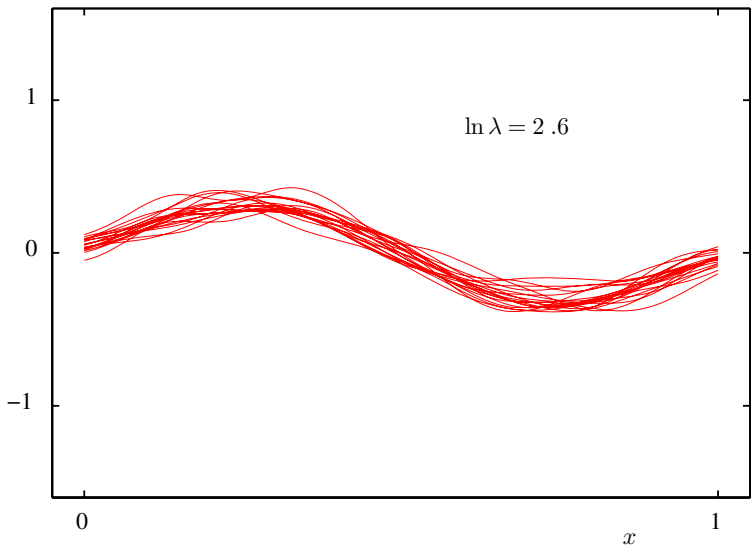
2. Now what we will do is train $\hat{f}(x)$. I am <u>not</u> saying that we are training a linear model like $\hat{f}(x) = \hat{\theta} \cdot x$, infact we already know that $\hat{f}(x)$ should resemble a sinosoid, if our model is good. Fortunately Mr Bishop knows a lot of other ways to train $\hat{f}(x)$

But not only that, Mr Bishop will sample 100 training datasets of size $n=25$, and for each one of them, he will train a $\hat{f}(x)$. So I denote

$\hat{f}_i(x) \equiv$ the prediction fuction trained by ith dataset

Sampled from distribution

$$(x,y) \equiv (x, \sin(2\pi x) + \varepsilon)$$



25 pairs x 100 datasets
And each $\hat{f}_i(x)$ is plotted in Red (thin)

3. Now Mr. Bishop tells us that the following quantity

$$E[\hat{f}(x)]$$

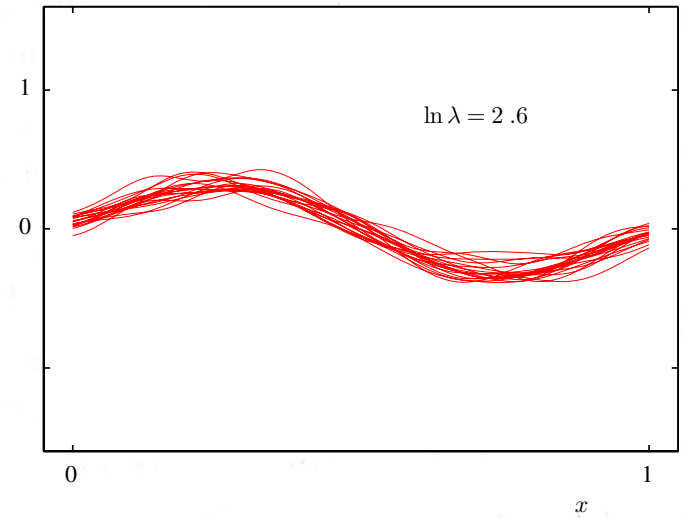
is the expectation of $\hat{f}(x)$ over all the datasets over which $\hat{f}(x)$ is trained

