Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

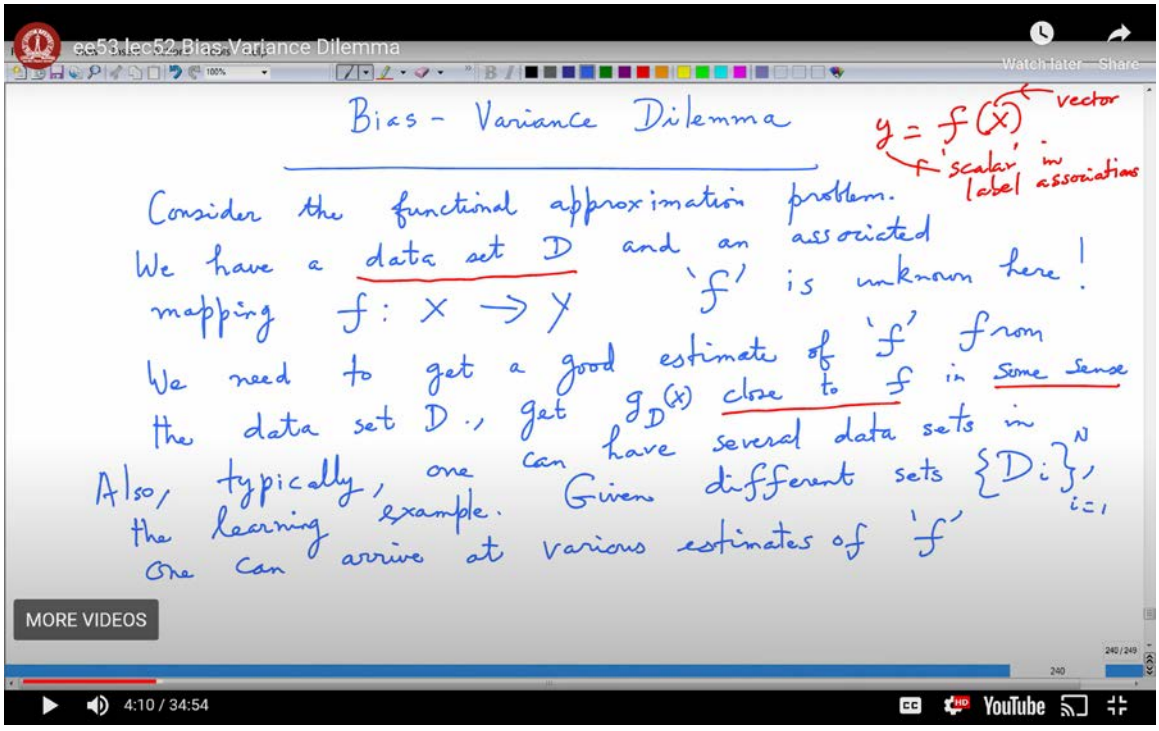
Indian Institute of Science, Bengaluru

Lecture – 52

Bias-Variance Dilemma

Having explored the regression problem, let us delve into the scenario where we encounter various regression functions that depend on different datasets. We aim to understand how bias and variance are evaluated across these datasets and data points. To set up this problem carefully, consider the functional approximation challenge.

(Refer Slide Time: 04:10)

