Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

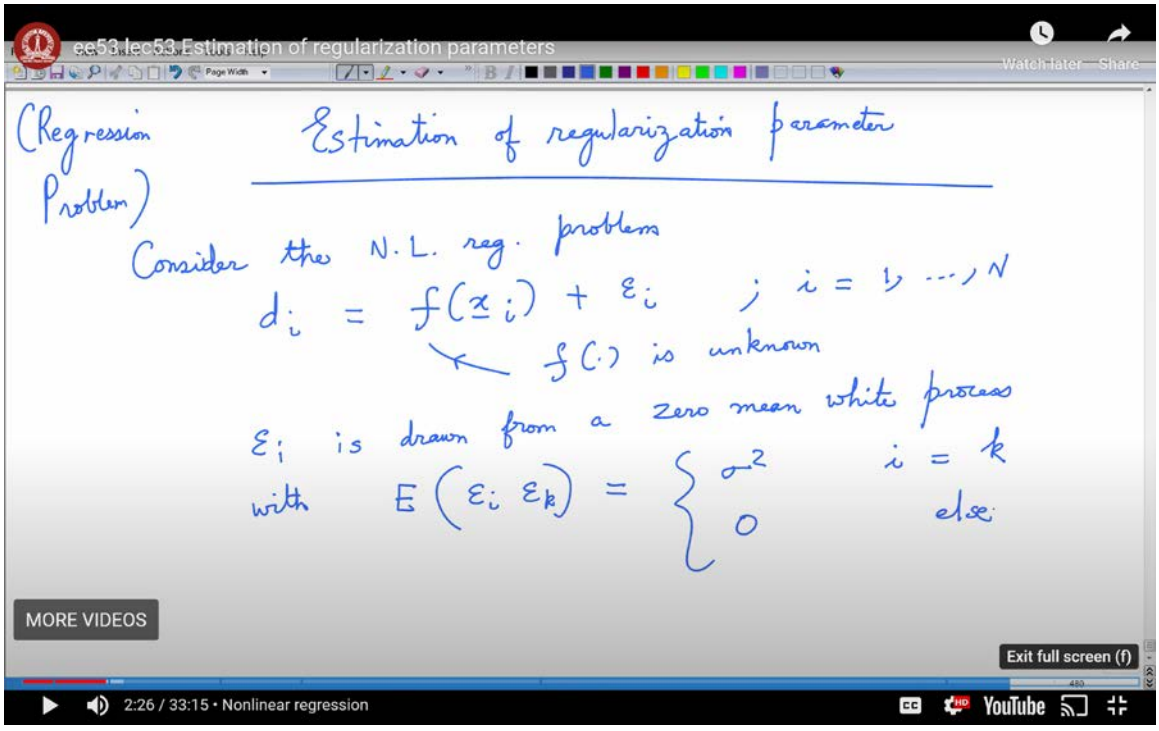
Indian Institute of Science, Bengaluru

Lecture – 53

Estimation of Regularization Parameters

Let's revisit the regression problem, but this time within the framework of regularization. Our exploration of neural networks often involves approximating or solving regression and classification problems, among other applications. Now, let's return to the regression problem, but with a focus on regularization.

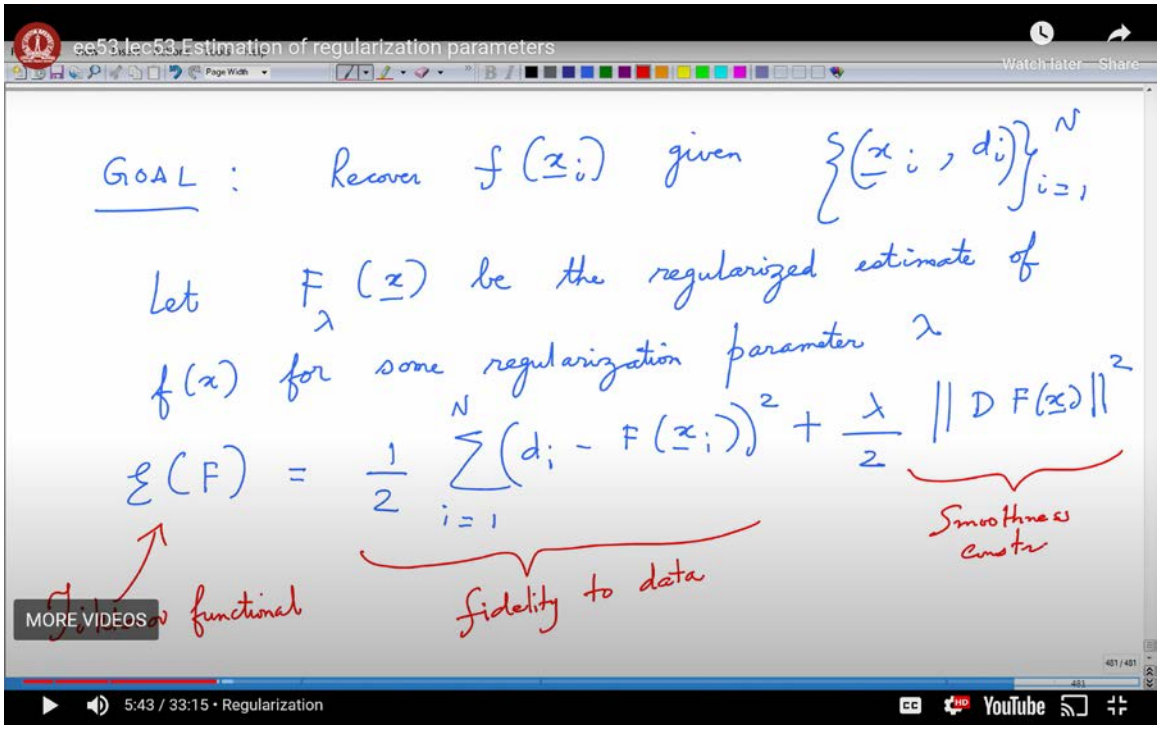
(Refer Slide Time: 02:26)



Consider the non-linear regression problem we have encountered numerous times before. We represent it as follows: $d_i = f(x_i) + \epsilon_i$ for i = 1 to N, where N denotes the number of data points, and f is an unknown function. The error term ϵ_i is assumed to be drawn from a

zero-mean white noise process, with covariance defined by $E[\epsilon_i \epsilon_k] = \sigma^2$ when i = k, and zero otherwise.

