Neural Networks for Signal Processing – I

Prof. Shayan Srinivasa Garani

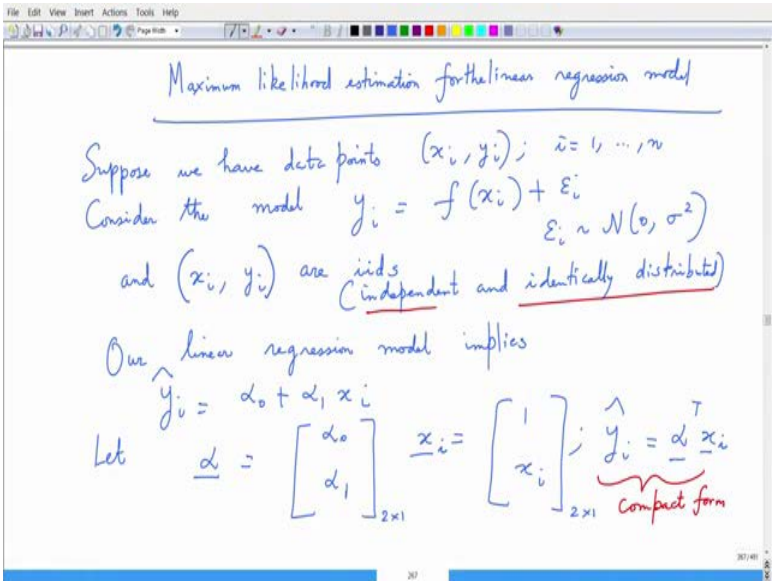
Department of Electronic Systems Engineering

Indian Institute of Science, Bengaluru

Lecture – 13

Linear Regression 2

(Refer Slide Time: 00:31)



Welcome to this module. We previously explored the least squares approach for solving the linear regression problem. Today, we will consider an alternative metric: maximum likelihood estimation (MLE) for the parameters based on the data.

Let's start by estimating the parameters for the linear regression model. We have data points (x_i, y_i) where i ranges from 1 to n, representing our observations.

We assume the model $y_i = f(x_i) + \epsilon_i$, where f represents the true relationship between x_i and y_i, and ϵ_i is a random error term. For our analysis, we assume that ϵ_i follows a normal distribution with zero mean and variance σ^2. All the error terms ϵ_i follow this distribution. Furthermore, we assume that x_i and y_i are independent and identically distributed (i.i.d) random variables, meaning each pair (x_i, y_i) is independently and identically distributed.

In our linear regression model, the estimated value of y_i, denoted as $\hat{y}_i$, is given by:

$$\hat{y}_i = \alpha_0 + \alpha_1 x_i$$

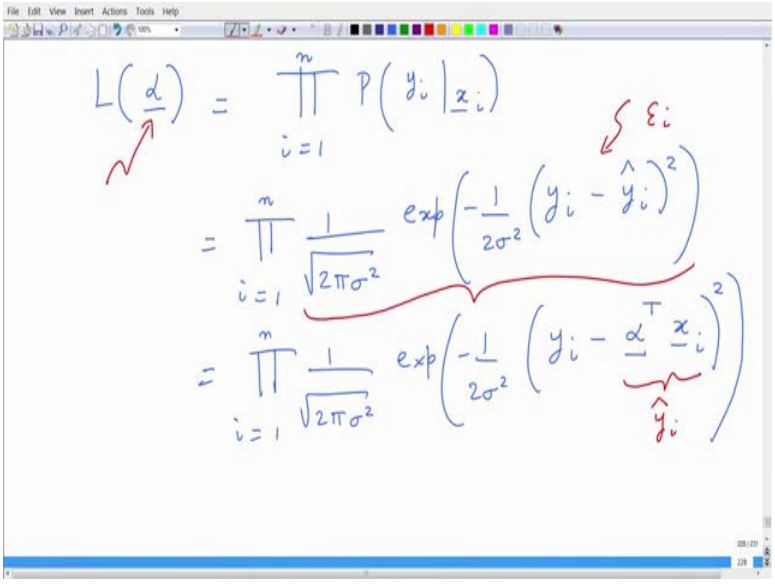
To simplify the mathematics, we'll use a vector formulation. Let the vector α be represented as a 2 × 1 column vector containing α_0 and α_1. We define x as a column vector with 1 and x_i. Thus, we can express $\hat{y}_i$, the estimate of y_i, as the inner product of these two vectors:

