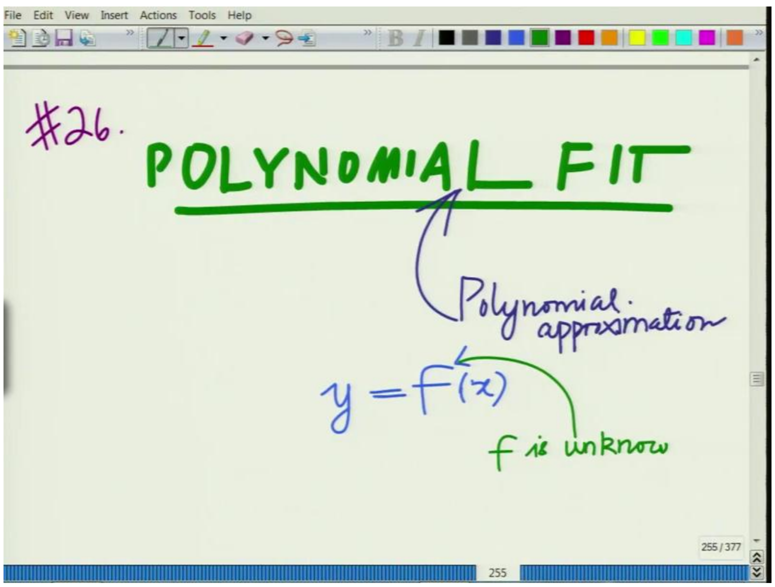
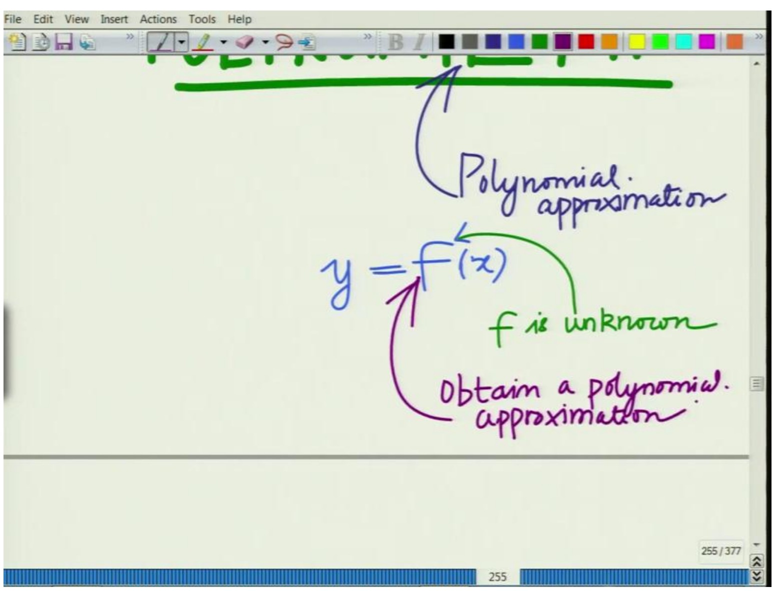
Applied Linear Algebra for Signal Processing, Data Analytics and Machine Learning
Professor Aditya K Jagannatham
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Lecture 26
Computation Mathematics Application: Polynomial Fitting

Hello, welcome to another module in this massive open online course. Let us look at another interesting application of least squares and that is in the context of polynomial fitting or polynomial approximation.

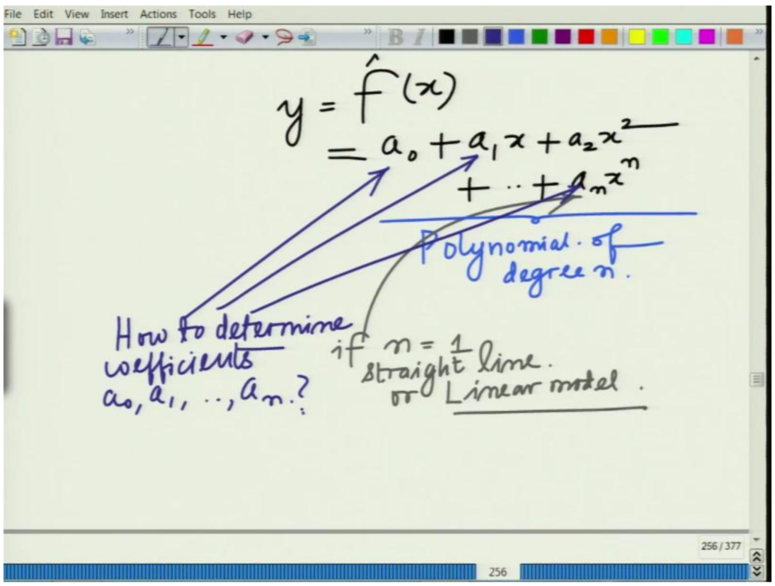
(Refer Slide Time: 0:26)