(Refer Slide Time: 05:43)



We have previously addressed this regression problem, but now we will introduce regularization to enhance it. To set this up, our goal is to estimate the function $f(x_i)$, meaning we need to approximate f using the given data points x_i and d_i for i = 1 to N. To achieve this, we need to incorporate regularity conditions.

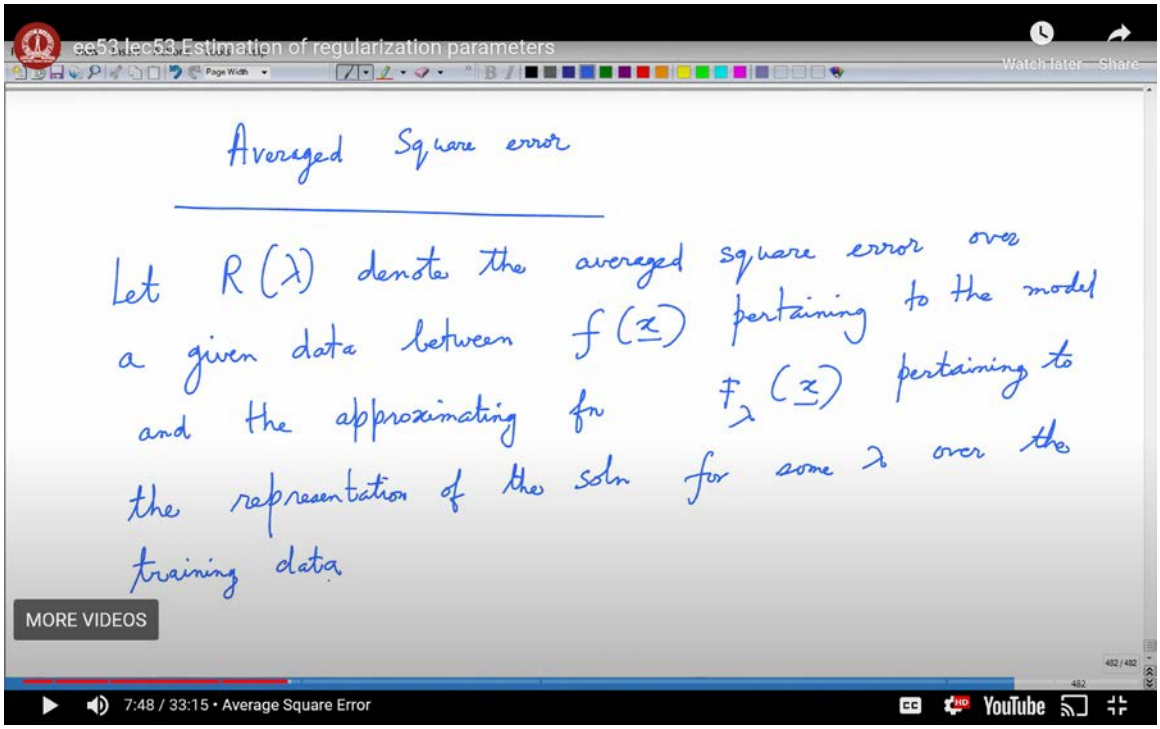
Let $F_\lambda(x)$ denote the regularized estimate of the unknown function f(x), where λ is the regularization parameter. This parameter imposes a smoothness constraint on the function we are approximating.

Our approach uses the Tikhonov functional E, which consists of two components. The first part measures the squared error between the desired function and the approximated function at each data point x_i. This component assesses the fidelity to the data. The second part introduces the regularization element, involving a differential linear operator applied

to the function f(x). This operator's norm, evaluated in the function space, imposes the smoothness constraint.

With this setup, let R(λ) represent the average squared error over the given data set.

(Refer Slide Time: 07:48)



Let's delve deeper into our analysis. We are comparing two functions: the unknown function f(x), which represents the model, and the approximating function $f_\lambda(x)$, which provides a solution based on some regularization parameter λ applied to the training set. The key distinction here, compared to the setup discussed earlier in basic regression, is the inclusion of this regularization term. This is why we introduce the parameter λ in the subscript.