$$\hat{y}_i = \alpha^T x$$

This compact form represents $\hat{y}_i$ as a scalar, obtained by the dot product of the two vectors.

Finally, a quick note before concluding this slide: the pairs (x_i, y_i) are indeed i.i.d. This means that each pair (x_1, y_1), (x_2, y_2), (x_3, y_3), etc., are independently and identically distributed, and this characteristic should be noted explicitly in your analysis.

(Refer Slide Time: 03:42)



Let's now formulate the likelihood function in terms of our parameter, which is α that we need to estimate. Assuming that our observations are independent and identically distributed (i.i.d), we can express the likelihood function as the product of the probabilities of observing y_i given x_i, for i ranging from 1 to n.

To simplify, we note that y_i given x_i essentially represents the noise term, which is ϵ_i. Since we assume that ϵ_i follows a Gaussian distribution, the probability density function for y_i

given x_i is given by:

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y_i-\hat{y}_i)^2}{2\sigma^2}\right)$$

where $y_i - \hat{y}_i$ represents the noise ϵ_i, and $\hat{y}_i = \alpha^T x_i$ is our estimate of y_i.

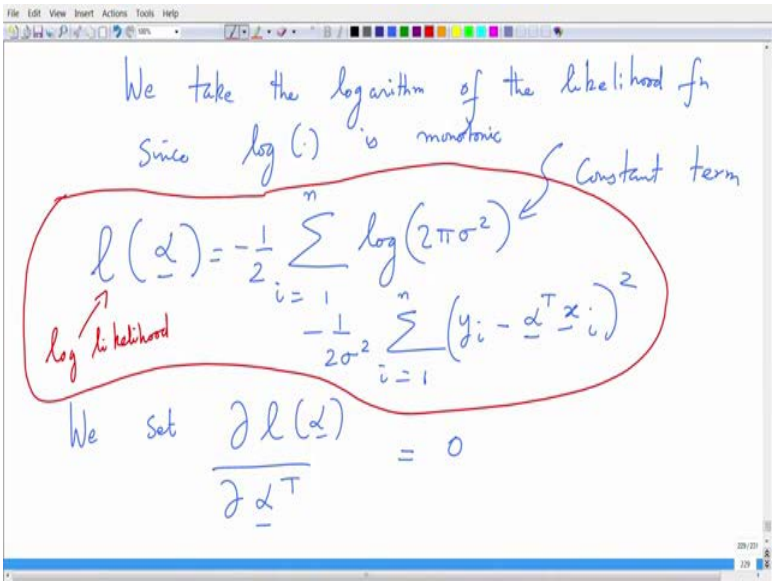
Plugging in our vector formulation for $\hat{y}_i$ into this equation, we get:

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y_i-\alpha^T x_i)^2}{2\sigma^2}\right)$$

Note that all these quantities are scalar values.

To simplify the likelihood function further, we take the natural logarithm of this product. This transformation converts the product into a sum, making it easier to handle mathematically.

