MATLAB Programming for Numerical Computation

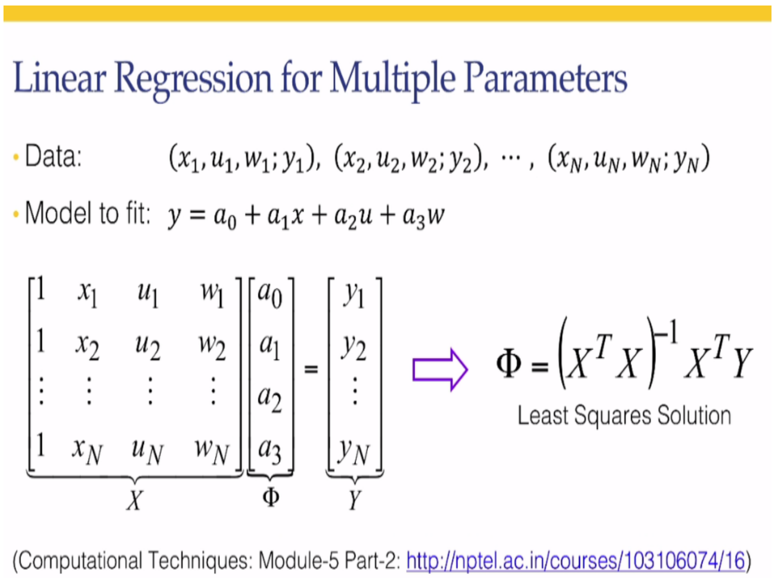
Dr. Niket Kaisare
Department of Chemical Engineering
Indian Institute of Technology, Madras

Module No. #06
Lecture No. #6.2

Regression and Interpolation – Linear least squares regression

Hello and welcome to MATLAB programming for numerical computations. We are in module 6. In this module we are covering regression and interpolation. In the first lecture of this module we introduced ourselves to regression and interpolation. In this lecture we are going to start with regression. Specifically, we are going to cover linear regression or linear least squares regression.

(Refer Slide Time: 00:40)



In the previous lecture we tried to fit data y equal to some function of the independent variable x. In that case the function that we were looking for was y = a0+a1x. Where, we were given data of x1, y1, x2, y2 and so on up to xn, yn. And the parameters that we wanted to fit were is a0 and a1. In that case we got our a0 and a1 as 2 unknowns which were expressed in terms of 2 equations.

2 equations in 2 unknowns so then solved in order to obtain a0 and a1. Now the matrix method is an even more powerful method of doing linear regression. And in matrix method we are not just restricted to 2 variables. We can extend it to multiple variables. In this case for example we have 4 parameters a0, a1, a2 and a3 okay. So when we have data in terms of x1, u1, and w1 which are the independent variables and y1 which is the dependent variable so this is the first data point.

Second data point is x2, u2, w2 and y2 and so on up to the nth data point xn, un and wn and yn. In this case we want to have a model fit of the form y=a0+a1x+a2u+a3w. In that case we form the matrix which is shown over here okay. The model that we see is 1 * a0 + x1 * a1+ u1 * a2 + w1 * a3.

So the first row multiplied by our phi vector is nothing but the right hand side of the equation evaluated at the first data point. Now let us look at the second row. Second row is nothing but 1 * a0 + x2 * a1+ u2 * a2 + w2 * a3 okay. So the second row is nothing but the a0+a1x+a2u+a3w evaluated at the second data point and so on and so forth okay.

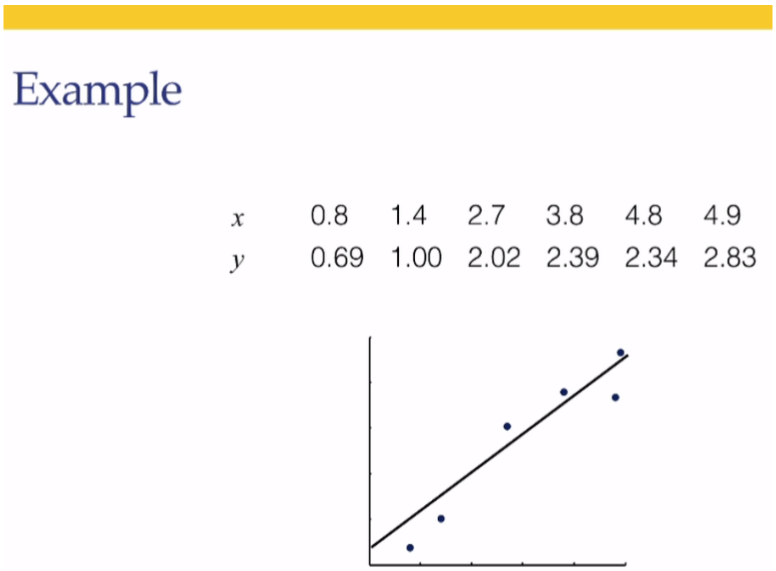
This we will have as n equations in 4 unknowns. For example we can have 15 data points to which we want to fit the 4 parameters. So in this case we might have n let us say if n was 15, we will have 15 rows and 4 columns only okay. So this does not have a unique solution okay. In that case we are interested in finding least square solution.

