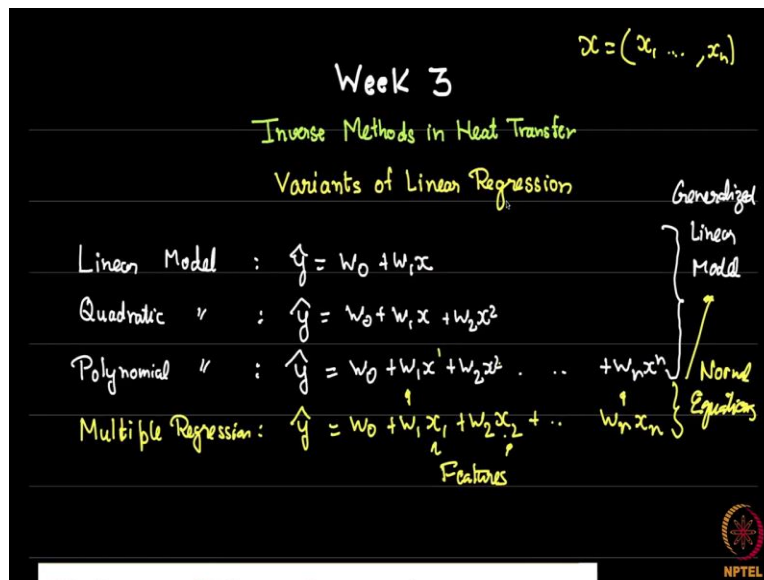


Inverse Methods in Heat Transfer
Prof: Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology, Madras

Lecture No 16
Variants on the Linear Model for inverse problems

(Refer Slide Time: 00:21)



Welcome back. In this video, we'll be looking at some variants of linear regression. we are still in week three. Recall that so, far, we had seen um simple linear model, which was

$$\hat{y} = w_0 + w_1x$$

We had also seen the quadratic model, which was

$$\hat{y} = w_0 + w_1x + w_2x^2$$

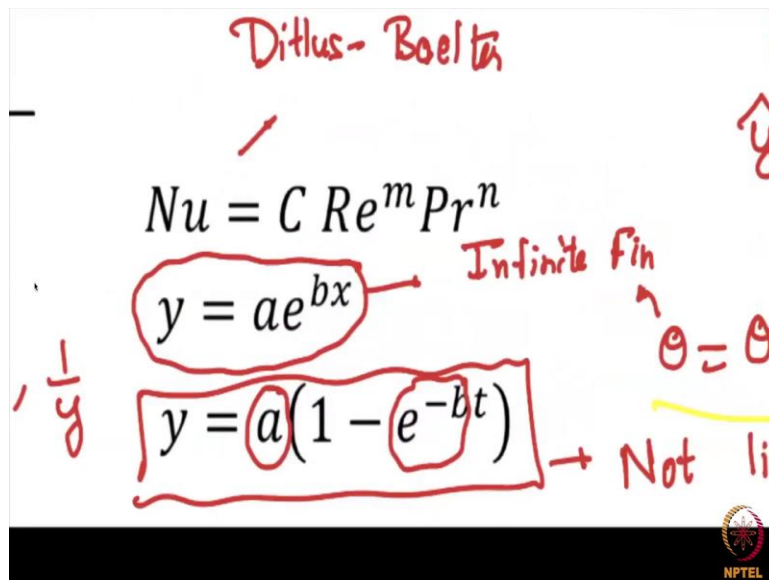
All these were models that we just had one single feature which was x. Then we saw that we could extend our analysis to polynomial models and I showed a few examples of this in the last coding video.

So, $w_0 + w_1x + w_2x^2$, you can add a cube term and go on w_nx^n . Now all these simply can be abstracted into a single idea of a linear model, the generalized linear model and the same generalized linear model can also be used for what is known as multiple regression or multi-linear regression. So, in this case, \hat{y} equal to $w_0 + w_1x_1 + w_2x_2$, where x_1, x_2 are different features and you can keep on going till w_nx_n . So, for example if x itself is made up of multiple Parts, x is a vector, x_1 through x_n .

So, you can use this multiple regression. Now we can also see that there is a one-to-one correspondence here. That is, you can think of x as simply x^1 and x_2 as x^2 and this is how we actually solve the linear model and all of these including this, can be handled with normal equations, which is on the last couple of parts. Now what we are going to do in this video is to look at variance.