(Refer Slide Time: 05:34)



By taking the natural logarithm of both sides of the likelihood function, we obtain the log-likelihood function. Here, the lowercase l denotes the log-likelihood. Simplifying this expression through some algebraic manipulation yields the following form:

$$\mathcal{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log(2\pi\sigma^2) + \frac{(y_i - \alpha^T x_i)^2}{\sigma^2}\right]$$

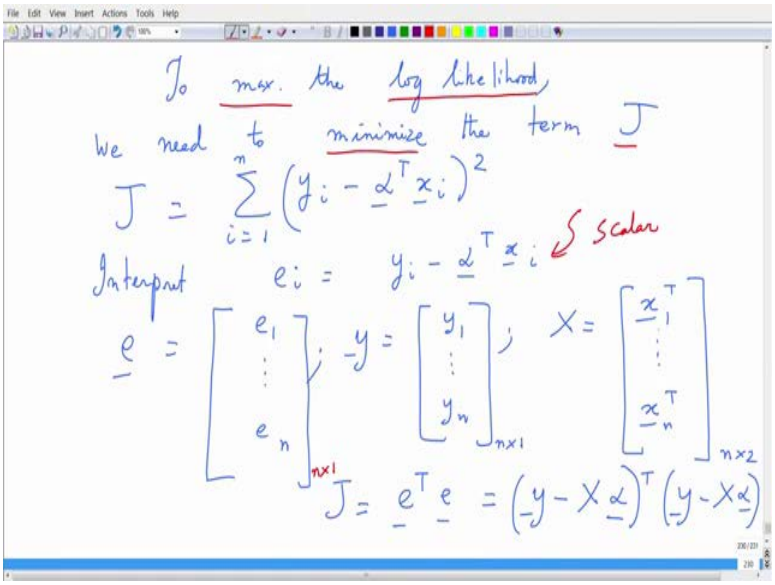
It's important to note that the term $-\frac{1}{2}\sum_{i=1}^{n}\log(2\pi\sigma^2)$ is a constant with respect to α and does not affect the maximization process. Therefore, we can ignore it when optimizing the log-likelihood function.

The remaining term involving α is what we focus on, as it contains the unknown parameter α that we need to estimate based on the observed y_i and the vectors x_i.

To find the maximum of the log-likelihood function, we set the derivative of the log-likelihood with respect to α equal to zero:

$$\frac{\partial \mathcal{L}(\alpha)}{\partial \alpha} = 0$$

(Refer Slide Time: 07:10)



Solving this equation is equivalent to minimizing the following quantity:

$$\sum_{i=1}^{n}(y_i - \alpha^T x_i)^2$$

This is a crucial observation, as maximizing the log-likelihood function reduces to

minimizing the residual sum of squares.

To maximize the log-likelihood function, we need to minimize the term J, given by:

$$J = \sum_{i=1}^{n} (y_i - \alpha^T x_i)^2$$

The negative sign in the likelihood function has been removed, leaving us with a positive quadratic cost function. This formulation is still not straightforward to simplify, so we use matrix calculus to assist with the process.

First, interpret the error term e_i as $y_i - \alpha^T x_i$. This error e_i is a scalar quantity. To streamline our calculations, we stack all the error observations from i = 1 to n into an n × 1 vector, which we denote as **e**.

Similarly, we construct the vector **y** which consists of the observations $y_1, y_2, \ldots, y_n$. Thus, **y** is an n × 1 column vector.

Next, we define the matrix **X**, which is formed by stacking all the vectors x_i for i = 1 to n. Each x_i is a 2 × 1 vector, resulting in **X** being an n × 2 matrix.

Now, the cost function J can be expressed in matrix notation. Specifically:

$$J = \mathbf{e^T e}$$

By substituting **e** into this, we get:

$$J = (\mathbf{y} - \mathbf{X}\alpha)^T (\mathbf{y} - \mathbf{X}\alpha)$$

Here, **X** is an n × 2 matrix, and **y** is an n × 1 vector. Consequently, **X** α is also an n × 1 vector. The matrix dimensions are consistent, so our formulation is correct.