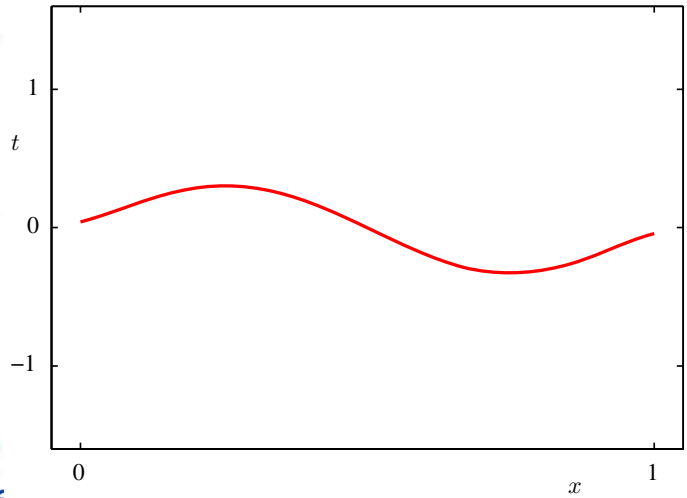
In order to correctly find it we will have to find the integral. Even though Mr. Bishop knows the integral because he knows almost everything. we will not be able to know it, because we only have data for our starting point. Usually we will have only 1 train data, but for our example Mr Bishop has already generated 100 datasets, & therefore we can estimate the following :-

$$E[\hat{f}(x)] = \frac{1}{100}\sum_{i=1}^{100}\hat{f}_i(x)$$

This may look complicated, but it's not We are just averaging the previous 100 trained functions $\hat{f}_i(x)$

This is what now gives us a new function denoted by <u>Bold red</u>, and this will also be a function of x.





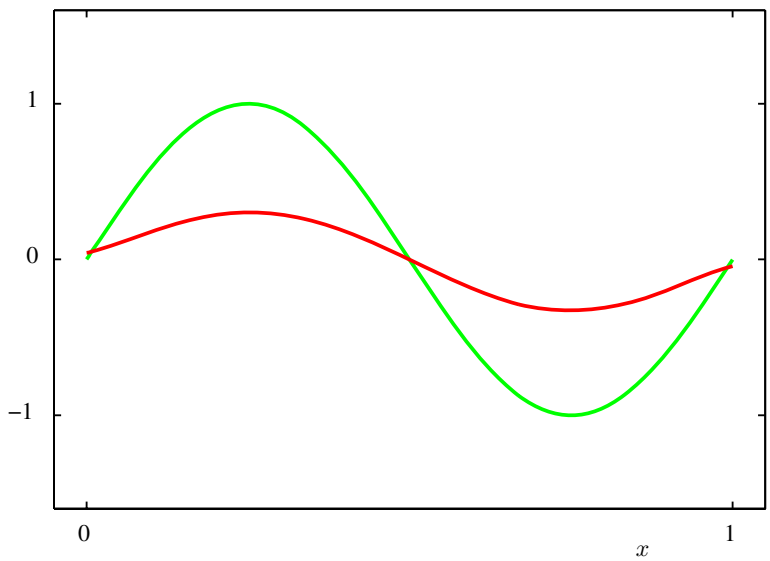
Aaaha!!!

Now we can find the Bias (or atleast its estimate)

$$(f(x) - E[\hat{f}(x)])^2 \equiv \text{Bias}^2$$

$$\underset{\text{(estimate)}}{\text{Bias}^2} = \left(f(x) - \frac{1}{100}\sum_{i=1}^{100}\hat{f}_i(x)\right)^2$$

$$= \left[\begin{matrix}\text{Green}\\ \text{curve}\\ @\ x\end{matrix} - \begin{matrix}\text{Bold Red}\\ \text{curve}\\ @\ x\end{matrix}\right]^2$$



Here Mr. Bishop placed the green curve & bold red curve side by side to contrast the bias at each point x

"But what about the variance? Mr Bishop" you asked gasping
Mr Bishop leaned on you. You could feel his breath on your neck. He leaned in.
Your heart starts throbbing "We will see that next". He said as he left the room
You sat on the floor helpless and weak. He won you lost....