So, another very interesting application is to develop is in the polynomial in the context of a polynomial fit that is to given a set of data, develop a polynomial approximation for this functional polynomial to develop a polynomial approximation or you can think of this is essentially polynomial fitting. So, essentially you have y equal to let us say you have a functional relation that is unknown, so y equal to f of x, but f is unknown, let us say this function is unknown. f is unknown and for this function, we develop, obtain a polynomial approximation, obtain a polynomial. So, we would like to obtain a polynomial approximation for this function

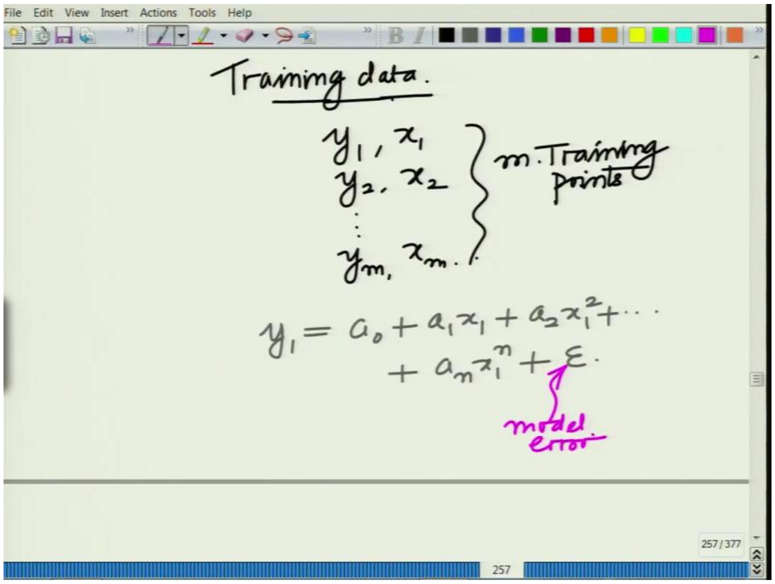
(Refer Slide Time: 2:14)



That is we want to approximate y as f hat of x, which is essentially a naught plus a1x plus a2x square plus so on an x raise to the power of n. So, this is your polynomial of degree n. So, this is the polynomial, nth order of polynomial or polynomial of degree n. So, how do we develop and of course, now, if n equals to 1, if the degree is 1, then it becomes a linear fit, which is what we have already seen in the linear regression.

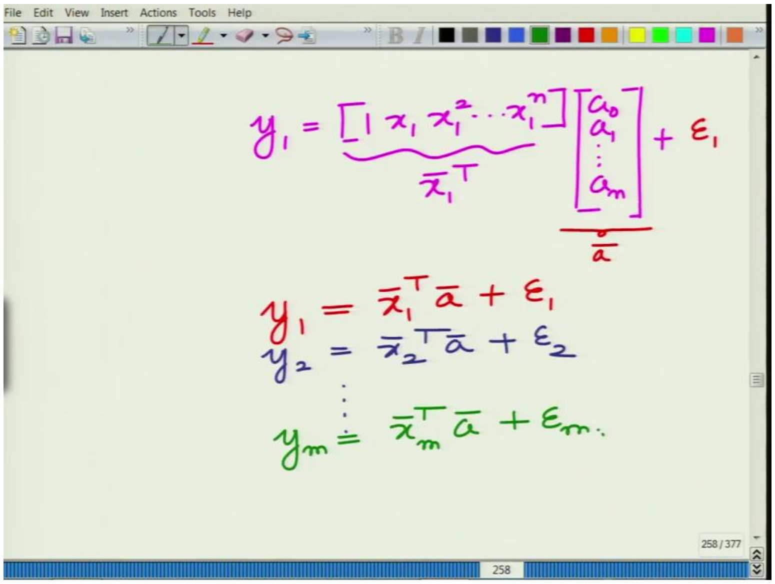
So, for n equal to 1, if n equal to 1, this becomes a straight line, it becomes a straight line or essentially a linear model. Now, let us see what we do is now, given this how do we approximate this. So, we have y1. So, how to find the coefficients? Now, the question again is how to determine the coefficients. How to determine the coefficients a naught, a1, so on up to an.