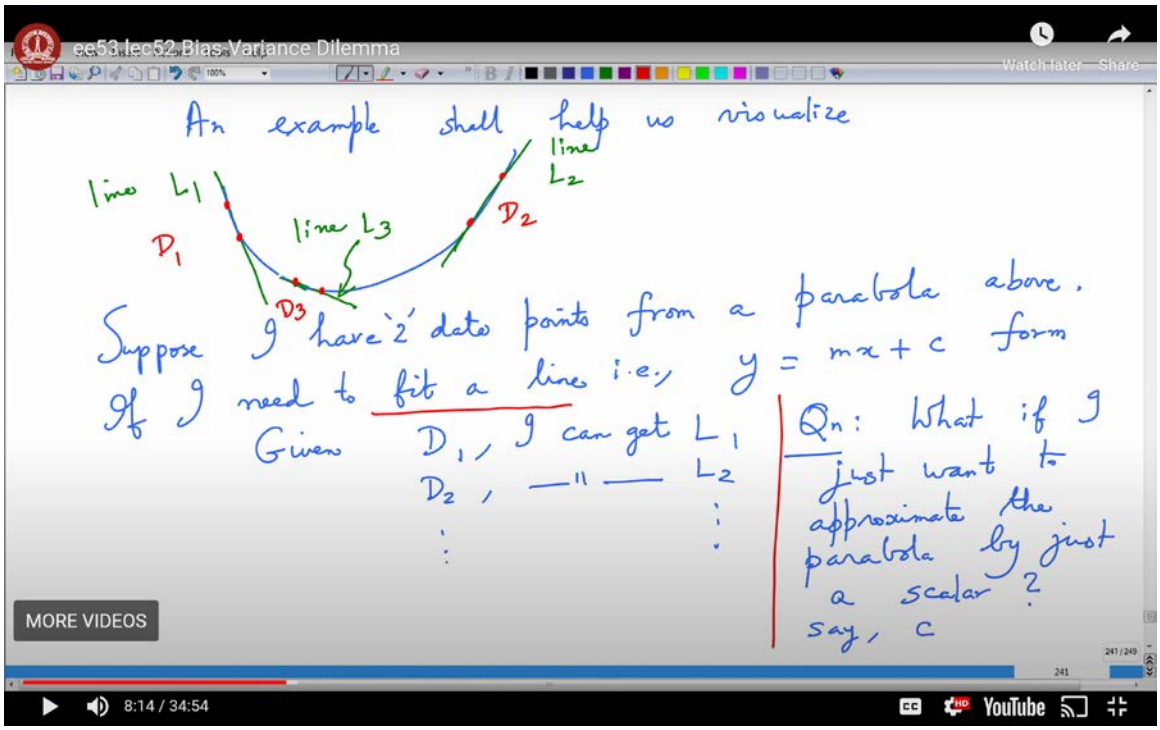


We are provided with a dataset D and a corresponding mapping f from x to y. Typically, this relationship is expressed as y = f(x), where f(x) denotes the function that maps x to y. Note that x can be a vector and y can also be a vector. However, in multi-class classification

problems, y can be a label, which is a scalar, while x is a vector. Alternatively, both x and y can be scalars. In most classification problems, Y is a scalar label, and x is a feature vector. This is a crucial detail to consider: Y can be a scalar in label associations, and x can be a feature vector.

Our goal is to estimate f accurately from the dataset. Typically, f is unknown, and we only have the system that provides values of y without knowing the explicit form of this unknown function.

(Refer Slide Time: 08:14)



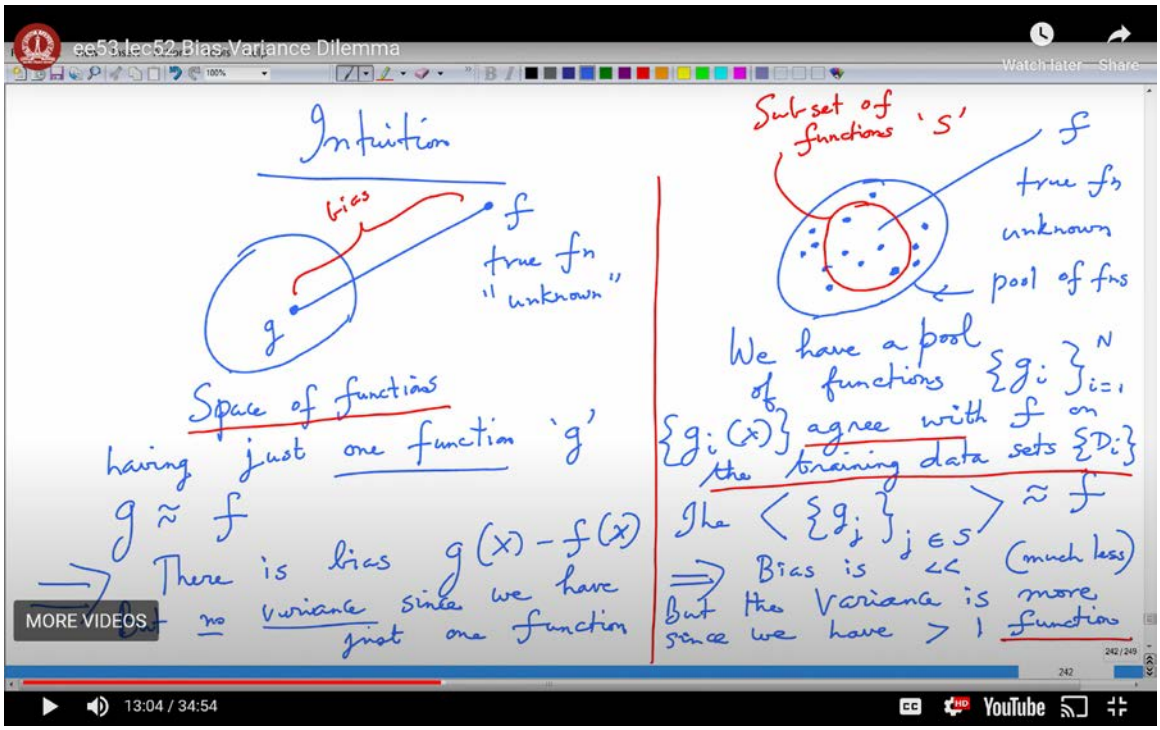
So, how do we approach this problem? Given a dataset D, we seek to determine a function G that is as close as possible to the unknown function f in some defined sense. What do we mean by "close"? We need to establish a performance metric, such as mean square error, likelihood function, or other relevant functions that measure how well G approximates f. This performance metric is crucial for assessing the quality of our approximation.

