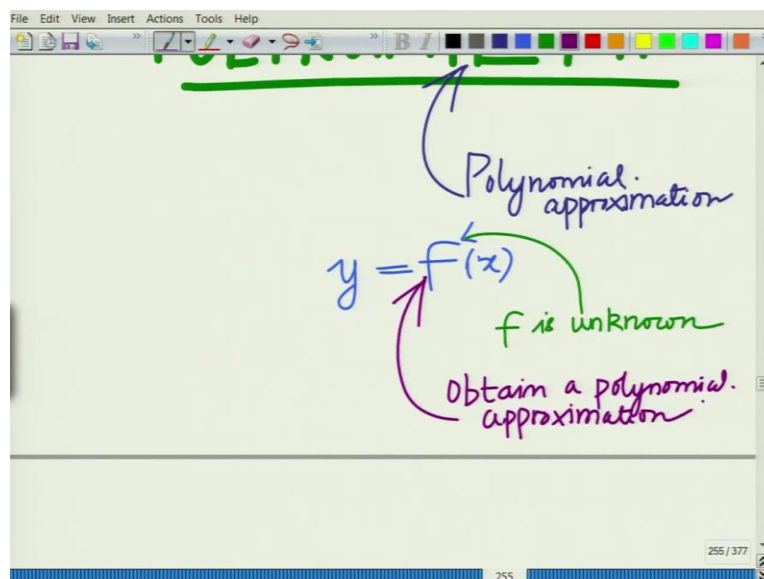
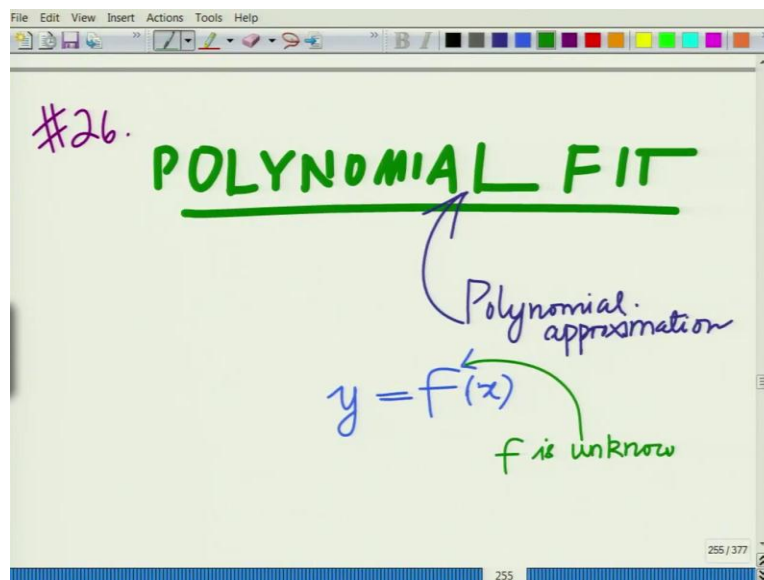


**Applied Linear Algebra for Signal Processing, Data Analytics and Machine Learning**  
**Professor Aditya K Jagannatham**  
**Department of Electrical Engineering**  
**Indian Institute of Technology Kanpur**  
**Lecture 26**  
**Computation Mathematics Application: Polynomial Fitting**

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)

The image shows a presentation slide with a white background and a blue border. At the top, there is a menu bar with 'File Edit View Insert Actions Tools Help'. Below the menu bar is a toolbar with various icons. The main content of the slide is handwritten in blue ink. It starts with the equation  $y = \hat{f}(x)$  followed by  $= a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ . Below this, there is a question: 'How to determine coefficients  $a_0, a_1, \dots, a_n$ ?' and a note: 'if  $n = 1$  straight line or Linear model.' A blue line is drawn under the polynomial equation, and the text 'Polynomial of degree n.' is written below it. The slide number '256 / 377' is visible in the bottom right corner.

That is we want to approximate  $y$  as  $\hat{f}$  of  $x$ , which is essentially a naught plus  $a_1x$  plus  $a_2x$  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,  $n$ th 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  $y_1$ . So, how to find the coefficients? Now, the question again is how to determine the coefficients. How to determine the coefficients a naught,  $a_1$ , so on up to  $a_n$ .