Now this is already extremely useful as you would have seen. This is already very useful and it encompasses a large number of models, but we can extend this further and there are several variants of linear regression that exist in various forms and we are going to look at that in this video thank you.

(Refer Slide Time: 03:24)



So, we have been consistently minimizing the following objective residual or loss function, which is summation of,

$$J = R^2 = \sum_i (y_i - \hat{y}_i)^2$$

If you go back to the very first video that we did this in week one or week two, I believe I think it is week two you saw that the original form was something slightly different we had this scaled by $\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}$ and we will come back to this point but generally this is what we have been looking at.

So, this is what we call least square regression or LSR or sometimes, it is called ordinarily Square. you will see all these names within the literature. But we have other variants. So, we

have other variants. each of them has their own uses. So, for example we can change J itself. So, one change is so, notice this is called an L_2 loss because this power here is 2. We can also do this which is just summation of $y_i - \hat{y}_i$.

Now why should we add squares we can also add absolute values this of course is not differentiable. So, that is we can minimize the sum of the individual losses and we will see what kind of utilization it has, if time permits, we can try to see this within the machine learning setting within the last few weeks of this course if we have the opportunity I will come back to them. another choice is what is known as the L Infinity.

Notice this term L Infinity all this depends on a quantity or variations of what I call Norm the last time. So, you can look at L Infinity loss also called the Minimax loss. So, that is minimizing the maximum difference. So, what you do is instead of just minimizing some of the gaps, you see which one is the maximum different thing between your model and your prediction and try to minimize that. This of course works poorly in case you have outliers, but again if time permits, we will see some uses of this towards the end of this course.

The final one is the example that I show here, we should call the weighted least Square. So, you can notice that if I set,

$$\lambda_i = \frac{1}{\sigma_i^2}$$

you have a slightly different version; you do not have just the sum of least squares. but each of the squares is weighted by a different quantity λ_i . So, this is what leads to what is known as the weighted least squares approach and we will see that within this video.

We will not be seeing these two, but this we will see Within today's video. Now apart from this uh there are also a couple of other variants, that can be created or problems which can actually come from other functions. So, seemingly non-linear functions can be mapped to linear functions. Now a simple example of that is here. So, for example we had x we had x^2 and we simply called this x_1 and x_2 and. Now what look like a non-linear function looks like a linear function.

But of course, this was still linear in w_0, w_1, w_2 but let us say we have a non-linear function in the parameters itself. So, the parameters we want to determine suppose it is non-linear there,

it can be turned into a linear form through transformations. So, let me give you a trivial example and then we will see more important examples that directly arise especially in heat transfer. So, suppose somebody has a model saying $\hat{y} = w_0^2 + w_1^2 x$.

When you will say, I will just call w_0^2 as let us say C_1 and w_1^2 a C_2 and now it is linearized. So, this is a simple example of how a seemingly non-linear function can be turned into a linear function but this is of course a trivial example I am just renaming constants. you can do something a little bit cleverer and again you would have kind of done this probably at school too, but we will just see the context here. The important thing is that it should be linear in the parameters and not necessarily in the functional form.

So, here is an example $y = ax^b$. So, suppose we take this example $y = ax^b$. So, now notice it is kind of linear in a but this multiplies some x^b . So, it looks non-linear however if we transform this and take a logarithm on both sides and we have $\ln y = \ln a + b \ln x$. Now we do the renaming we call this variable as \hat{y} , \hat{y} is now \ln of the previous \hat{y} this I call w_0 this I call w_1 .

And this can simply be a feature or you can call it some other variable z . So, now you have $\hat{y} = w_0 + w_1 z$ and this is once again a linear problem. So, this is a simple example of a linear problem. let us take this one which can subject itself to the same kind of trick. So, this kind of example, if you remember from the first week this kind of occurs within an infinite fin. So, there you would have something like $\theta = \theta_b e^{-mx}$.

So, the parameters of the problem are these two θ_b and m these are the unknown parameters that we will typically solve in an inverse problem. So, if we take this example or let me take this example, $\theta = \theta_b e^{-mx}$, then I can take $\ln \theta = \ln \theta_b - mx$. So, now this we can call \hat{y} this we can call w_0 this we can call v and this is of course x . So, we once again have the form $\hat{y} = w_0 + w_1 x$ which is linear.