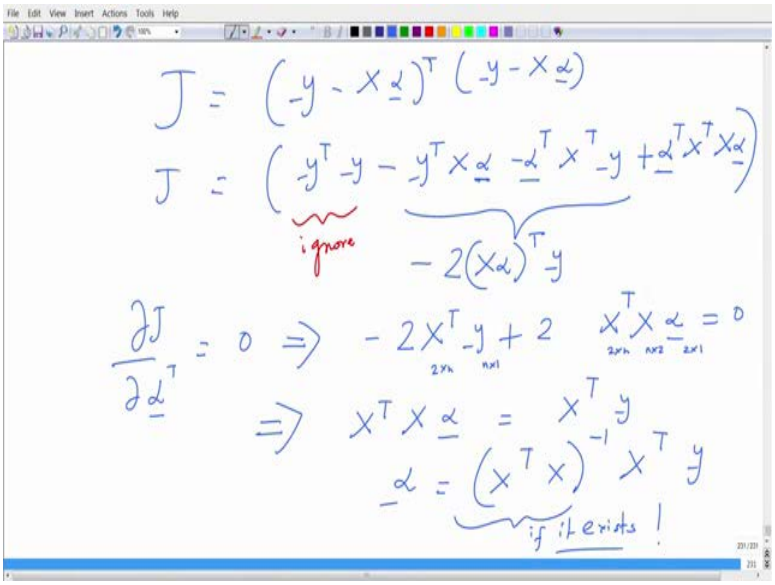
Let's proceed with simplifying our cost function. The cost function J can be expressed as:

$$J = (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha})$$

where $\boldsymbol{\alpha}$ is a vector. To simplify this, we first expand the expression:

$$J = \mathbf{y^T y} - \mathbf{y^T X}\boldsymbol{\alpha} - (\mathbf{X}\boldsymbol{\alpha})^T\mathbf{y} + (\mathbf{X}\boldsymbol{\alpha})^T(\mathbf{X}\boldsymbol{\alpha})$$

(Refer Slide Time: 09:46)



Using matrix algebra, we recognize that $(\mathbf{X}\boldsymbol{\alpha})^T = \boldsymbol{\alpha}^T\mathbf{X^T}$. Therefore:

$$J = \mathbf{y^T y} - \mathbf{y^T X}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{X^T y} + \boldsymbol{\alpha}^T\mathbf{X^T X}\boldsymbol{\alpha}$$

Notice that the terms involving $\mathbf{y}^T\ \mathbf{y}$ do not contain $\boldsymbol{\alpha}$ and thus can be ignored in the optimization process. We then focus on simplifying the remaining terms:

$$J = -2\boldsymbol{\alpha}^T\mathbf{X^T y} + \boldsymbol{\alpha}^T\mathbf{X^T X}\boldsymbol{\alpha}$$

Next, we take the partial derivative of J with respect to $\boldsymbol{\alpha}$ and set it equal to zero:

$$\frac{\partial J}{\partial \boldsymbol{\alpha}} = -2\mathbf{X^T y} + 2\mathbf{X^T X}\boldsymbol{\alpha} = 0$$

Solving for $\boldsymbol{\alpha}$:

$$\mathbf{X^T X}\boldsymbol{\alpha} = \mathbf{X^T y}$$

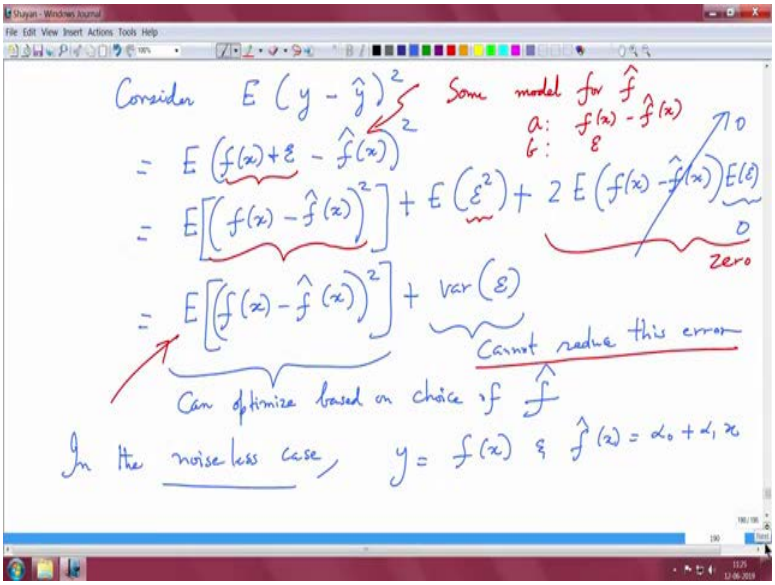
To isolate $\boldsymbol{\alpha}$, we compute:

$$\boldsymbol{\alpha} = (\mathbf{X^T X})^{-1}\mathbf{X^T y}$$

This equation requires the matrix $\mathbf{X^T X}$ to be invertible. If it is, we can solve for $\boldsymbol{\alpha}$ uniquely. However, if $\mathbf{X^T X}$ is not invertible, we must use a pseudo-inverse or apply regularization techniques to find a solution.

This process simplifies the derivation for estimating model parameters in linear regression using maximum likelihood estimation. We have explored two techniques: one using quadratic optimization with residual sum of squares and the other using maximum likelihood estimation. Depending on the situation, you can choose the technique that is most appropriate for your needs.

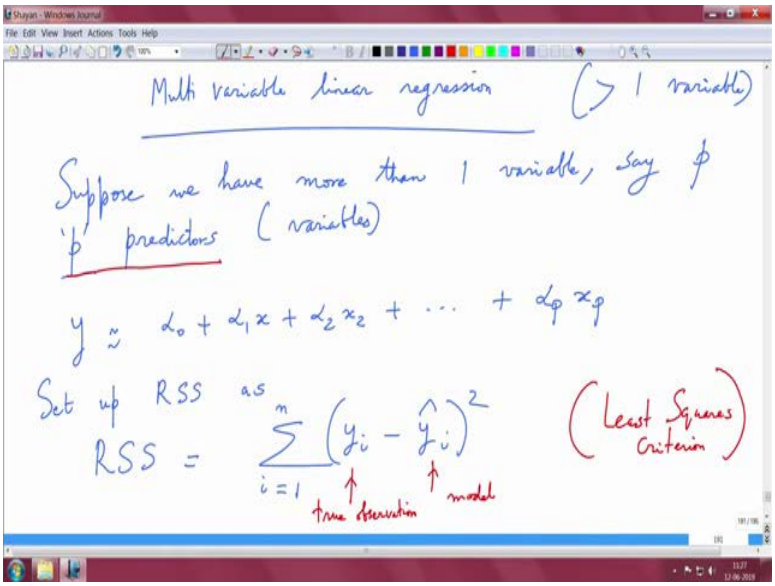
(Refer Slide Time: 15:15)



Now that we have discussed the concept of observation noise in the context of simple linear regression, let's move on to analyzing the regression problem with multiple variables. This scenario introduces additional complexity and involves more intricate calculations. We will explore how these complications arise and discuss strategies for simplifying the problem.