We are not going to cover of course the way of deriving the least square solution that was covered in computational techniques course module 5 part 2, the link for which is given over here. We had also shown in this computational techniques course that this least square solution for 2 variables x and y gives the exact same solution as the 1 we saw in lecture 6.1 the previous lecture okay.

So in this case what we are going to do is if we have just 2 variables, so that means y as a function of x if I want to say y=a0+a1x, the first column is going to be 1, 1, 1, 1, 1 and so on, the second column is going to be the data x1 x2 up to xn that will form our matrix x. Our matrix y is going to be nothing but y1, y2 up to yn and phi is going to be $(x` * x)^{-1} * x'*y$.

Let us say we wanted to fit a model of the form y=a1x+a2u+a3w, we did not have the a0 okay. Then this particular column is going to go out and this particular guy is also going to go out, so without this column and without a0 we will have x1 a1 u1 a2 w1 a3 = y1.So if we did not have a0 we would not have this column if we did not have u and w we would not have these 2 columns.

If we had a0+a1x+a2 u we did not have w we would not be having this column we would not be having a3 either okay. So this is very easy to think of how to form this matrix x, how to form this vector y okay. And that makes it a very powerful tool in order to do linear least squares.
(Refer Slide Time: 05:48)



What we are going to do now is to take the same example that we did in lecture 6.1 and solve it using the matrix method of multiple of linear least squares in multiple parameter. We will do it for 2 parameters for now and we will see that this method is easily extendable to multiple parameters. Let us go to MATLAB and try to do that okay.

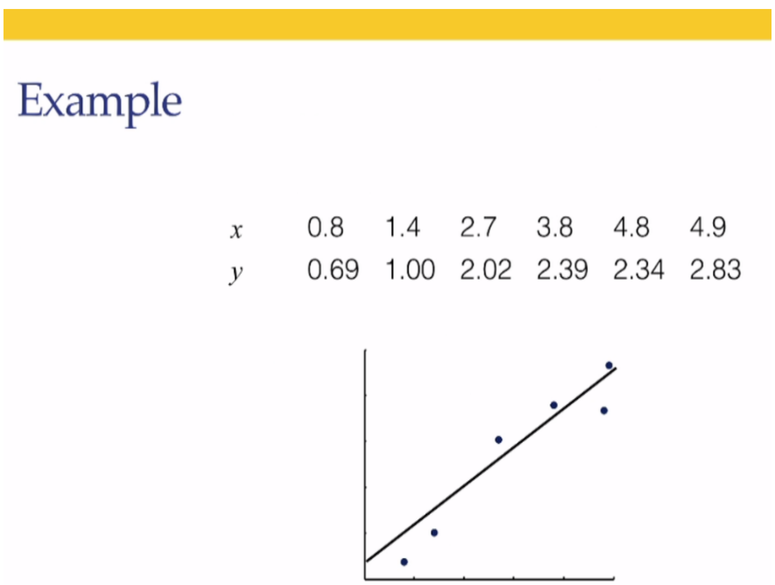
(Video Starts: 06:10) This was the code that we used in the previous lecture. Let us just save it, multiRegression, we will just call that okay. And we will retain the guys in the top, we will delete this and we will delete the interpolation part okay. Now let us calculate the linear regression and plot it. Our matrix x is going to be nothing but 1, 1, 1, 1 is going to be the first column.

We are going to have n elements in the first column. So we need to get this by using the command ones N number of rows and single column that is going to be first column. Our second column is nothing but our data x. So we will just put this as x over here and that is our capital, our capital y is nothing but was small y okay. This is was what our capital y is y1, y2 up to yn which is nothing but the same as the small y.

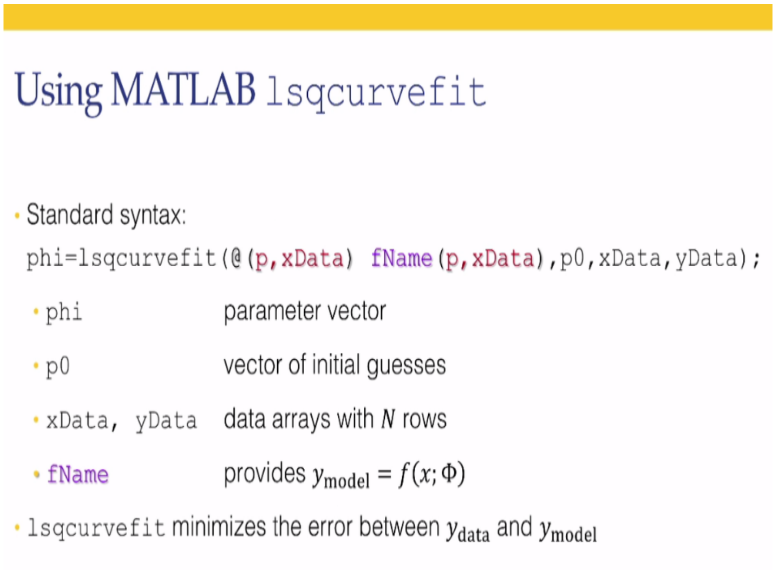
Our phi was nothing but inverse of let us go back and look at the equation x' * x, and this is very important and it is not x, x transposed but it is x' * x, (x` * x)$^{-1}$ * $x'$ sorry, multiplied by x` * y. We have cleared everything and now let us run this multiRegression. Okay and we see the exact same results as we had seen in lecture 6.1. We see those 6 data points and this is the best fit line. (Video Ends: 08:16)