At this stage, you might wonder if we only have one dataset. Typically, that's not the case. If we are sampling, we can create multiple datasets, each with its own distinct

characteristics. This means that the function g we derive is dependent on the specific dataset we use. Let's consider different datasets, denoted as D_i, where i ranges from 1 to n. For each dataset, we can generate various estimates of the unknown function f. This setup should now be clear.

Since we cannot sample the entire function due to its potentially infinite nature, and because we lack the storage space for an infinite number of values, we sample at specific points and conditions, recording these datasets accordingly.

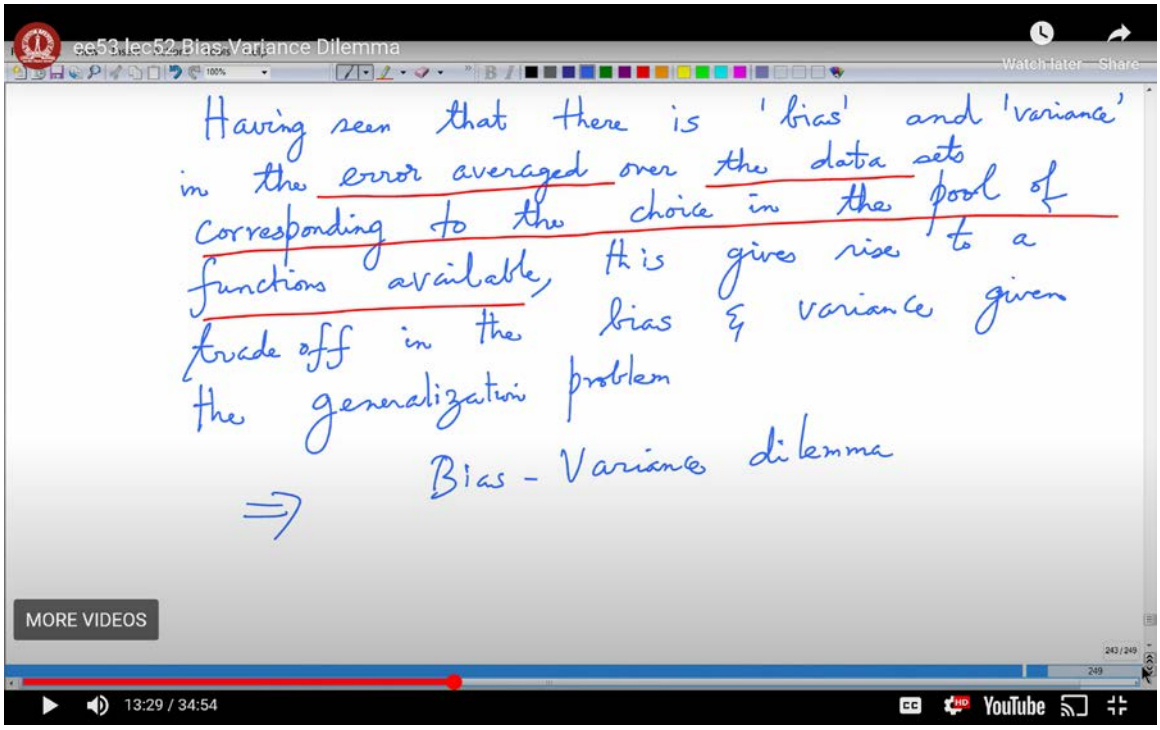
(Refer Slide Time: 13:04)



Now, let’s delve into an example to illustrate this concept further. Imagine an association between feature vectors and different animals, like cats. We have various types of cats, each represented by feature vectors. We sample datasets, labeling them accordingly, some as cats and some as not. However, these samples are not exhaustive. We have a range of feature vectors associated with different cat families. This subtlety is crucial when discussing function sampling and dataset creation.