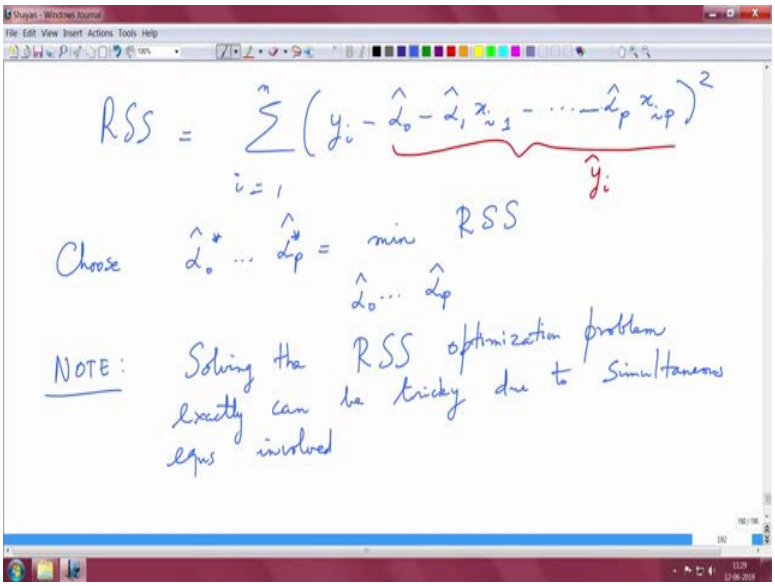
Let us consider the multiple-variable linear regression problem, also known as multivariable linear regression. In this case, we have more than one predictor variable—let's say p predictors. For example, in an agricultural scenario like predicting crop yield, variables might include soil pH content, moisture levels, pest control measures, and so on. These represent the different predictors we are working with.

(Refer Slide Time: 15:44)



Now, we set up the residual sum of squares, defined as $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where y_i is the observed value and $\hat{y}_i$ is the corresponding predicted value for each observation i. This is the true observation versus the model's estimate. We use the least squares criterion to formulate this, aiming to minimize this residual sum.

(Refer Slide Time: 17:11)

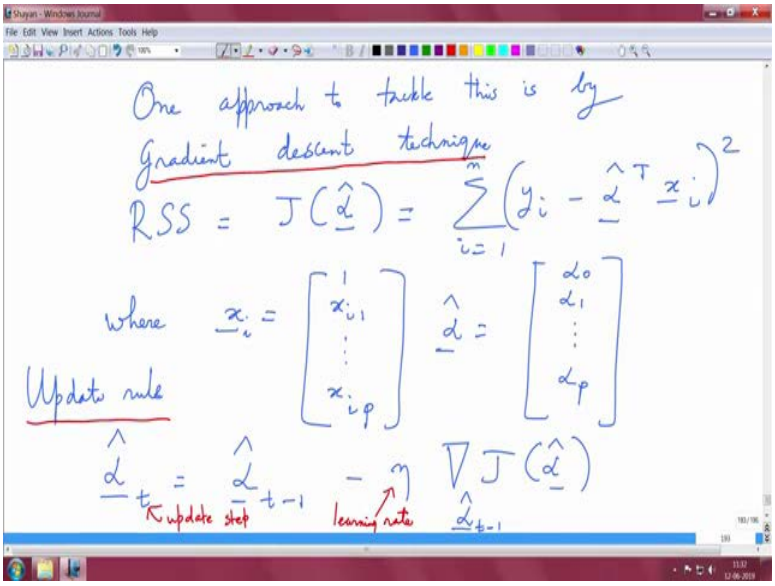


Let's simplify the problem. We start with the expression $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where $\hat{y}_i$ is the predicted value. This quantity includes a negative sign associated with each coefficient, such as $\hat{\alpha}_0$, $\hat{\alpha}_1$, and so forth. The objective is to minimize the residual sum of squares with

respect to these parameters $\hat{\alpha}_0$, $\hat{\alpha}_1$, up to $\hat{\alpha}_p$. By optimizing this quantity, we determine the optimal parameters $\widehat{\alpha_0^*}$, $\widehat{\alpha_1^*}$, and so on, up to $\widehat{\alpha_p^*}$.

You might wonder why we didn't perform this analytically for the single-variable case. In practice, calculating partial derivatives with respect to all parameters can become quite complex and cumbersome. For instance, if you have 20 or 25 variables, solving for all these partial derivatives and then addressing the resulting equations becomes impractical. Therefore, we avoid this direct approach due to the difficulty and impracticality of handling numerous parameters analytically.