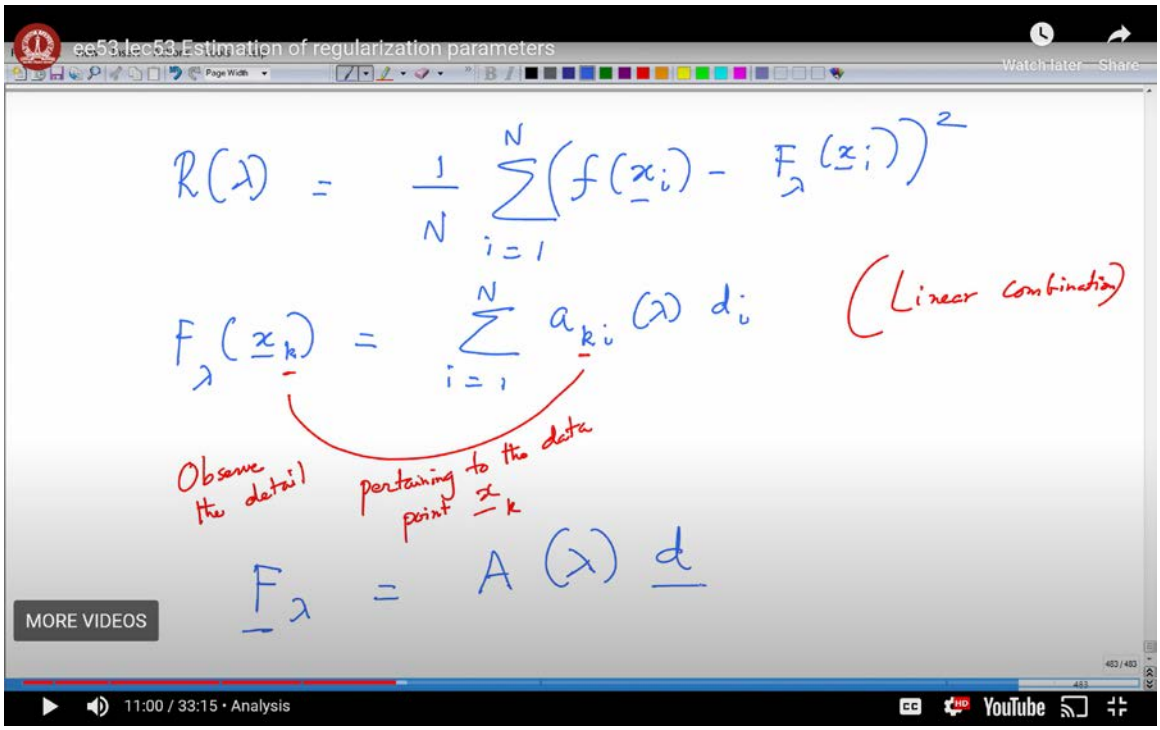
Now, let’s examine R(λ) more closely. This measure averages the error between the actual function values $f(x_i)$ and the approximated values $f_\lambda(x_i)$. The function $f_\lambda(x_i)$ represents the approximation under the regularization constraint, which is why λ is included as a subscript. The unknown function $f(x_i)$ is what we aim to approximate, and R(λ) is defined as this error measure.

The function f_λ as a function of the data points can be expressed as:

$$f_\lambda(x_i) = \sum_{k=1}^{n} a_{ki}(\lambda) \cdot d_i$$

Here's what this expression means: we have our desired sample values d_i combined using coefficients $a_{ki}(\lambda)$, which depend on λ, to obtain the function value at point x_k. Note that the subscript k pertains to the data point, and i denotes the index in the summation, representing a linear combination of the samples.

(Refer Slide Time: 11:00)



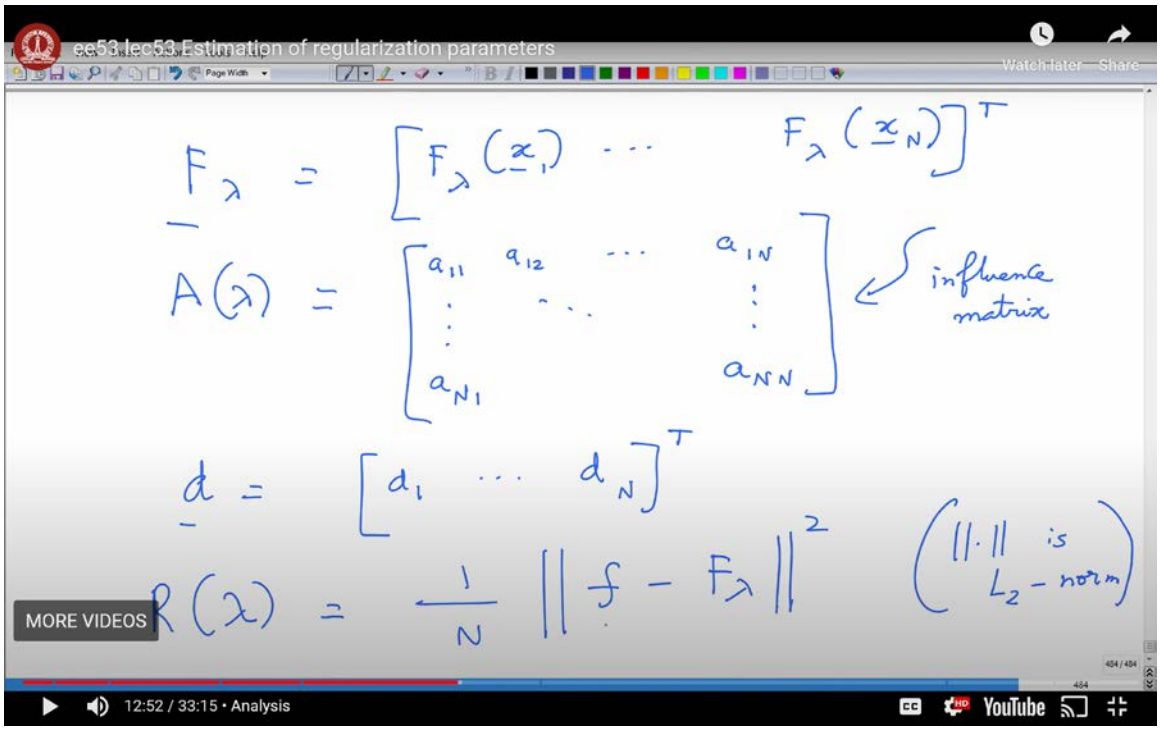
We can represent this in matrix form. Let f_λ be the column vector of f_λ evaluated at each data point $x_1, \ldots, x_n$. This vector has a size of n × 1. Let A(λ) be the influence matrix, an n × n matrix with elements a_{ij}, where i and j range from 1 to n. The influence matrix A(λ) is composed of coefficients that influence the approximation. The vector D contains the desired responses $d_1, d_2, \ldots, d_n$ and is also an n × 1 column vector.