To visualize this better, let's consider a parabola represented by a blue curve. Suppose we are given only two points from this curve. For example, let's say we have dataset D_1 with two points on the left side of the curve, dataset D_2 with two points on the right side, and dataset D_3 with two points near the bottom of the curve.

(Refer Slide Time: 13:29)



If we need to fit a line of the form y = mx + c to these datasets, what lines can we derive? For dataset D_1, we would simply connect the two points with a straight line. From this, we can determine the slope m and intercept c using coordinate geometry, resulting in line L_1. Similarly, for dataset D_2, we get line L_2, and for D_3, we get line L_3.

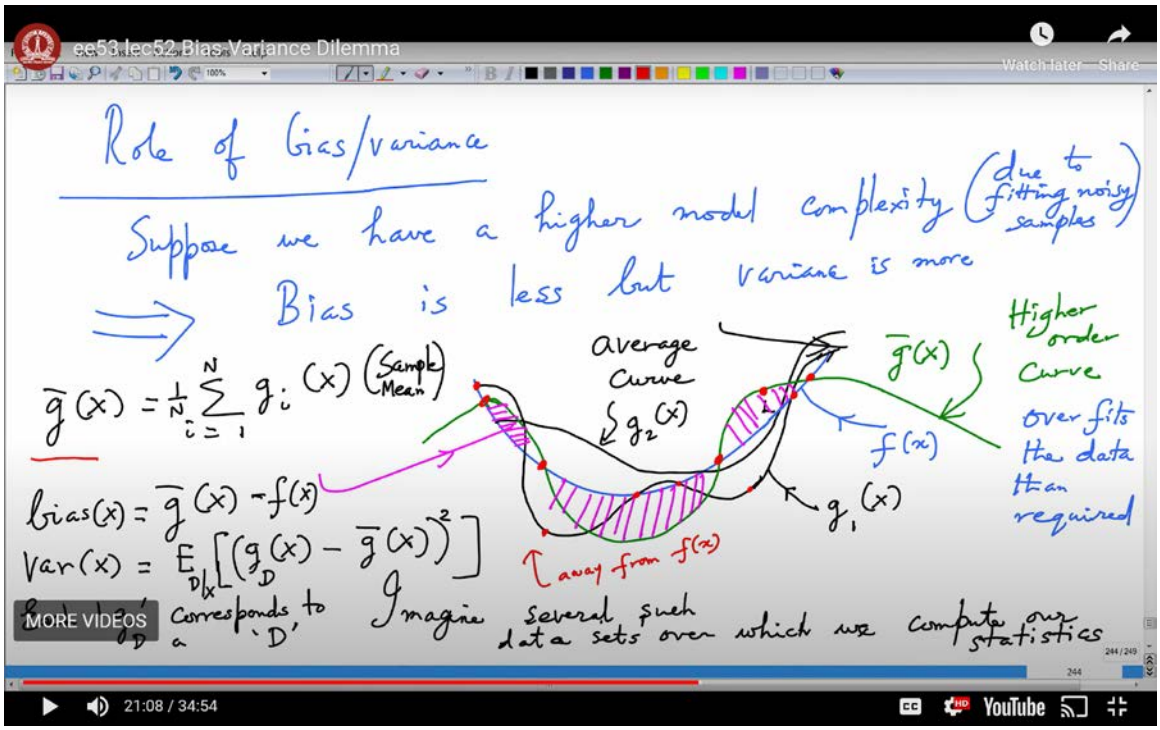
However, if our goal is to fit a line that represents the parabola as closely as possible using just one variable, a scalar, we need to fit a model with two parameters, the slope and intercept, to the given datasets. Instead of fitting a line to the data, we might want a point that best represents the parabola's overall shape.

So, what do we do with this information, and how can we assess the quality of our fit? These are fundamental questions in regression analysis. If we consider increasing the

model complexity, from a single scalar to a line, then to polynomial functions of higher degrees such as quadratic, cubic, quartic, and so on up to an n-th order polynomial, we need to explore the trade-offs associated with this complexity.