(Refer Slide Time: 4:40)



Start again, So to determine a naught, to determine a1 up to an, consider once again your training data or the available data, where you have y1 equal to and then you have again you have y1x1, y2 is essentially f of x2, you have y1 and you have x1, you have y2, you have x2, so on, you have ym, you have xm once again, you have the m, you have the m training points. And now, we want to model y, each y as y1, we want to model this as y1 equals a naught plus a1x1 plus a2x1 square, plus anx1 n plus epsilon where epsilon is the, this is the modeling error. So, this is your model error and this is the polynomial approximation, this is the polynomial fit.

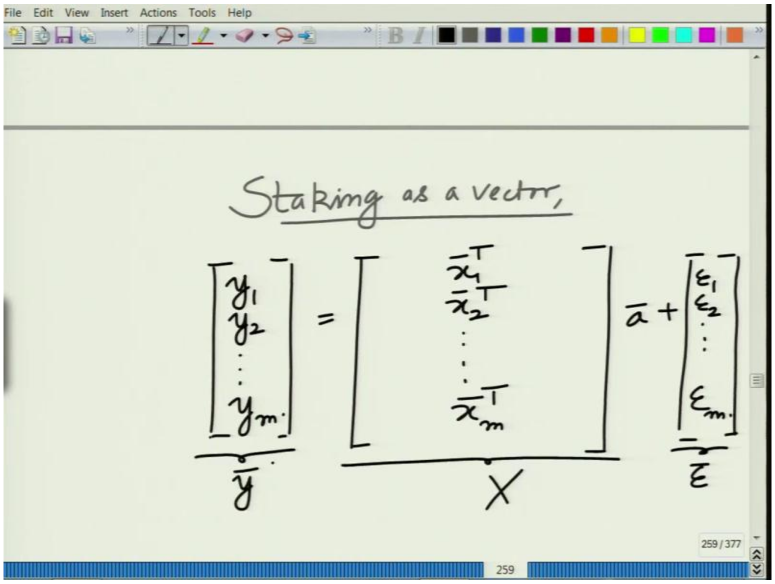
(Refer Slide Time: 6:35)

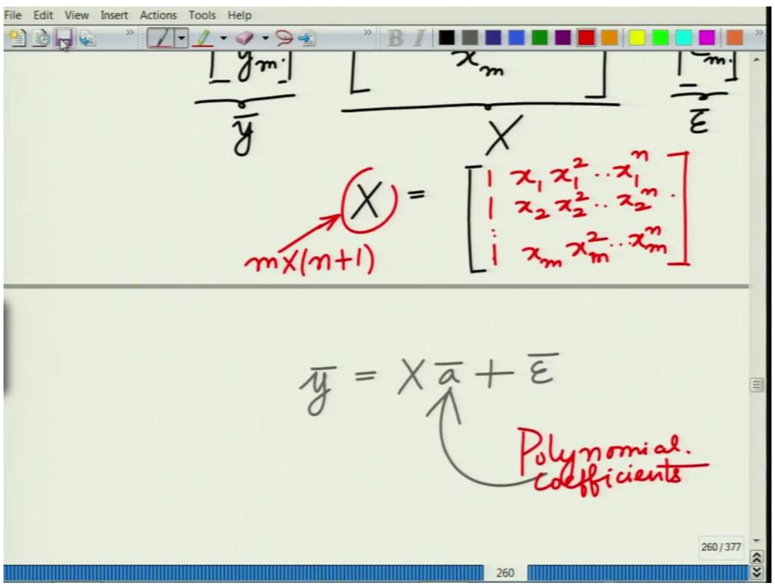


And once again, you can write this as y1 is equal to 1 x1 x1 square, so on up to x1 n times the column vector a naught, a1, so on up to an plus epsilon and this, you can call this as x1 bar transpose and this is essentially your vector A bar and this of course is the model error, let us call this epsilon 1, epsilon 1.