4. Just like we estimated the bias from the 100 training sets we can find the estimate for variance. The variance was

$$\mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)]\right)^2\right]$$

we already know

$\mathbb{E}[\hat{f}(x)]$ can be estimated by $\frac{1}{100}\sum_{i=1}^{100}\hat{f}_i(x)$

So the variance can be estimated as

$$\text{Variance (estimate)} = \frac{1}{100}\sum_{j=1}^{100}\left[\hat{f}_j(x) - \frac{1}{100}\sum_{i=1}^{100}\hat{f}_i(x)\right]^2$$

So what we are doing is as follows for a given x, we are substracting the average of 100 $\hat{f}_i(x)$'s from each of the trained function $\hat{f}(x)$. Then we square this find the average.

$\hat{f}_i(x)$ were the thin 100 curves that were red drawn earlier.

$\frac{1}{100}\sum_{i=1}^{100}\hat{f}_i(x)$ was the bold red plot drawn earlier

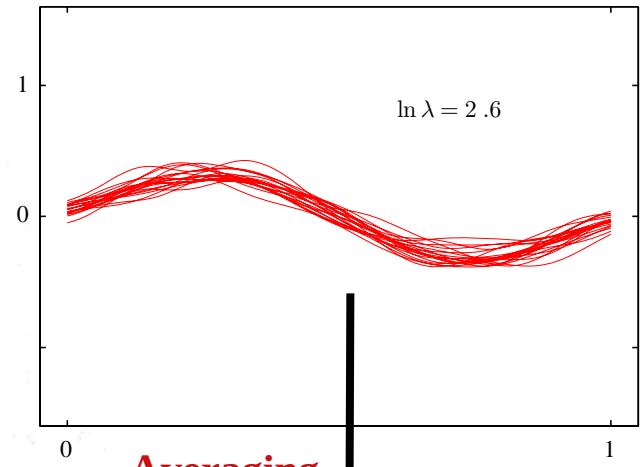
So here we imposed the thin red lines along with the bold red line together.

For each $\hat{f}_j(x)$ we can now visualize

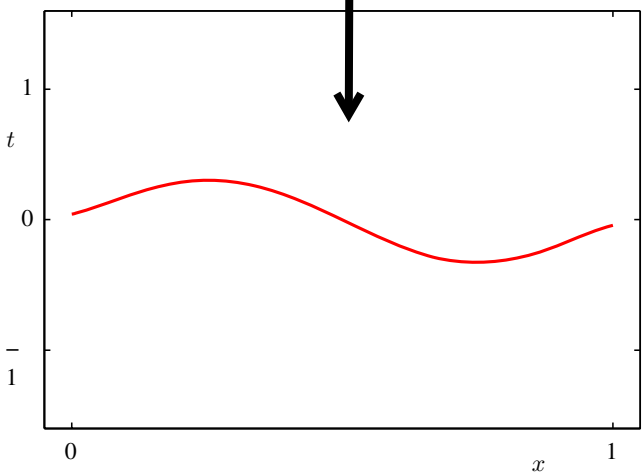
$$\left(\hat{f}_j(x) - \frac{1}{100}\sum_{i=1}^{100}\hat{f}_i(x)\right)^2$$

as

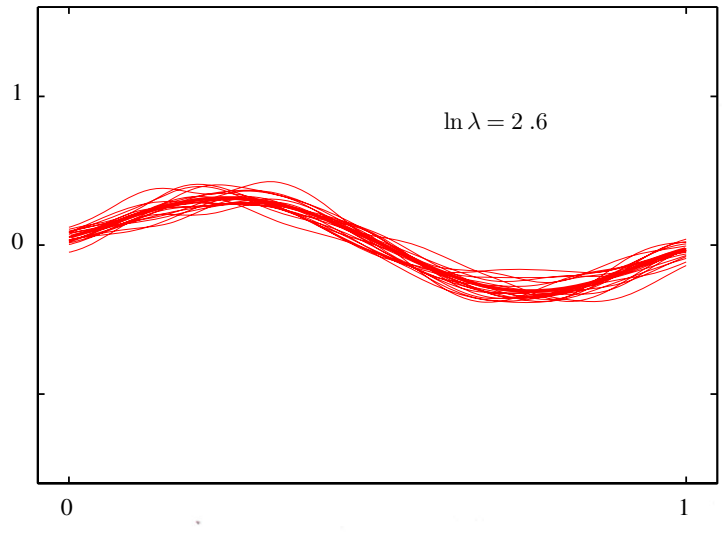
$$\left\{\begin{matrix}\text{jth thin} \\ \text{red curve} \\ @\ x\end{matrix} - \begin{matrix}\text{Bold Red} \\ \text{curve @} \\ x\end{matrix}\right\}^2$$