When discussing regression, it's natural to question whether increasing model complexity improves our ability to approximate the true function. Let's delve deeper into this issue to build a clearer intuition for our analysis.

(Refer Slide Time: 21:08)



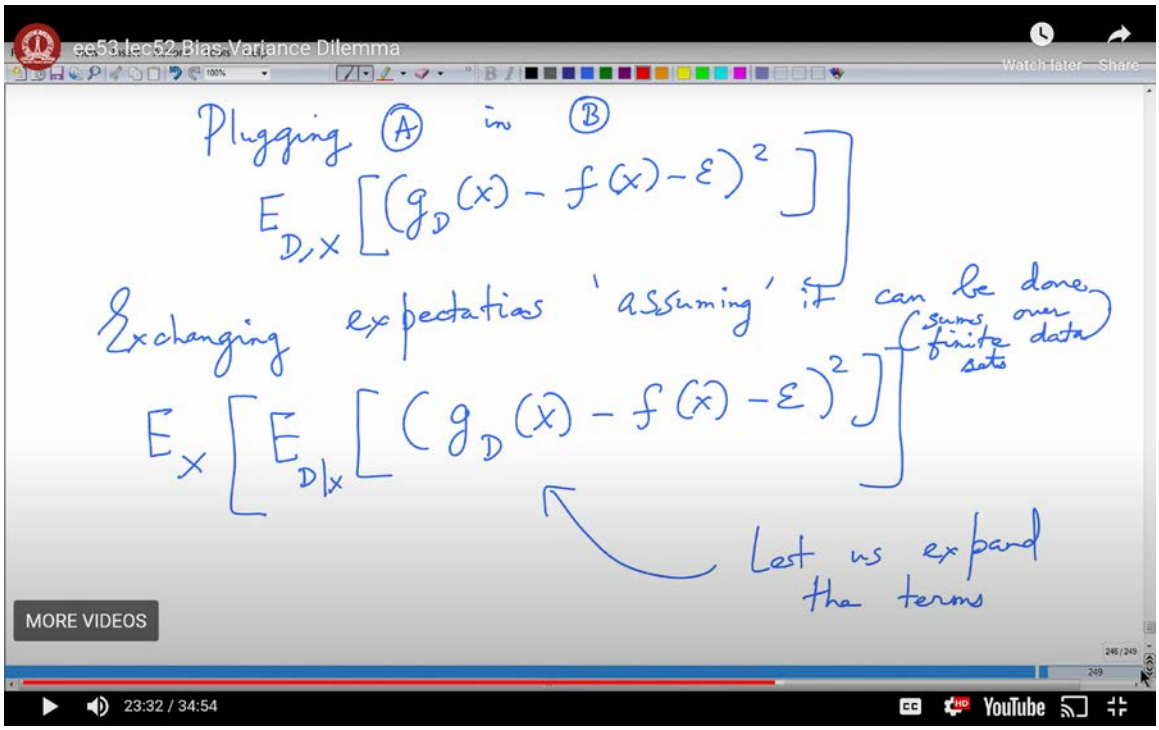
Imagine a space of functions where we have only one function g acting on a dataset. Here, the true function f is unknown. If we compute the deviation g(x) - f(x) (noting that f is unknown), our approximation g is a function of x based on the data points. This deviation represents the bias we have. With only one function g, there is no variance because there's no variation in our function set.

Now, let's extend this scenario to a situation where we have a pool of functions g_i, with i ranging from 1 to N. Each g_i agrees with the true function f on its respective training dataset D_i, where i ranges from 1 to N. While we might not have space to elaborate fully, the idea

follows straightforwardly. For each dataset D_i, there is a corresponding function g_i that approximates the true function f.

Within this pool of functions, there might be a subset S of functions that are particularly close to the actual function f. It's important to understand that each g_i depends on its dataset D_i. Just like in the parabola example, where different sampling yields different lines, each function g_i will differ based on its dataset D_i. Therefore, while all functions in S are close to f, the variance can be higher because we now have multiple functions to account for. With only one function, there was no variance, but with a pool of functions, variance naturally arises and depends on x.