Thus, f_λ can be written as:

$$f_\lambda = A(\lambda) \cdot D$$

where $\boldsymbol{f}_\lambda$ is the vector of function values at $x_1, \ldots, x_n$, $A(\lambda)$ is the influence matrix, and $\boldsymbol{D}$ is the vector of desired responses.

(Refer Slide Time: 12:52)



The term $R(\lambda)$ represents the squared norm of the difference between $\boldsymbol{f}$ and $\boldsymbol{f}_\lambda$, specifically the L^2 norm. Note that sometimes the bar over $\boldsymbol{f}$ is omitted, implying that it refers to a vector.

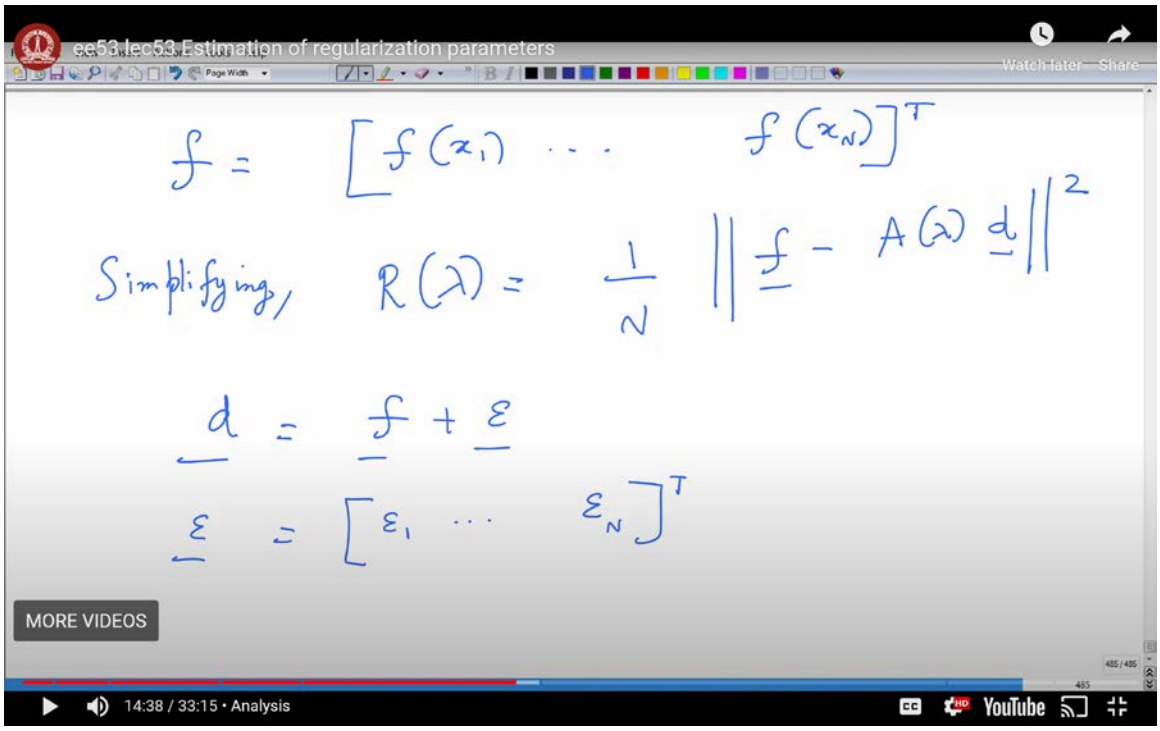
To simplify $R(\lambda)$, we have:

$$R(\lambda) = \frac{1}{n}|f - A(\lambda) \cdot D|^2$$

where $\boldsymbol{f}$ represents the function values at $x_1, \ldots, x_n$, and $\boldsymbol{D}$ is the vector of desired responses. Additionally, $\boldsymbol{D}$ can be expressed as $\boldsymbol{f} + e$, where $\boldsymbol{e}$ represents the noise realization with components $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$ over the n data points.

Let’s proceed with further simplification. For clarity, I will introduce numbering for the equations in this setup.

(Refer Slide Time: 14:38)



Let's label the equations for clarity. We'll refer to the initial equation as Equation 1 and the second equation as Equation 2. By substituting Equation 2 into Equation 1, where d = f + ϵ, and then simplifying, we obtain:

$$R(\lambda) = \frac{1}{n}|f - A(\lambda)d|^2$$

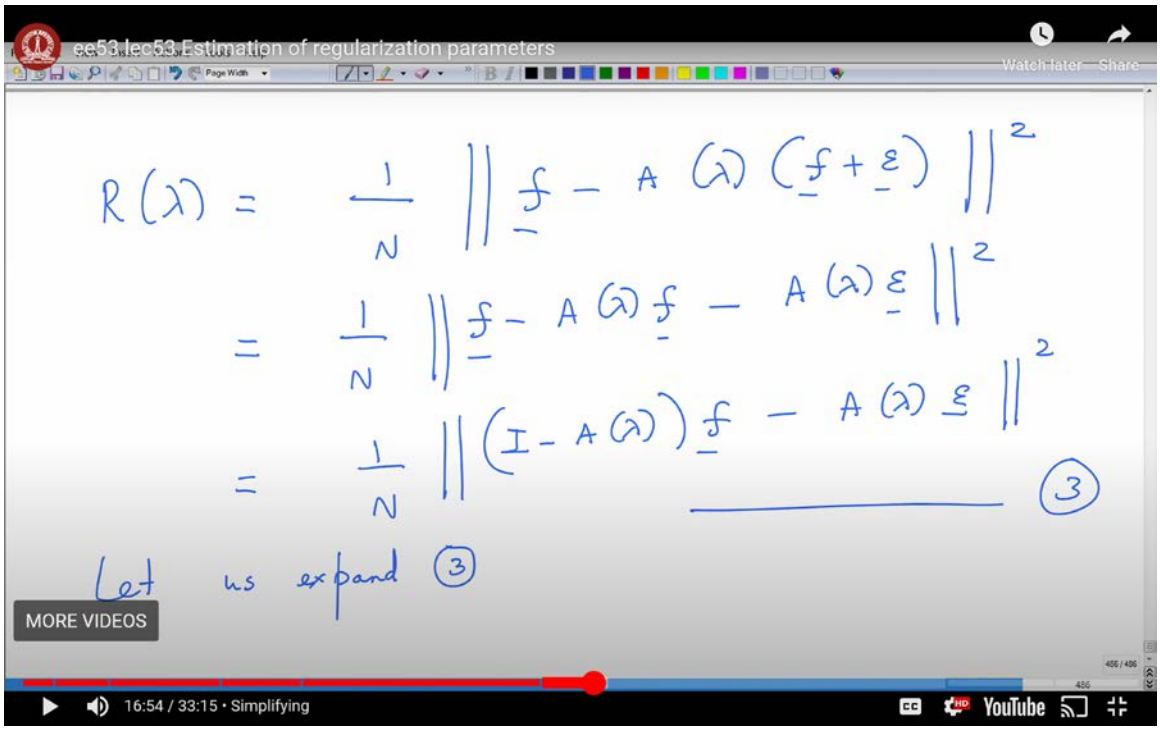
Given that $\boldsymbol{D}$ = f + ϵ, we need to substitute this into the norm expression:

$$R(\lambda) = \frac{1}{n}|f - A(\lambda)(f + \epsilon)|^2$$

To simplify, let's expand this expression:

$$R(\lambda) = \frac{1}{n}|f - A(\lambda)f - A(\lambda)\epsilon|^2$$

(Refer Slide Time: 16:54)



We can further simplify this as:

$$R(\lambda) = \frac{1}{n}|\big(I - A(\lambda)\big)f - A(\lambda)\epsilon|^2$$

Next, we expand this norm squared:

$$R(\lambda) = \frac{1}{n}\big[|\big(I - A(\lambda)\big)f|^2 - 2\big(I - A(\lambda)\big)f^T A(\lambda)\epsilon + |A(\lambda)\epsilon|^2\big]$$

This can be written in a more compact form:

$$R(\lambda) = \frac{1}{n}\big[|\big(I - A(\lambda)\big)f|^2 - 2\big(I - A(\lambda)\big)f^T A(\lambda)\epsilon + |A(\lambda)\epsilon|^2\big]$$

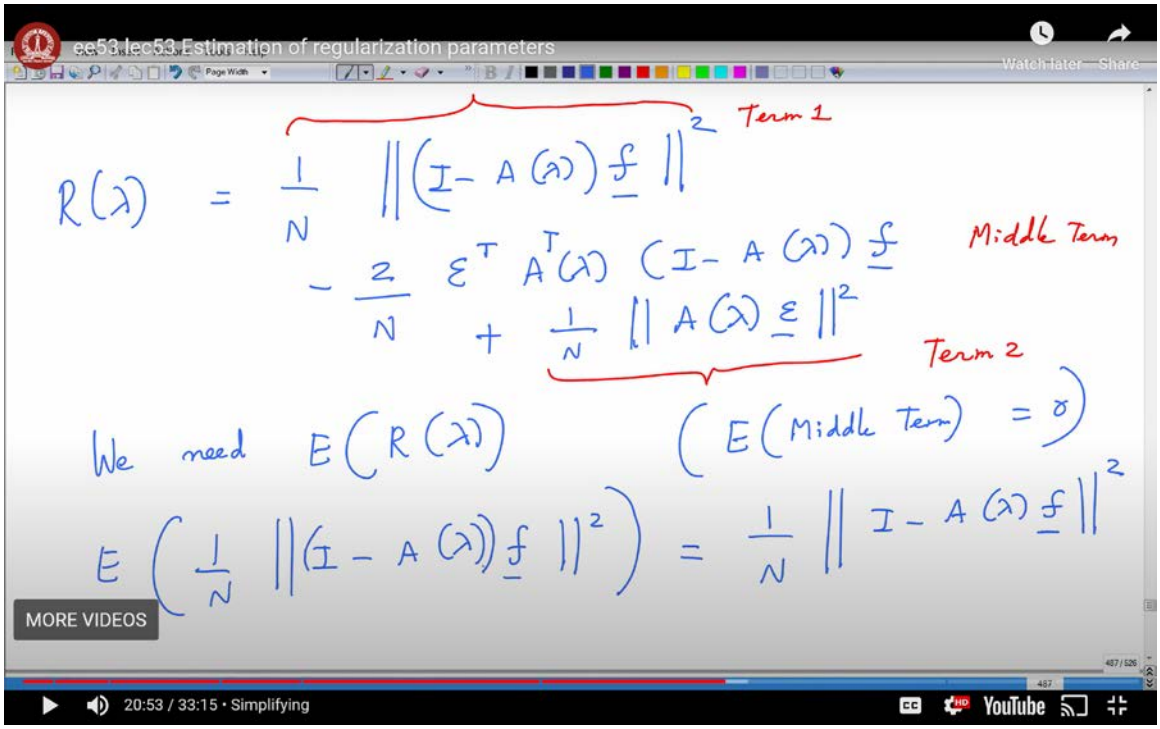
Here, the term involving ϵ simplifies due to the properties of the expectation. Since ϵ has zero mean, the cross term $-2\big(I - A(\lambda)\big)f^T A(\lambda)\epsilon$ vanishes when taking the expectation. Thus, we focus on computing the expectations of the remaining terms.