Averaging



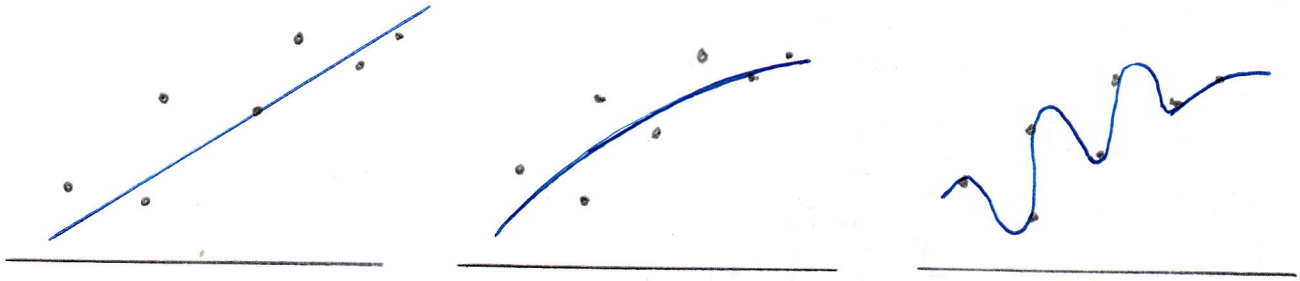
Bold red



Bold red overlapped over thin red

5. Now we ask Mr. Bishop about the bias-variance trade off. We still don't know what it means. Mr Bishop tells us

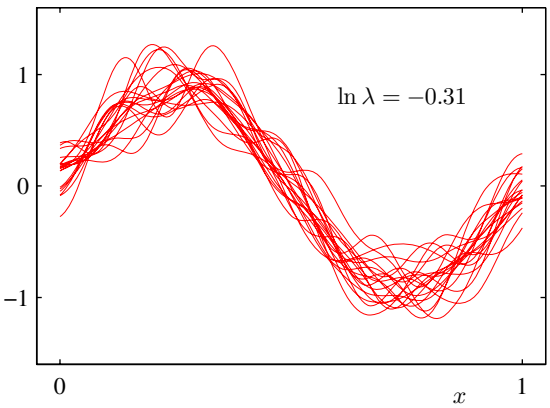
Bias-Variance Tradeoff is observed when the model complexity increases
Increasing model complexity means making $\hat{f}(x)$ more and more in accordnce with the TRAINING set

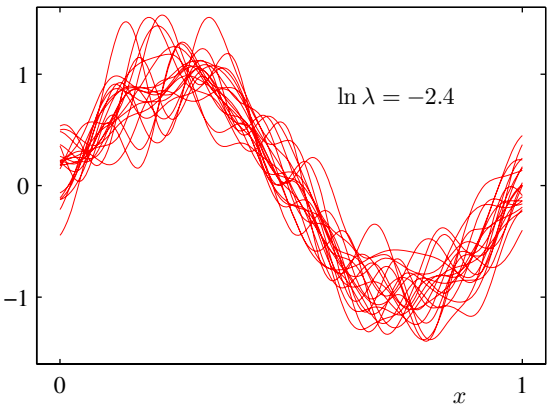


Here I show you two cases where the training set is same but the model complexity increases. One was draw earlier. So that makes total of three. As model complexity increases $\hat{f}(x)$ tries to fit more and more TRAINING points as closely as possible.