(Refer Slide Time: 23:32)



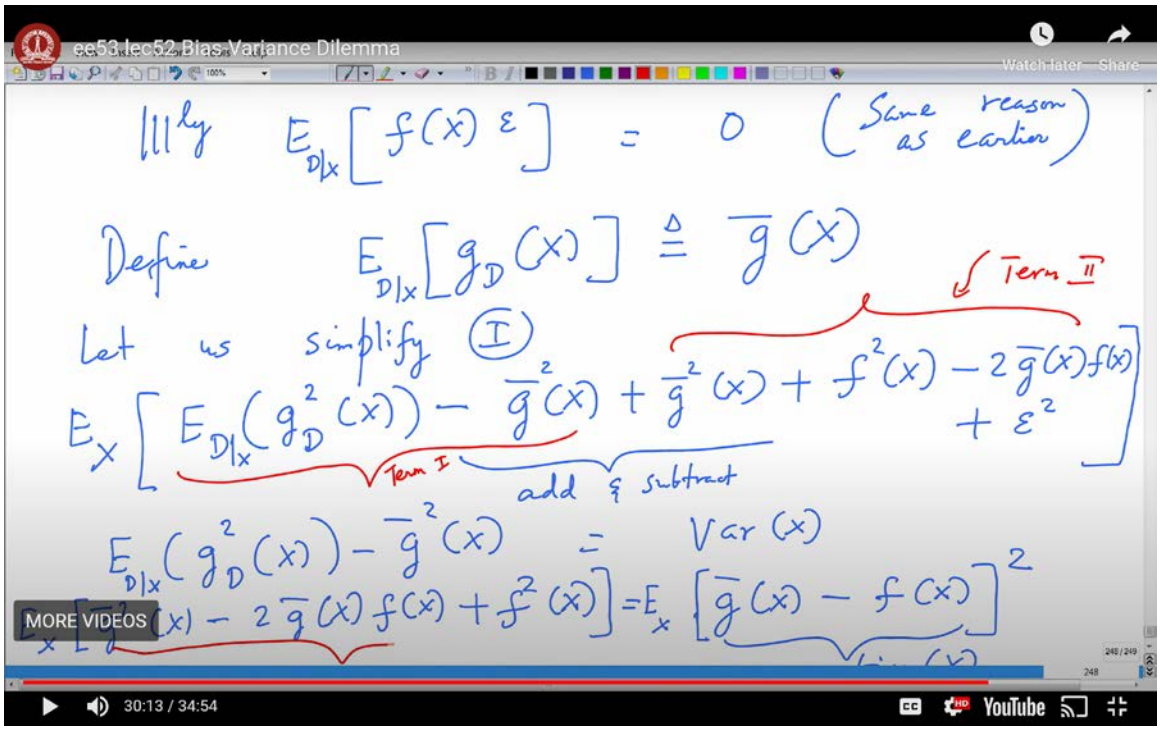
Given the presence of both bias and variance, we can compute the error averaged over the datasets. This error represents the deviation from the true unknown function. By calculating the statistics, specifically, the bias and variance, we can gain insights into the trade-off between these two factors. Initially, with a single function, we have some bias but no variance. With a pool of functions, we achieve lower bias because the functions are closer

to the true function on average, but we introduce variance because of the diversity in the functions. Thus, understanding this trade-off is key to optimizing our model.

There is indeed a trade-off between bias and variance, which is known as the bias-variance dilemma in regression problems. So, how can we analyze this trade-off further, and what strategies can we use for our analysis?

To deepen our understanding of bias and variance, let's revisit the example of sampling a parabola. When we had just two points and tried to fit lines through them, the resulting lines depended on the specific data points we sampled. Now, if we increase the model complexity, the scenario changes. For instance, if we had noiseless data, a second-order polynomial (a parabola) would perfectly fit the points. We could determine the coefficients A, B, and C to fit $Ax^2 + Bx + C$ exactly.

(Refer Slide Time: 30:13)



However, in practice, our measurements are invariably subject to noise, which is a fundamental aspect to consider. Given noisy samples, increasing the model order can allow us to fit the curve through these noisy points. As we move from a second-degree

polynomial to a third-degree, fourth-degree, and so on, the polynomial curve can accommodate noise better. For example, a fourth-degree polynomial might fit the noisy data points well, but it might overfit the model because the true function is quadratic. In this case, fitting a fourth-degree polynomial captures the noise rather than just the underlying trend.