And therefore, I can express or I can develop an approximation for y1 as y1 equals x1 bar transpose a bar plus epsilon 1. Similarly, for the training point y2x2, I can develop the approximation y2 equal to x2 bar transpose a bar plus the error 2. So, you can write y2 as x2 bar transpose a bar plus epsilon 2, so on and so forth and finally, you can write ym equal to xm bar transpose a bar plus epsilon m.

(Refer Slide Time: 8:28)

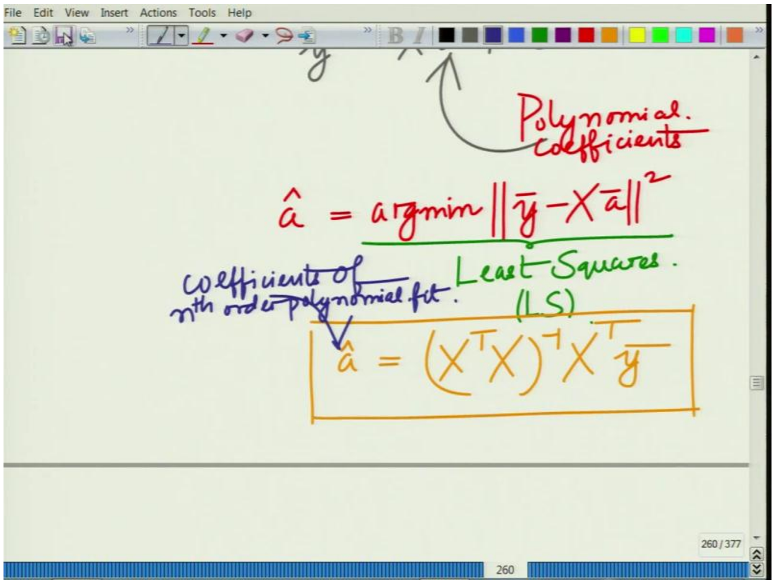




And now, again staking these as a vector, the vector model can be developed as we have y1, y2, yn or rather y1, y2, ym. This is equal to x1 bar transpose, x2 bar transpose, xm bar transpose times a bar plus you have the errors epsilon 1, epsilon 2, up to epsilon m. So, this is x, this is y bar and if you look at the matrix X, this has an interesting structure.

This will where X you can see the matrix X is composed of the polynomial, is composed of the terms 1 x1, x1 square, x1 n, 1 x2, x2 square, x2 n, so on, 1 xm, xm square so on, xm raise to the power of n. So, this is the structure of n, this is the structure of the matrix x. Naturally you can clearly see this is a matrix which is of size m cross n plus 1 that is the size of this matrix. So, therefore, that gives us the model y bar equal to once again our favorite model or by this what should be well known to you, x a bar plus the error. What is a bar? a bar contains the polynomial coefficients, so, we have this.

(Refer Slide Time: 11:07)



And now, once again you can determine a bar as the minimum as or the argmin as the rather minimizer of this cost function y bar minus x a bar square. So, you can write this as a hat to distinguish this the minimizer of y bar minus x a bar square. So, this is your once again you can see least squares in action and by this time, we will know that the solution is given as a hat equals x transpose x inverse x transpose y bar. So, these are the polynomial coefficients.

So, these are the coefficients of the nth order polynomial fit. So, that is essentially how the least squares can again, once again be used in a practical context to fit a polynomial, develop a polynomial approximation or essentially you can also think of this as a nonlinear approximation to a given set of responses.

So, once again you have these independent variables that is x1, x2, x1, the xs and you have the dependent variable of response y and you are developing sort of a polynomial approximation for the response based on the input data or the explanatory variables. So, once again least squares can be used for this application. So, let us stop here and continue with other, other related concepts in the subsequent modules. Thank you very much.