Mr. Bishop tells us that he can increase the model complexity more and more using the hyperparameter λ. How this is done is not of our concern here

But Mr Bishop trains $\hat{f}(x)$ on the previous 100 traing sets with increasing complexity 2 times. These are the graphs below





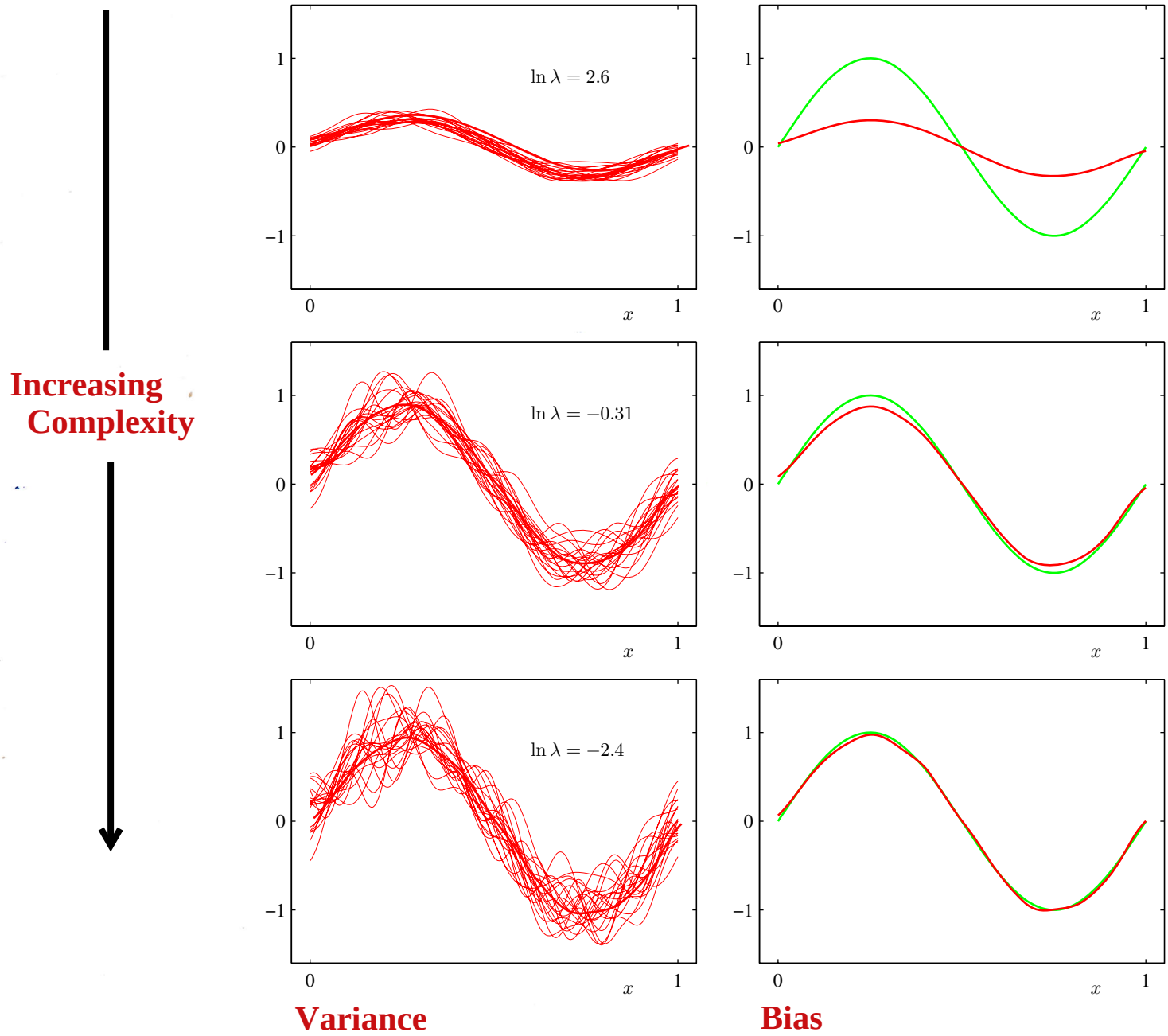
The model complexity increased by changing the paramater lambda