So, just for transformation by taking a logarithm on each side, we got there. Here is a very popular form. this is of course again coming from heat transfer. This is you will have in case you remember from heat transfer, this convection relationships or if you have multiple correlations, they come in the form of what is known as the Dittos Bolter equation for a pipe.

So, stuff like that when you come here. once again, you Now have noticed this case of two features so, now this is not just x this is x_1 and x_2 and your output is y .

So, if we take that case and we do Nusselt number equal to,

$$Nu = C Re^m Pr^n$$

and we wish to determine what m and n are using a lot of data points again we take a logarithm,

$$\ln Nu = \ln C + m \ln Re + n \ln Pr$$

This we can call x_1 , this we can call x_2 , this is w_1 , this is w_2 , this is w_0 , this is \hat{y} .

And just renaming these variables gives you $\hat{y} = w_0 + w_1x_1 + w_2x_2$. So, this kind of tricks, you can do multiple times. I will show you one further example you can come here. So, if you come to this expression here, this does not look like it is a simple linearizable example however with the transformation once again of setting one variable as $1/x$ and another variable as $1/y$ let me show you that trick here.

So, we had this case of $y = \frac{ax}{b+x}$. So, now take 1 over this. So, $\frac{1}{y} = \frac{b+x}{ax}$, this gives you, $\frac{1}{y} = \frac{b}{ax} + \frac{1}{a}$. Now once again you call this as \hat{y} take this $\frac{b}{a}$ call that as w_0 take this $\frac{1}{x}$ and call this as x_1 instead of calling this w_0 I will call this w_1 and $\frac{1}{a}$ is called w_0 and once again you get $\hat{y} = w_0 + w_1x_1$. So, this is once again linear.

So, all sorts of cases that you can work out, now unfortunately you take some cases like this one. this regardless of what trick you try will always be not linearizable. So, there is no simple rule that I can give you give you to tell which one will be linearizable and which one will be not but in this case, it happens to be not linearizable any trick you do will not turn it into a linear problem.

So, you have a here and you have e^{-b} here and this is always going to be non-linear. So, for this is what we will do next week, we have what are known as non-linear regression analysis. So, we will do non-linear regression analysis here. Now another case this is of great use is in cases like $y = ae^{-bt}$, this can be linearized because as you can see it is exactly of this form by $y = ae^{bx}$.

Now this example we will use within the exercises today, this week for unsteady conduction. So, the exercises for week three will have an unsteady conduction example, which you are supposed to convert to a linear problem and then perform a linear regression. So, that is the exercise for this week. So, what we saw Now in the first part of this video is that there are multiple equations or multiple forward models which can even though they look non-linear in the parameters can be turned to be linear in the parameter simply through a transformation.


(Refer Slide time: 16:19)

Example of weighted least squares

- Data set by Galton (1877).
- Measuring effect of parent's characteristics on child
- Compares pea diameter of parent plant with average diameter of up to 10 plants grown from seeds of the parent plant

Parent	Progeny	SD
0.21	0.1726	0.01988
0.2	0.1707	0.01938
0.19	0.1637	0.01896
0.18	0.164	0.02037
0.17	0.1613	0.01654
0.16	0.1617	0.01594
0.15	0.1598	0.01763

- Does the standard deviation make a difference in the fit?



Now we are going to look at this all-important case of weighted least square. So, now I want to show you this case of weighted least squares, remember what I had said that typically our loss function is,

$$J = \sum (y_i - \hat{y}_i)^2$$

This is of course the model this is reality. So, all we are doing is we are waiting each of the data points the same. that is the error in the first point is the same as, has equal importance as they error in the second point.

Now imagine you have a few friends and one person says something like this movie is good and even though their taste differs from yours you know that; they are at least reliable whereas one other person keeps on like varying their outputs. So, they are very unreliable another way to say it. So, you are not going to wake the information by the first friend the same as the information by the second friend.