So, we need to compute:

$$1.\mathrm{E}\left[\frac{1}{n}|(I-A(\lambda))f|^2\right]$$

$$2.\mathrm{E}\left[\frac{1}{n}|A(\lambda)\epsilon|^2\right]$$

(Refer Slide Time: 20:53)



The first term does not involve randomness and thus is simply:

$$\frac{1}{n}|(I-A(\lambda))f|^2$$

For the second term, we calculate the expected value of $|A(\lambda)\epsilon|^2$. This can be expressed as:

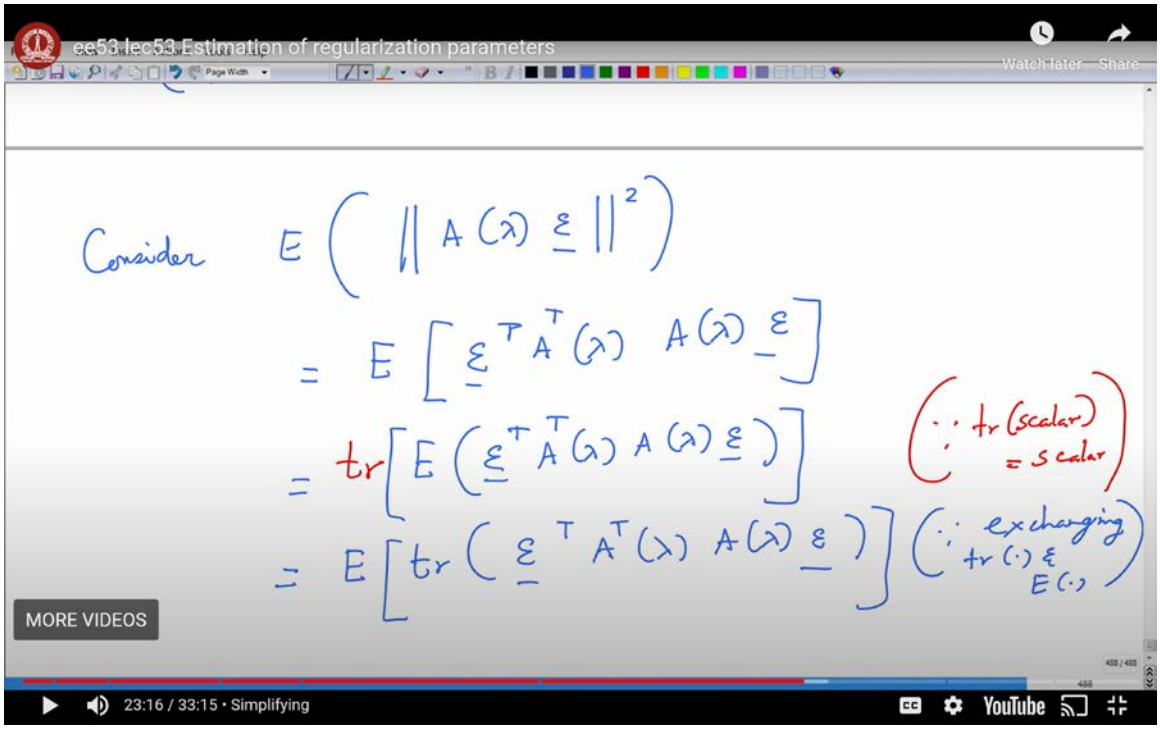
$$\mathrm{E}[|A(\lambda)\epsilon|^2] = \mathrm{E}[\epsilon^T A(\lambda)^T A(\lambda)\epsilon]$$

Given that ϵ has zero mean and its covariance is σ^2 I, this expectation simplifies to:

$$\mathrm{E}[\epsilon^T A(\lambda)^T A(\lambda)\epsilon] = \sigma^2 \mathrm{Tr}\left(A(\lambda)^T A(\lambda)\right)$$

where Tr denotes the trace of the matrix.

(Refer Slide Time: 23:16)



Thus, the term involving the noise ϵ becomes:

$$\frac{\sigma^2}{n}\mathrm{Tr}\left(A(\lambda)^T A(\lambda)\right)$$

This completes the simplification process for R(λ).