Although how this is done may differ from method to method, depending on the typr of objective we have

Here as we go down model complexity inceases, ie the curve tries to fit more and more to the train data

with this being done we can now see how the bais and variance will change when we increase the model complexity

That is done next



Now we have the whole picture. We have three cases with the same 100 train data sets. As we move down the complexity of the model increases, ie the trained $\hat{f}(x)$ tries to fit more and more to it's training data.

See how the green curve & bold red curve come closer when model complexity is increased. Therefore the Bias goes on to decrease more and more for a given x

However look at the thin red lines compared to the bold red line (which is their average). As model complexity increases, we see the thin red lines deviate from the bold red lines more and more, ie variance starts increasing.

And this is the Bias-Variance Tradeoff. As model complexity increases, the Bias decreases, but the Variance Increases and vice versa.

With that being said Mr Bishop took our leave. No one knows who he is and why did he know all this. On a completely different note, the above example was taken from:

Pattern Recognition and Machine Learning by **Christopher M. Bishop**

▷ Generalization and Regularization

Now in order to train the linear function $\hat{f}(x) = \theta \cdot x$ we will minimise the empirical risk

$$R_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} (y^{(t)} - \theta \cdot x^{(t)})^2$$

But this will be done over the training data.

Earlier we discussed that y could be thought of being made of two parts when $X = x$ is given

$$Y | X = x = f(x) + \varepsilon$$

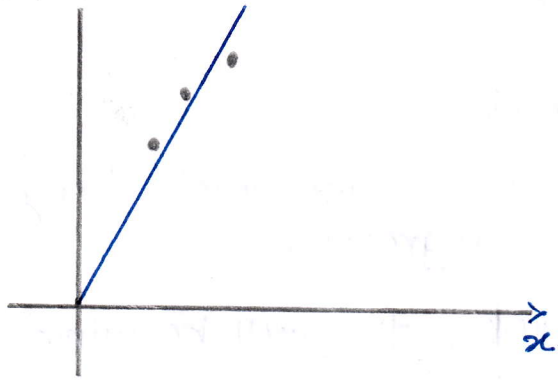
This ε was the source of our randomness in Y once $X = x$ is given. We with our methods can only hope to estimate $f(x)$ with the given train data.

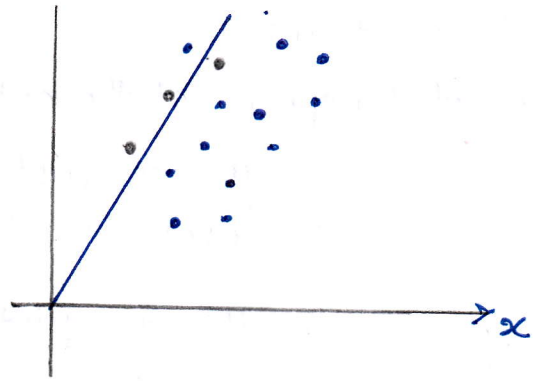
This ε is reffered to as noise. It's random

- Now for us, we have no way of knowing how much part of a given observ-ation is $f(x)$ part and how much is the noise ε part.

But if we have the noisy data, the algorithm would still try to fit a line (or hyperplane) through the points.

- Another issue that can occur is, what if we have limited data





Here we have $x \in \mathbb{R}$. The train set looks like in the left plot. These are three points denoted by '●'. The test set looks like '•' on the right plot.