Okay let us go back to power point okay. Okay, so in this lecture what we have done is we took the same example as lecture 6.1 but used a more powerful matrix based technique for doing linear regression. In the next lecture we are going to use this method and extend it to a practical problem for finding out parameters for a real case study. That case study will be taken from our 11th grade, example of finding the rate of reaction for a given reaction something that we did in chemistry in our eleventh and twelfth grade okay.

(Refer Slide Time: 08:53)



(Refer Slide Time: 08:56)



So this was the example that we used using our linear least squares. The other way of doing this is using MATLAB command lsqcurvefit. lsqcurvefit requires the optimization tool box okay. So let us go MATLAB and do this okay. (Video Starts: 09:13) So function let us say fval= myLinExample phi, xData, yData okay. What is my lsqcurvefit supposed to or the function supposed to return function is going to return just ymodel=f (x, phi).

It is not going to return the yData and it is not going to return the errors or things like that it is just going to return the function. So I do not even need yData over here. It is just xData. That it is going to return. So what this is going to return is the function form f (x) for all the xData. And what was our f(x)? Our f(x) was nothing but a0+a1x. So that is what this guy is going to return.

So fval is going to be phi (1) +phi (2) * xData. Now phi (2) is a scalar, phi 1 is a scalar. So our scalar phi (2) is going to multiply all of the data. And our phi (1) is also going to add to each and every element of the phi (2) * xData. So this is indeed going to be a0+a1*x. So let us save this as myLinExample. Let us go to multilinear regression okay.

And let us say phi = myLinExample okay. Oops! Sorry, not myLinExample, it should be lsqcurvefit at x, xData, myLinExample(x, xData), p0, xData and yData. So let us say the initial values of parameters were say 1 and 1, xData, yData okay .That is all I need and let us say using lsqcurvefit. Let us call this as phi (1) instead of phi.

And let me say hold on over here okay. And plot, what I want to plot is basically just this guy and I will plot this as. Let us say dotted lines of plot 0.5. 0.5 phi (1) + phi (2) multiplied by the same guy and dotted. Let us make this as a green color dotted line okay. Let us save this and its clear all, close all and let us run multi linear regression okay.

We have a problem over here okay. Our xData was not xData was just x. So let me change that yData was just y. So let me just change that also okay. See what had happened okay. I had just finally gone to our MATLAB where, I had given the syntax for the overall. You know this solution that we wanted to obtain and I had simply copied and pasted over that.

Now xData and yData is something that is not defined in this code. At all the xData is covered in the vector x, the yData is covered in the vector y. So it has to be the vec, the vectors that were previously defined we do not have to blindly give the same vector name okay. That is what it is actually complaining. It is complaining that this xData is not defined. So now let us run multi linear regression and hopefully this should work okay.

So it says the local minimum found and let us look at the result okay. So this is the red line that was there as before and on top of that red line there is that green line. I do not know, if you can see let me line with, let me just increases to 3.0. So, that you can see this. Yeah so now you can see the green line which is on green dotted line which is on top of that red dotted line may be I will just make this line a little bit thinner. Yeah so you can see the red and the green dotted okay.

So as you can see this that the 2 lines are exactly on top of each other that is the red line was, nothing but our matrix based method to find the linear regression. And the green line was our MATLAB lsqcurvefit based method to find the linear regression okay. I am sorry, I actually that is not what we did over here okay.

Let me edit this and now let me run okay. Yeah so that is again what we see over here. Let us convert this into instead of a red color, let us change it to a black color line or rather instead of green color will convert it to a black color line and mark 2 okay. So now we can see. So that is the black dotted line exactly falling on top of the red solid line.

The red solid line was for the straight line that was computed using our matrix method. And this one was for the straight line that was computed using the MATLAB based lsqcurvefit. Let us see the values of phi. phi is this let us see the values of phi1 which is also exactly the same. (Video Ends: 16:37) So what we have done now in this lecture is, to use MATLAB method and lsqcurvefit as well as write our own multi linear regression code using the matrix method in order to find the value of phi which best fits the model of the form y = a0+a1x and so on.

With that I come to the end of this lecture. In the next lecture we will take up an example of the rate of reaction. And we will show how to use, how to convert that rate of reaction problem into a linear regression problem and next we will use the matrix method in order to solve that problem okay. So I will see you in lecture 6.3 thank you bye.