(Refer Slide Time: 18:46)



To tackle this problem, we utilize the popular gradient descent approach. Here's how we proceed:

The residual sum of squares is a function of the parameter vector $\hat{\alpha}$, which includes all parameters from $\hat{\alpha}_0$ to $\widehat{\alpha_p}$. We represent $\hat{\alpha}$ as a column vector, and similarly, we define a vector $\mathbf{x}_i$ as $[1, x_i, x_i^2, \ldots, x_i^p]$. In this vector, the first term is 1 to account for the intercept $\hat{\alpha}_0$, while the remaining terms correspond to the predictors x_i up to x_i^p.

Thus, we can rewrite our equation in matrix form, expressing it as an inner product of $\hat{\alpha}$ with $\mathbf{x}_i$. This form helps us streamline our calculations.

Gradient descent then provides the update rule for $\hat{\alpha}$ at time t. Here, t denotes the iteration

step. Normally, n is used for discrete quantities, and t for continuous time, but since n is already representing the number of data points, we use t here for clarity.

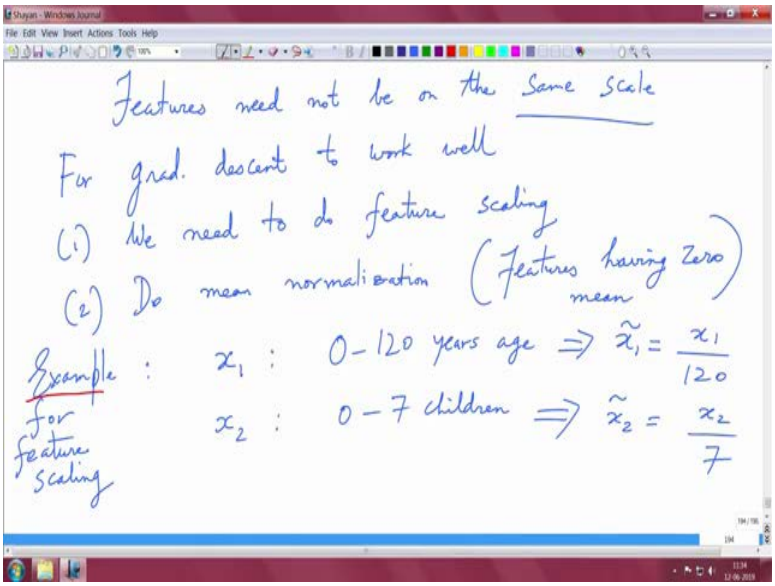
The update rule is given by:

$$\widehat{\alpha^{(t)}} = \widehat{\alpha^{(t-1)}} - \eta \nabla J\left(\widehat{\alpha^{(t-1)}}\right)$$

where η is the learning rate, and $\nabla J\left(\widehat{\alpha^{(t-1)}}\right)$ is the gradient of the cost function with respect to $\hat{\alpha}$ at time t-1. The negative sign indicates that we are moving in the direction of the negative gradient to minimize the cost function.

This update rule allows us to iteratively improve our estimate of $\hat{\alpha}$, ultimately solving for the entire vector in one shot. If the vector form feels cumbersome, you can simplify the problem by breaking it down into scalar components and solving for each variable individually.

(Refer Slide Time: 22:11)



One critical question we need to address from a practical standpoint is how to ensure that gradient descent performs effectively. When working with real-world data, attributes can vary significantly in scale. For example, consider two attributes: the age of individuals, which ranges from 0 to 120 years, and the number of children, which can range from 0 to 7. These attributes have vastly different scales.

To address this, we need to normalize the data. There are two key normalization steps that are crucial for the gradient descent algorithm to function properly: scale normalization and mean normalization. Let's break down each of these steps.

First, let's discuss feature scaling. Suppose we have two variables: x_1, representing age, which ranges from 0 to 120 years, and x_2, representing the number of children, which ranges from 0 to 7. To normalize these variables, we scale them to a range between 0 and 1.