If the linear model fits the best line with these three points, we clearly see that it will not be a good predictor for our test set.

It may seem that we can find more than 3 points for our train set, but this problem of limited data will be relevant when we deal with large d.

And we all know that large d can cause a lot of problems.

In order to tackle these issues we will do a slight modification in our objective function $J(\theta)$ to obtain a new better solution

Let's see

So our new objective will be defined as

$$J_{n,\lambda} = \frac{\lambda}{2}\|\theta\|^2 + R_n(\theta)$$

Ie

$$J_{n,\lambda} = \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{n}\sum_{t=1}^{n}\left(y^{(t)} - \theta\cdot x^{(t)}\right)^2$$

Lets look at the terms closely

> $R_n(\theta)$ ie the empirical loss is still, there, which means some part of us still want to minimise the square loss, and find a good fit according to train data.

> $\|\theta\|^2$ This term is the norm^2 of θ vector we want to minimise this also. Obviously in the extreme case if its very close to zero then its contribution will be very low

> λ : we are finiliar with the hyperparameter λ. It's the weighing factor that decides the contribution of $\frac{\|\theta\|^2}{2}$ term in contrast to the $R_n(\theta)$ term.

We can see what happens at the extreme cases

$\lambda \to 0$ (very small) $\Rightarrow$ $J_{n,\lambda}(\theta)$ is just $R_n(\theta)$ ie we just doing the previous linear regression

$\lambda \to \infty$ (very large) $\Rightarrow$ $J_{n,\lambda}(\theta)$ is just $\frac{\lambda}{2}\|\theta\|^2$. This will be minimised when θ vector $= 0$ ie all θ_i's $= 0$

But unlike classification where the loss term indicated the relative position & distance from decision boundary, & the $\|\theta\|^2$ term related to the width of the boundary (fat), we here don't have the intuition of what we are doing, when we are describing the objective function above. Lets try to gather some intuition Now.

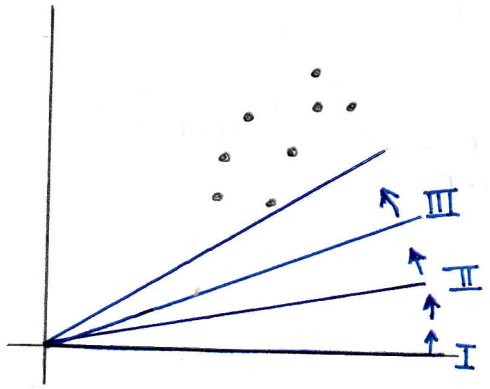
For the sake of visualization we will take

$x \in \mathbb{R}$, $\theta \in \mathbb{R}$, $\theta_0 = 0$

1 d case with line through origin

➤ Developing intuition for $J_{n,\lambda}(\theta)$

Let say we have $\theta \in \mathbb{R}$ and $x \in \mathbb{R}$. The function/estimator we are looking for is of form $\hat{f}(x) = \hat{\theta} x$ ie a line through origin, that would minimise $\frac{1}{n} \sum_i (y^{(t)} - \theta \cdot x^{(t)})^2$ in case of normal linear regression



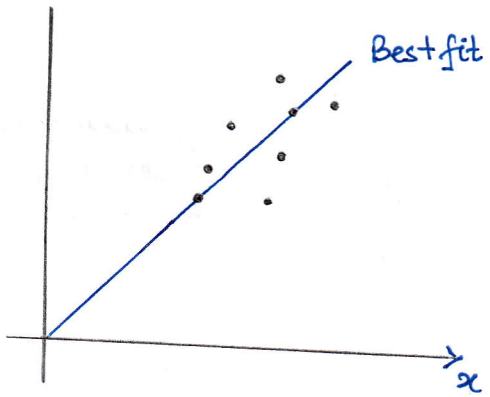
I look at it this way. It may not be how the algorithm proceed but we can look at it just for our convinience.