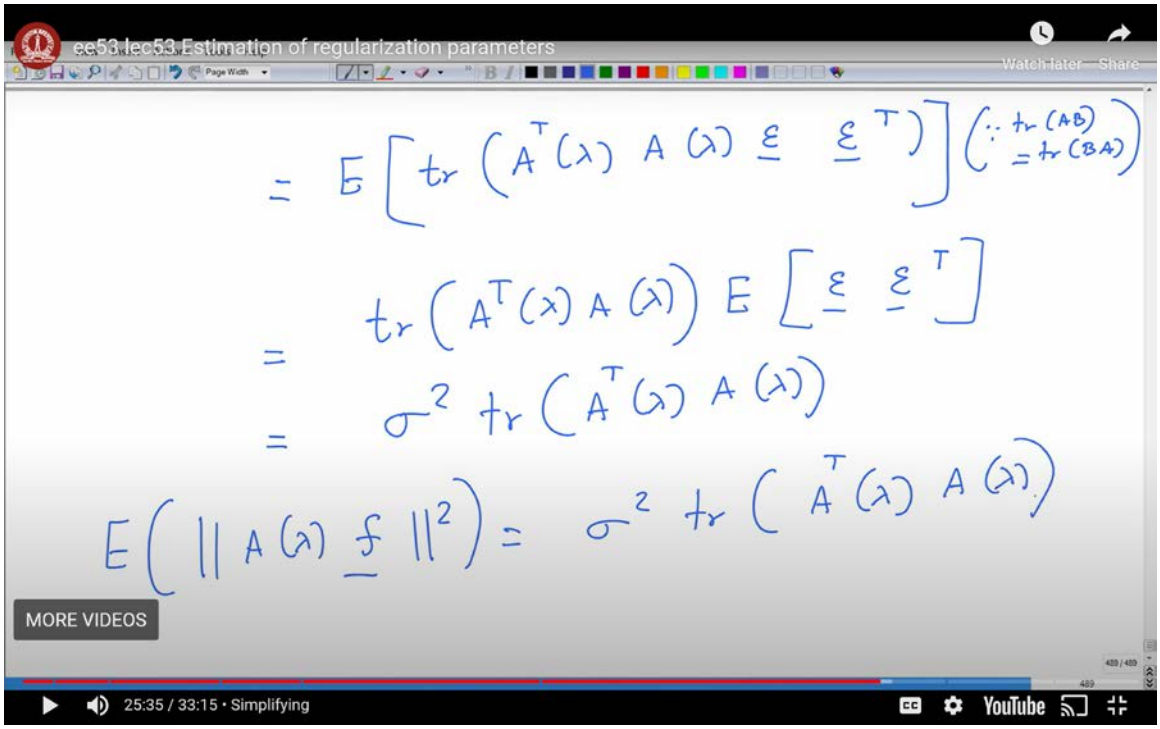
We have established that the norm squared is a scalar quantity. Therefore, we can apply the trace operation, as the trace of a scalar is simply the scalar itself. To proceed, consider the trace of the expected value of the expression $\epsilon^T A(\lambda)^T A(\lambda)\epsilon$. We can interchange the expectation and trace operations because the trace of a linear operator is invariant under cyclic permutations. This property allows us to simplify the expression as follows:

$$\mathrm{E}[\mathrm{Tr}(\epsilon^T A(\lambda)^T A(\lambda)\epsilon)]$$

By invoking the cyclic property of the trace, we rearrange it to:

$$\mathrm{E}[\mathrm{Tr}(A(\lambda)^T A(\lambda)\epsilon\epsilon^T)]$$

(Refer Slide Time: 25:35)



This reordering makes it easier to simplify because we can pull out the term $A(\lambda)^T A(\lambda)$ from the trace and compute its expectation. The expectation of $\epsilon\epsilon^T$ is σ^2 I (where σ^2 is the noise variance and I is the identity matrix), so:

$$\mathrm{E}[\mathrm{Tr}(A(\lambda)^T A(\lambda)\epsilon\epsilon^T)] = \sigma^2\mathrm{Tr}\big(A(\lambda)^T A(\lambda)\big)$$

Thus, the expected value of the norm squared of A(λ) ϵ is:

$$\mathrm{E}[|A(\lambda)\epsilon|^2] = \sigma^2\mathrm{Tr}\big(A(\lambda)^T A(\lambda)\big)$$

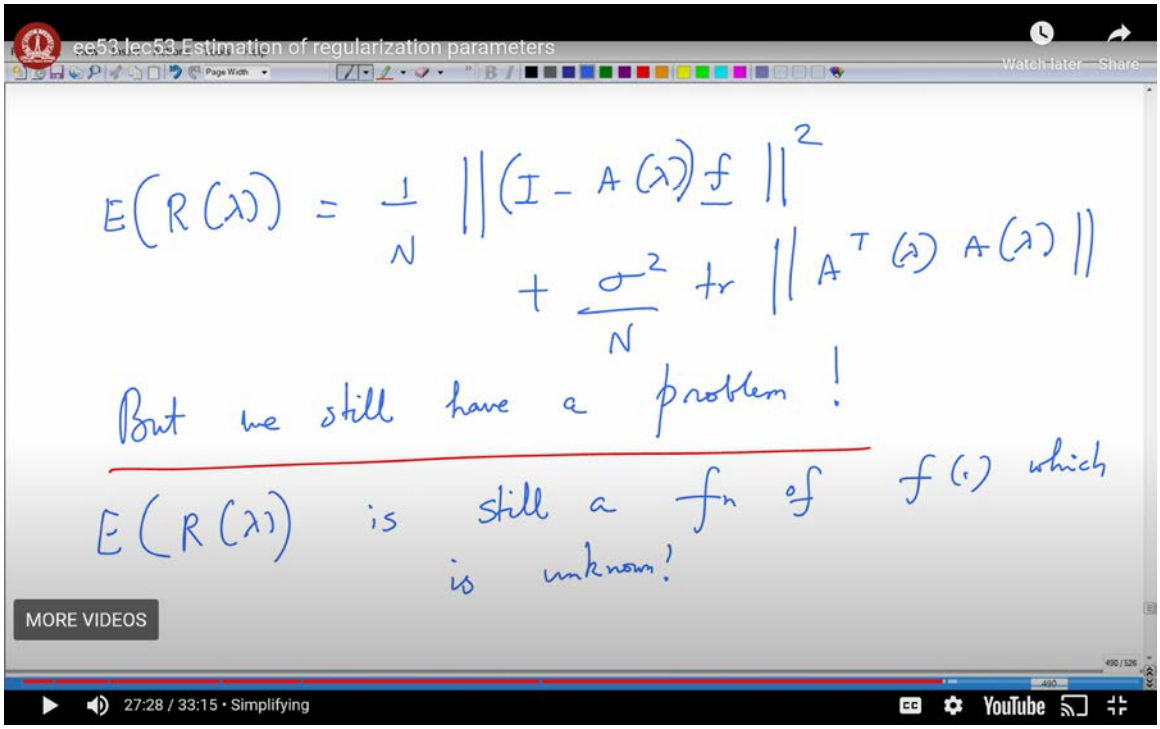
There was a minor mistake earlier where f should have been ϵ, so this simplifies to:

$$\mathrm{E}[|A(\lambda)\epsilon|^2] = \sigma^2\mathrm{Tr}\big(A(\lambda)^T A(\lambda)\big)$$