Similarly, if you have multiple sensors in a slab. let us say multiple thermocouples and each of them like some of them are old, some of them are new and each of them is giving a reading you cannot wait them. So, these are different accuracies you cannot give equal weight to equal weightage to all of them. So, this kind of experiment which I am showing here is one such example, it is one of the oldest examples of a data set you can see this from 1877. This is by botanist whatever language we wish to use people then were multiple things, statisticians, botanists etcetera.

So, this is why a person called Galton and the idea was this he was measuring peas. So, mutter as they say in Hindi. So, he was measuring peas and he was trying to see the how the parent T diameter affects the child. So, progeny here means child. So, here is the diameter and this is not one single parent obviously he is measuring up to you know 10 children from a given parent etcetera and all these are statistical. there are a lot of data points, this kind of is the mean or an average of this data point.

So, what he saw of course was that there is some kind of correlation. So, you can see that as parent size decreases as it is with human parents. the child size also decreases, but notice that all these things are different. what are these? this is the standard deviation. all of you would be familiar with standard deviation from school. So, remember you would calculate,

$$\sigma = \sqrt{\frac{(y - \bar{y})^2}{n - 1}}$$

We will come to this definition again when we come to the statistics portions.

All you need to know is that not all data are equally reliable. of course, in this case, the variation is small. for example, this varies a little bit less at this size compared to something like this. So, if this is 16 then this is 20. So, 0.016, this is 0.020 but nonetheless these are at least different variations. So, the question is this, if you fit and you want to account for the fact that the errors are different, do we actually make a difference in the fit well the coefficients actually change once again we are going to try a linear model this is x.

Now this is y. So, we are going to say $\hat{y} = w_0 + w_1x$ but we want to see whether w_0 and w_1 will depend on whether the accuracy of the sensors is more or not in this case the accuracy of

the measured data is more or less. So, the way this makes a difference of course, is the difference that I had shown you earlier which was to say that J is no longer simply $(y_i - \hat{y}_i)^2$.

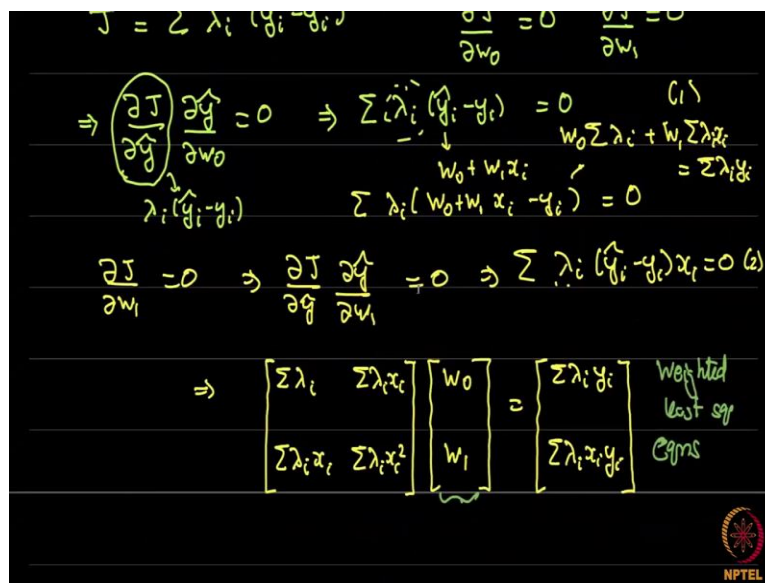
But I want to scale this by the inverse of this standard deviation or inverse of the variance σ_i^2 is called the variance as you might remember σ_i itself is the standard deviation. So, we can write this as,

$$J = \sum_i \lambda_i (y_i - \hat{y}_i)^2$$

I don't want to use w because many people use w here but w we have already used for the parameter. So, Lambda i are the weights which is to say if it is highly accurate then Sigma will be 0 then I want to give a lot of weight to it because. Now Lambda is going to be,

$$\lambda_i = \frac{1}{\sigma_i^2}$$

(Refer Slide Time: 21:47)