(Refer Slide Time: 4:40)

Training data.

$$\left. \begin{array}{l} y_1, x_1 \\ y_2, x_2 \\ \vdots \\ y_m, x_m \end{array} \right\} m \text{ Training points}$$
$$y_1 = a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n + \epsilon$$

model error

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)

$$y_1 = \underbrace{[1 \ x_1 \ x_1^2 \ \dots \ x_1^n]}_{\bar{x}_1^T} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}}_{\bar{a}} + \epsilon_1$$
$$y_1 = \bar{x}_1^T \bar{a} + \epsilon_1$$
$$y_2 = \bar{x}_2^T \bar{a} + \epsilon_2$$
$$\vdots$$
$$y_m = \bar{x}_m^T \bar{a} + \epsilon_m$$

And once again, you can write this as  $y_1$  is equal to  $1 \times x_1 \times x_1$  square, so on up to  $x_1^n$  times the column vector  $a_0, a_1$ , so on up to  $a_m$  plus epsilon and this, you can call this as  $\bar{x}_1$  transpose and this is essentially your vector  $\bar{a}$  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  $y_1$  as  $y_1$  equals  $\bar{x}_1$  transpose  $\bar{a}$  plus epsilon 1. Similarly, for the training point  $y_2, x_2$ , I can develop the approximation  $y_2$  equal to  $\bar{x}_2$  transpose  $\bar{a}$  plus the error 2. So, you can write  $y_2$  as  $\bar{x}_2$  transpose  $\bar{a}$  plus epsilon 2, so on and so forth and finally, you can write  $y_m$  equal to  $\bar{x}_m$  transpose  $\bar{a}$  plus epsilon m.

(Refer Slide Time: 8:28)

Stacking as a vector,

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\bar{y}} = \underbrace{\begin{bmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \vdots \\ \bar{x}_m^T \end{bmatrix}}_X \bar{a} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}}_{\bar{\epsilon}}$$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\bar{y}} = \underbrace{\begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix}}_X \bar{a} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}}_{\bar{\epsilon}}$$

$\textcircled{X} = \begin{bmatrix} | & x_1 & x_1^2 & \dots & x_1^n \\ | & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ | & x_m & x_m^2 & \dots & x_m^n \end{bmatrix}$   
 $m \times (n+1)$

$$\bar{y} = X \bar{a} + \bar{\epsilon}$$

Polynomial coefficients

And now, again stacking these as a vector, the vector model can be developed as we have  $y_1, y_2, \dots, y_m$  or rather  $y_1, y_2, y_m$ . This is equal to  $x_1$  bar transpose,  $x_2$  bar transpose,  $x_m$  bar transpose times  $\bar{a}$  plus you have the errors  $\epsilon_1, \epsilon_2, \dots, \epsilon_m$ . So, this is  $X$ , this is  $\bar{y}$  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  $1 \times x_1, x_1$  square,  $x_1$  to the power  $n, 1 \times x_2, x_2$  square,  $x_2$  to the power  $n$ , so on,  $1 \times x_m, x_m$  square so on,  $x_m$  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  $n + 1$  that is the size of this matrix. So,

therefore, that gives us the model  $\bar{y}$  equal to once again our favorite model or by this what should be well known to you,  $\bar{x}a$  plus the error. What is  $\bar{a}$ ?  $\bar{a}$  contains the polynomial coefficients, so, we have this.

(Refer Slide Time: 11:07)

The image shows a presentation slide with handwritten mathematical content. At the top, there is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Actions', 'Tools', and 'Help'. Below the menu, there is a toolbar with various icons. The main content of the slide is handwritten in red and blue ink. The title 'Polynomial coefficients' is written in red. Below it, the formula  $\hat{a} = \text{argmin} \|\bar{y} - X\bar{a}\|^2$  is written in red. A blue arrow points from the text 'Polynomial coefficients' to the  $\bar{a}$  in the formula. Below this, the text 'Least Squares (LS)' is written in green. To the left, the text 'Coefficients of nth order polynomial fit.' is written in blue. Below this, the formula  $\hat{a} = (X^T X)^{-1} X^T \bar{y}$  is written in orange and enclosed in a blue box. At the bottom right of the slide, there is a small text '260 / 377'.

And now, once again you can determine  $\bar{a}$  as the minimum as or the argmin as the rather minimizer of this cost function  $\bar{y}$  minus  $X\bar{a}$  square. So, you can write this as  $\hat{a}$  to distinguish this the minimizer of  $\bar{y}$  minus  $X\bar{a}$  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  $\hat{a}$  equals  $X^T X$  inverse  $X^T \bar{y}$ . So, these are the polynomial coefficients.

So, these are the coefficients of the  $n$ th 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  $x_1, x_2, x_1$ , the  $x$ s 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.