For x_1, we normalize it as follows:

$$\widetilde{x_1} = \frac{x_1}{120}$$

Here, x_1 is divided by the maximum value of 120 years, which brings it into the range of 0 to 1.

Similarly, for x_2:

$$\widetilde{x_2} = \frac{x_2}{7}$$

This scales x_2 by dividing by the maximum number of children, 7, normalizing it to the 0 to 1 range.

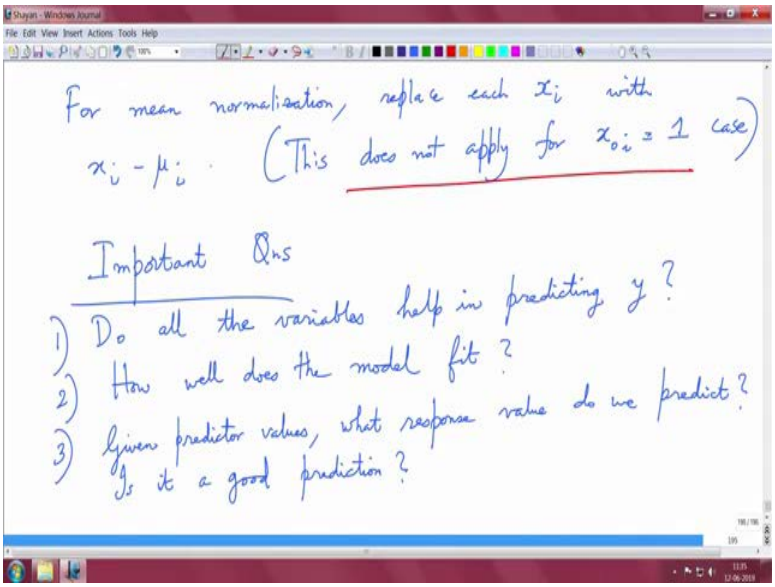
In cases where attributes can have negative values, such as a bank balance that can be negative if a person owes money, we need to adjust the normalization. Instead of scaling to the 0 to 1 range, we would typically scale to a range of -1 to 1. This ensures that negative values are appropriately handled within the normalized data range.

By applying these normalization techniques, we ensure that all features are on a comparable scale, which helps gradient descent converge more effectively and efficiently.

Next, we should consider mean normalization. To perform this, we need to adjust each variable x_i by subtracting its mean μ_i. This process effectively removes the bias from the data. Typically, mean normalization is applied after feature scaling. However, it's important to note that this adjustment does not apply to the intercept term x_0, which is simply a constant value of 1 for all observations. For all other variables $x_1, x_2, \ldots, x_p$, mean

normalization should be performed appropriately.

(Refer Slide Time: 24:29)



Once both feature scaling and mean normalization are completed, you can proceed with gradient descent on the transformed data. It is crucial to choose an appropriate learning rate for the gradient descent algorithm to ensure effective convergence.

After discussing these techniques, we need to address some important questions:

1. Do all variables contribute equally to predicting the response? For instance, if you are evaluating the impact of advertising across different media—newspapers, radio, television, and social media—not all predictors may contribute equally to the response. Some variables might have stronger dependencies or might be statistically insignificant. Evaluating this is crucial for understanding the relevance of each variable.

2. How well does the model fit the data? This involves calculating the Residual Sum of Squares (RSS) and assessing if further simplification of the model is possible. It's important to determine how well the model represents the underlying data.

3. Given the predictor values, what response value should we predict? This question addresses the practical application of the model. For instance, in policy or decision-making scenarios, accurate predictions are essential as they can influence significant decisions. A sound model with reliable predictions is therefore crucial.

Understanding these aspects helps in evaluating the model's accuracy and effectiveness. Although linear regression might be considered a basic and traditional statistical model, it remains fundamental due to its wide range of implications and applications. We will conclude this discussion here and then explore variations of the regression model, including interaction effects.