We will see a probabilistic kind of derivation to this idea, when we come to the probability portions of this course. Now how does this make a difference? So, remember, Now I am going to write,

$$J = \sum_i \lambda_i (y_i - \hat{y}_i)^2$$

Now I again I have to set the same thing Del J if I have a model with the w_0 and w_1 , I have to set

$$\frac{\partial J}{\partial w_0} = 0; \frac{\partial J}{\partial w_1} = 0$$

and this would say

$$\frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_0} = 0$$

So, the first equation so, if you notice this, this will basically give me,

$$\sum \lambda_i (\hat{y}_i - y_i) = 0$$

why because $\frac{\partial J}{\partial \hat{y}}$ is now going to be $\lambda_i (\hat{y}_i - y_i)$ almost no change from before except this λ_i

is extra. similarly, if I do $\frac{\partial J}{\partial w_1} = 0$ this will give me,

$$\frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_1} = 0$$

So, this will give me,

$$\sum \lambda_i (\hat{y}_i - y_i) x_i = 0$$

So, these are two equations and of course $\hat{y}_i = w_0 + w_1 x_i$. So, if you write these equations out, I am going to skip some steps, I would request you to go back to your notes for the case without the lambdas and check with this. But if you open this up, let me just open this up just for one case,

$$\sum \lambda_i (w_0 + w_1 x_i - y_i) = 0$$

So, the first equation is going to become $w_0 \sum \lambda_i + w_1 \sum \lambda_i x_i$ that comes from here and this term moves to the right-hand side and you will get equal to $\sum \lambda_i y_i$.

So, if I write this in matrix form, just like last time I have w_0, w_1 . So, you will see that the first term is multiplied by $\sum \lambda_i$ the second term by $\sum \lambda_i x_i$ and on the right hand side will be $\sum \lambda_i y_i$, go back to your notes and check you can check the normal equation mode also this would be $\sum 1$ this would be $\sum x_i$ and this would be $\sum y_i$, that is what happens when all of them are equally weighted.

So, really speaking all we have is the same set of equations except in every Sigma there is an extra Lambda setting which is not surprising. So, if you go back to your notes, you will see that, all you will notice is instead of these lambdas if you just replace this Lambda by 1 you will get the previous one and if you replace each of the sigma's with wherever there is a sigma

you put Sigma of that multiplied by λ_i you will recover the weighted least square equations. So, these are the weighted least square equations and you just need to solve these for the values of w_0 and w_1 obviously this is affected by the value of Lambda.

Now if the variation in Lambda is not much you can kind of take-out Sigma Lambda out more or less as an average in Lambda, in this case the variation is not much. So, you cannot expect too much of a big variation in the predictions of w_0 and w_1 . if the variation in Lambda is high then you can expect of course big variation in w_0 and w_1 .

So, I am just going to quickly show you a code for this and you are welcome to write this code using the normal equation approach. I will quickly derive the normal equation approach again without too much detail. But you can use that for there is a weighted least square example given in your assignment two. You can use some version of that by yourself. I will encourage you to write a code by yourself for this.

(Video Starts: 26:53)

So, what I am showing here is a code for the Gallatin data and I have written coding the inverse conduction problem which is I should change it to the weighted least squares problem. this is a copy-based issue., let us come here the forward model is still $w_0 + w_1$ and x notice that I have done the same trick I have cut and pasted whatever I had in my slide here, it is just a convenient way of doing it I encourage all of you to do this within MATLAB scripts.

Now the thing that has changed of course is that the weighted least square formula is now $\lambda_i(y_i - \hat{y}_i)^2$, where $\lambda_i = \frac{1}{\sigma_i^2}$. σ_i are the standard deviations given in the data set. I have now written this data out you can see that here 0.2, 1.2, 0.19 etcetera and as usual I had taken a transpose, I have also written the y data, just like before you can compare this code with the inverse conduction code also.

The extra thing here is the standard deviations, which basically measure how accurate or how much confidence we have in each one of these data points. So, the first one represents our confidence in the point to 1.1726 data points. the higher the confidence the higher the weight you wish to give. another way of saying is lower the sigma lower the error expectation the higher the weight you want it.

So, here I have this Lambda so, which I have defined as $\frac{1}{\sigma^2}$. Now a quick trick, I can do just to set the baseline is to declare one Sigma or zero the sigma zero is all once as you can see. And why am I doing this, this is to get the ordinary non-weighted least squares. So, I can try to get the normal weight Square least squares answer without accounting for this and I will take Sigma 0 here just to set a baseline and the formula remains the Same as I had shown you last time.