The algorithm starts from

$\theta = 0$

ie the equation is $y = \theta x = 0$

Its the horizontal line along x axis



Some step upgrades θ, according to data points. So the slope which is θ here starts increasing.

And calculates $R_n(\theta)$, it keeps on increasing θ & keeps on decreasing $R_n(\theta)$ from I to II to III & so on

Now there comes a point where $\hat{f}(x) = \hat{\theta} \cdot x$ fits the train data properly

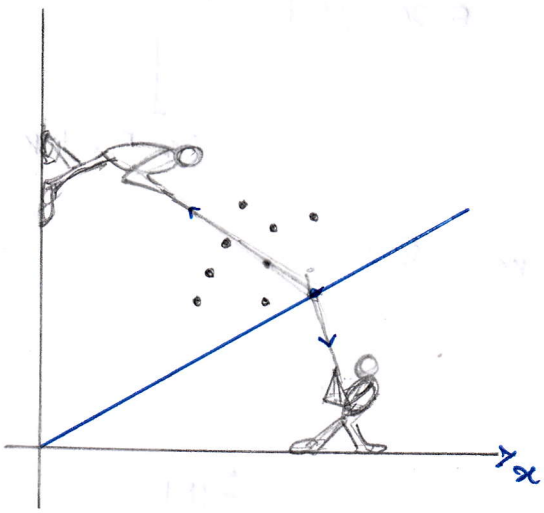
This is the best fit line that minimises

$$\frac{1}{n} \sum_{t=1}^{n} (y^{(t)} - \theta \cdot x^{(t)})^2$$

and the algorithm stops here.

Now if we see how our algorithm will proceed if we add this $\frac{\lambda}{2} \|\theta\|^2$ term to the objective.

The thing to note here is $\|\theta\|^2$ is actually $\|\text{slope}\|^2$ because θ is the slope of our line $\hat{f}(x) = \theta x$ here. {Remember $x \in R$}



So now two forces are at a tug-o-war

The points want the line to fit as best as possible to reduce $\frac{1}{n} \sum (y^{(t)} - \theta x^{(t)})^2$

The other force wants to reduce $\|\theta\|^2$ ie it wants to keep the slope at close to zero as possible

And the optimized solution comes to a draw between these two forces

> SGD is back baby!!!

Just like previously we can solve / optimize this new objective using Stochastic Gradient Descent.

→ Initialize $\theta = 0$

→ Randomly select i from $\{1, 2, 3, \ldots\ldots n\}$

→ Update θ on the iteration

$$\theta_{k+1} = \theta_k - \eta_k * \nabla_\theta \left[\frac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2} + \frac{\lambda}{2} \|\theta\|^2 \right]$$

So we will need $\nabla_\theta J_n$

$$\nabla_\theta \left[\frac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2} + \frac{\lambda}{2} \|\theta\|^2 \right]$$

we already know

$$\nabla_\theta \frac{(y^{(i)} - \theta \cdot x^{(i)})^2}{2} = -(y^{(i)} - \theta \cdot x^{(i)})\, x^{(i)}$$

See page 5.9

we need to find $\nabla_\theta \frac{\lambda}{2} \|\theta\|^2$

$$\nabla_\theta \frac{\lambda}{2} (\theta_1^2 + \theta_2^2 + \theta_3^2 + \cdots\cdots \theta_d^2)$$

$$\nabla_\theta \frac{\lambda}{2} \|\theta\|^2 = \frac{\lambda}{2} \begin{bmatrix} 2\theta_1 \\ 2\theta_2 \\ \vdots \\ 2\theta_d \end{bmatrix} = \lambda \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} = \lambda \theta$$

So finally we have

$$\theta_{k+1} = \theta_k + \eta_k * \left[-(y^{(i)} - \theta \cdot x^{(i)}) x^{(i)} + \lambda \theta \right]_{@\ \theta = \theta_k}$$

& this brings us to the end of this lecture. BYEE!!!

FIN