Now, we can express the expected value of R(λ) as:

$$\mathrm{E}[R(\lambda)] = \frac{1}{n}\left[\left|\left(I - A(\lambda)\right)f\right|^2 + \sigma^2 \mathrm{Tr}\left(A(\lambda)^T A(\lambda)\right)\right]$$

(Refer Slide Time: 27:28)



However, this average error still depends on the unknown regression function f. To obtain a reasonable estimate of R(λ), we replace f with the vector ***D***. This adjustment gives us the estimate $\hat{R}(\lambda)$:

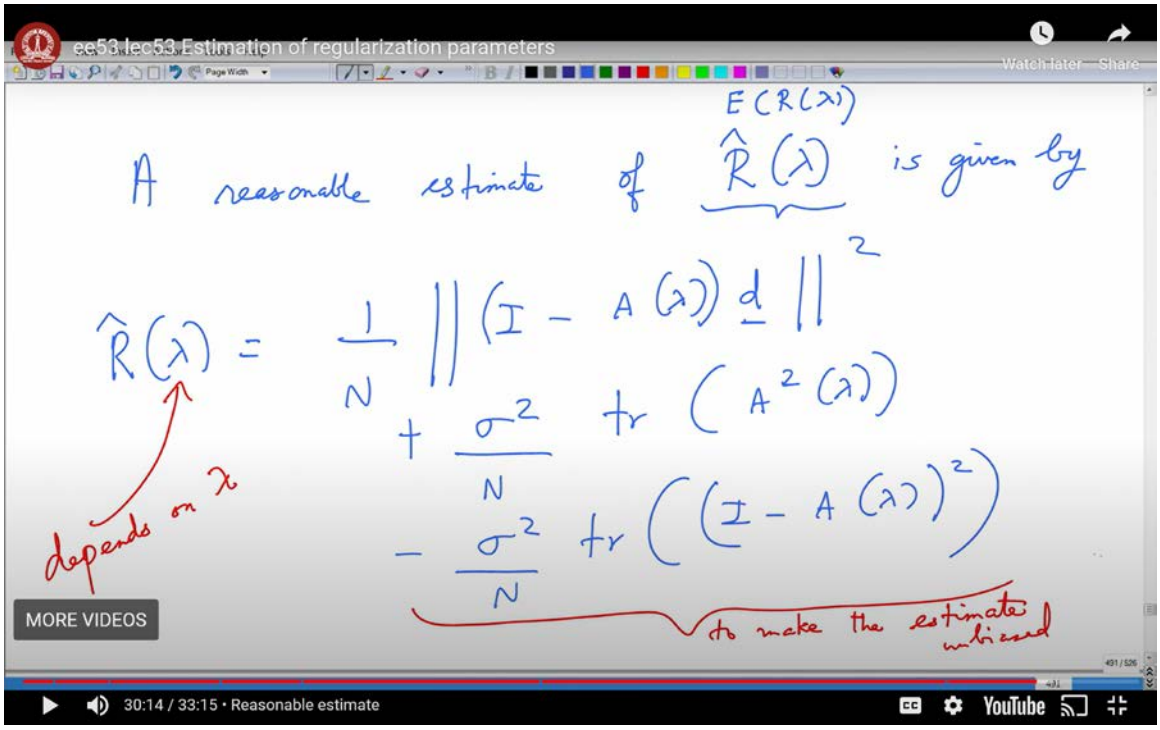
$$\hat{R}(\lambda) = \frac{1}{n}\left[\left|\left(I - A(\lambda)\right)d\right|^2 + \sigma^2 \mathrm{Tr}\left(A(\lambda)^T A(\lambda)\right)\right]$$

This estimate $\hat{R}(\lambda)$ is designed to be unbiased. I encourage you to verify that this estimate is indeed unbiased as an exercise. Note that $\hat{R}(\lambda)$ depends on the regularization parameter λ, which is crucial for adjusting the regularization effect.

To clarify the notation, let's reintroduce the braces. The notation $\hat{R}(\lambda)$ essentially represents the expected value of R(λ), which is included here for clarity. By minimizing the average error over all possible choices of λ, you can determine the optimal λ and, consequently, obtain a refined estimate.

At this point, you might wonder how to choose the optimal λ. Recall that λ is related to the influence matrix, which varies based on λ. By directly substituting the influence matrix parameters into the Tikhonov functional and solving for the matrix entries, you effectively identify the parameters that optimize the regularized cost function.

(Refer Slide Time: 30:14)



However, it's important to note that simply inserting the values of a_{ij} into the functional might not be sufficient to solve the problem. You may need to impose additional constraints on the influence matrix entries to ensure a unique solution.

The main takeaway from our analysis is that you can augment your regression problem with a regularization term, leading to a regularized cost function. By applying straightforward algebraic manipulations, you arrive at an error estimate dependent on λ. This estimate reflects the problem's geometry and can be optimized by adjusting the parameters within the functional and setting up conditions for optimality.

In summary, this overview of regularization has illustrated how to transition from simple regression models, such as linear and logistic regression, to incorporating regularization

and optimizing the resulting solution. This approach highlights the relevance of neural networks in addressing such problems. With this, we conclude this module and will proceed with further topics.