So, you have $\sum \lambda$ you can look up your node $\sum \lambda$, $\sum \lambda x$, $\sum \lambda x^2$ and then $\sum \lambda y$, $\sum \lambda xy$. If you remember the LHS it was which is shown here $\sum \lambda$, $\sum \lambda x$ it was a symmetric mix Matrix $\sum \lambda x$, $\sum \lambda x^2$ on the right-hand side we have $\sum \lambda y$ and $\sum \lambda xy$ and as useful w is equal to LHS by RHS or inverse of LHS by RHS.

So, I am going to run this code just. So, that you can see the w's, so notice the w at this point w, came out to 0.127 and 0.21. So, this is what happened without please remember without the correction due to least squares. So, please observe the physical plot here I have not drawn a parity plot in this case. So, the model prediction is the red line and the actual P data are on the dotted lines here.

Now of course what we need is not Sigma0 but we need Sigma. So, this is the actual case. Now when you run the actual case, you will see that the value changed, not by much because there was not huge variation here. Now suppose I was super confident in some data point I will just show you that case. So, here you see it has changed by a little bit, but it is hardly visual. suppose I am really confident about this point here and I make this really small. So, now notice how it is doing here, versus when I make this really small, it will try to match that a little bit more closely that is this point.

So, this is flipped. so, the 0.21, So, once again let me show the difference if I have 0.019 the Gap is 5 whereas if I am really confident about it and this error becomes really low, 10 times low, it will try to weight the least square line. So, that it tries to predict that correctly.

So, this is the advantage of a weighted least squares approach if we are relatively confident about some points versus others, we can actually weight them appropriately. We will come

back to this, when we consider also physics informed neural networks towards the end of the course. So, this here was a simple demonstration of a weighted least square code. this is of course explicitly programmed I will encourage you to program this in the normal equations approach also.

I will quickly write down the expression for the normal weighted least squares approach shortly.

(Video Ends: 32:14)

(Refer Slide Time: 32:15)

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & & \ddots & 0 \\ \dots & & & \lambda_m \end{bmatrix}$$

Λ is a diagonal matrix with λ 's on the diagonal

NORMAL Equations

$$X^T \Lambda X W = X^T \Lambda Y \quad \rightarrow \text{General weighted LS formula}$$

$\Lambda = I$

$$X^T X W = X^T Y \rightarrow \text{OLS}$$

So, just to summarize what we saw was that without weighting. So, that is what is known as the ordinary least squares. we saw that the numbers were something like w_0 equal to 0.127 and w_1 was something like 0.21, whereas with waiting which is the weighted least squares approach to get something like w_0 equal to 0.128 and w_1 equal to 0.205 or some somewhere. So, small change like I said the weight change can be large in case the differences in standard deviation are large.

Now the question that we wish to ask is, can we write the normal equations for this? I am just going to give you the final result and I will let you derive it and if maybe time permits, I will do it towards the end of the course if required. otherwise, I am just going to give you the final equations for this. So, define x the same way you know x_1, x_2 etcetera. The same design Matrix y etcetera also the same as before but define one extra matrix which I am going to call Capital Lambda which is a diagonal matrix.

This has Lambda 1, Lambda 2 up until Lambda m on the diagonals and it is 0 everywhere else. So, so Lambda is a diagonal matrix with lambdas on the diagonal.

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_m \end{bmatrix}$$

In this case the normal equation turns out to be the following. So, they turn out to be X^T remember what the previous equation was it was simply $X^T X W$ in this case it is

$$X^T \Lambda X W = X^T \Lambda Y$$

So, of course in the non-weighted case. This is the most General Weighted Least Square formula. the non-weighted case was $X^T X W = X^T Y$. this is for ordinary least square. you can derive from here to here by simply setting Lambda equal to the identity Matrix which simply means that all sensors are equally weighted. So, this is the expression for normal equations.

I encourage you to try to derive it in the same way that I derived it earlier it is possible but I did not want to spend too much time on this thing here, where we have other portions to move out to. So, what we saw within this video were variance of linear regression, we saw that some of them could be linearized, some functions could be linearized and some functions are deserve a weighted least square approach.

Both these are actually sitting within the assignment for this week. So, I hope you try the assignments and I will give you further insight into this procedure, thank you.