To visualize this, imagine a parabola representing the true function f(x). The red circles on this curve represent the noisy sample points. These points might deviate from the true curve. For instance, some points might be located away from the curve, illustrating the presence of noise in our data. Given these noisy points, we can fit a curve $g_1(x)$ to these points based on the model's complexity. This curve, however, might also fit the noise. Similarly, if we choose a different set of sample data points, D_2, we can fit another curve $g_2(x)$ through these new points.

We can also compute an average curve, which is the mean of all the curves $g_1(x)$, $g_2(x)$, $g_3(x)$, up to $g_N(x)$. This average curve, denoted as $\bar{g}(x)$, represents the mean of these curves and helps smooth out the noise.

At this point, you might wonder whether to use a sample mean or perform a statistical average over a distribution of these curves. We will discuss this expectation and its implications later in the lecture.

At the moment, let's assume that $\bar{g}(x)$ is computed using the sample mean. However, there is no strict rule that prevents us from calculating the expectation over all possible datasets to find an average curve. We can define two important quantities here: the bias and the variance.

The bias of x is given by the deviation of the average curve $\bar{g}(x)$ from the true function f(x). Mathematically, this is expressed as:

$$\text{Bias}(x) = \bar{g}(x) - f(x)$$

This quantity measures how far our average fit deviates from the true function f(x), which remains unknown.

The variance of x is defined as:

$$\text{Variance}(x) = E\left[\left(g_D(x) - \bar{g}(x)\right)^2\right]$$

Here, $g_D(x)$ denotes the curve fitted to a particular dataset D, and $\bar{g}(x)$ is the average curve. This quantity represents the average squared deviation of the curves $g_D(x)$ from the average curve $\bar{g}(x)$ over all datasets D given x. Essentially, this variance captures the extent of fluctuation of our model's predictions due to different datasets.

In this context, we aim to explore the bias-variance dilemma, essentially, the trade-off where lower bias might come with higher variance, and vice versa. If the data is noisy, the variance increases because noise adds a squared positive quantity to the error. Thus, the squared error, when averaged over datasets and data points, tends to be larger due to this added noise.

Now, let's work through an analysis of the regression problem. Suppose we have:

$$y = f(x) + \epsilon,$$

where f is an unknown function, x and y could be vectors, but for simplicity, let's assume x is a vector and f(x) is a scalar-valued function yielding y. Typically, in classification problems, y could be a label, making it intuitive for y to be a scalar. However, y does not have to be restricted to scalars.

Here, ϵ represents the regression noise, a random variable with zero mean and variance σ^2, which is statistically independent of f and the approximating function g(x), which depends on the dataset D. We use the subscript D to indicate that g is a function of the dataset D, and averaging this gives us an average curve.

We are interested in the expectation of the squared error between $g_D(x)$ and y. Specifically, we evaluate:

$$E[(g_D(x) - y)^2]$$

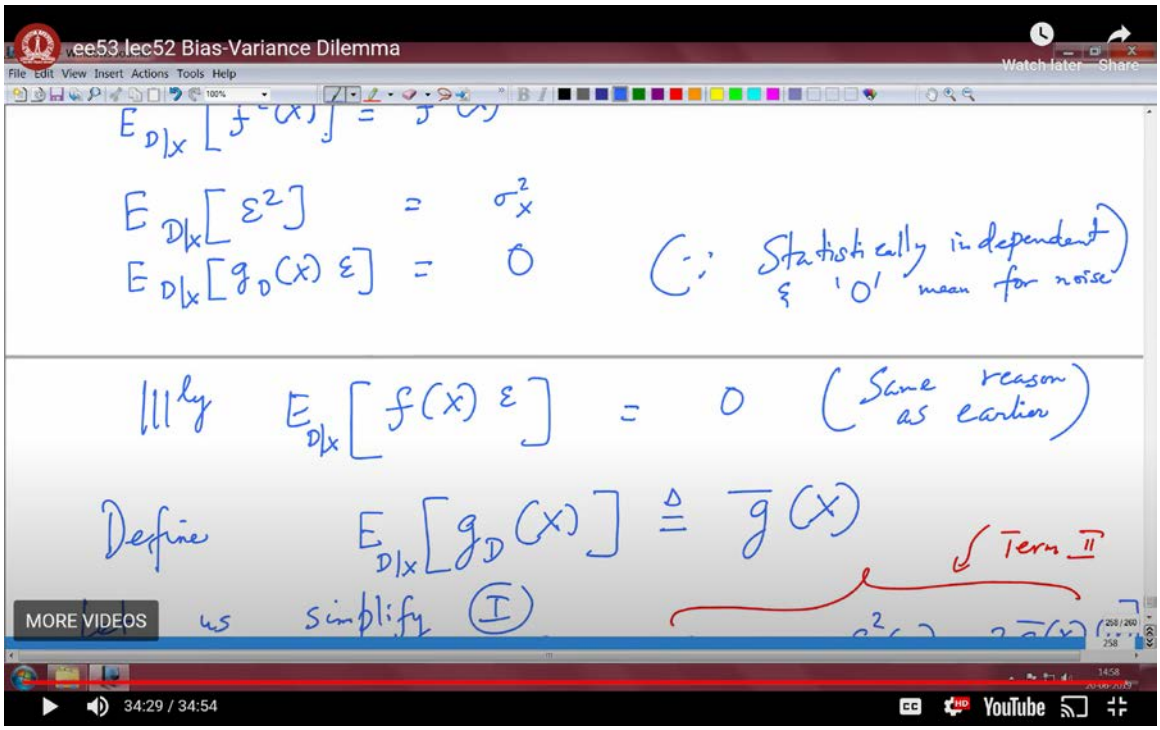
where y = f(x) + ϵ. We need to average this squared error over all datasets and all data points. Given the joint distribution of x and D, we have:

$$E[(g_D(x) - f(x) - \epsilon)^2]$$

If the datasets D and data points x are statistically independent, this expression factors into the product of their marginal distributions:

$$E[(g_D(x) - f(x) - \epsilon)^2] = E_D\left[E_x[(g_D(x) - f(x) - \epsilon)^2]\right]$$

(Refer Slide Time: 34:29)



You can exchange the order of expectations under the assumption that the data sets are finite and the exchange is valid. The calculation proceeds by fixing x and averaging over all datasets, and then averaging the squared error, which involves expanding the terms inside the brackets.

To simplify the expression, we start by expanding terms using the algebraic identity for squaring a binomial:

$$(a+b+c)^2 = a^2 + b^2 + c^2 + 2ab + 2bc + 2ca$$

Applying this to our context, we have:

$$g_D^2(x) + f^2(x) + \epsilon^2 - 2 \cdot g_D(x) \cdot f(x) - 2 \cdot g_D(x) \cdot \epsilon + 2 \cdot f(x) \cdot \epsilon,$$

Here, $g_D(x)$ represents the model fit to the dataset D, f(x) is the true function, and ϵ is the noise term. We need to average this expression over both x and D. To simplify, we can decompose this averaging process into manageable parts. For convenience, let's consider averaging over x first and then D, or vice versa.

Since expectation is a linear operator, we can evaluate each term separately:

- The expectation of $f^2(x)$ given x is simply $f^2(x)$, as it is a function of x alone.
- The expectation of ϵ^2 given x is σ^2, as noise is consistent across datasets and has the same variance.
- The expectation of $g_D(x) \cdot \epsilon$ given x is 0 because the noise ϵ is statistically independent of $g_D(x)$ and has zero mean.
- Similarly, the expectation of $f(x) \cdot \epsilon$ given x is also 0 for the same reasons.

Next, we define $\bar{g}(x)$ as the average curve fit across all datasets for a given x:

$$\bar{g}(x) = E_{D|x[g_D(x)]}$$

Now, simplifying the terms, we add and subtract $\bar{g}^2(x)$:

$$E_{D|x[g_D^2(x)]} = E_{D|x\left[\left(g_D(x) - \bar{g}(x)\right)^2\right]} + \overline{g^2}(x)$$

The first term on the right is the variance of $g_D(x)$, which we denote as:

$$\text{Variance}(x) = E_{D|x\left[\left(g_D(x) - \bar{g}(x)\right)^2\right]}$$

The simplified expression for the squared error then becomes:

$$f^2(x) - 2 \cdot \bar{g}(x) \cdot f(x) + \epsilon^2$$

Expanding this term, we have:

$$g_D^2(x) = \text{Variance}(x) + \overline{g^2}(x)$$

Thus, when comparing $g_D^2(x)$ and $\overline{g^2}(x)$, we find:

$$\text{Term 1} = \text{Variance}(x)$$

$$\text{Term 2} = g_D^2(x) - \overline{g^2}(x) - 2 \cdot \bar{g}(x) \cdot f(x) + f^2(x)$$

Using the identity for squaring a binomial, this simplifies to:

$$\text{Term 2} = \left(g_D(x) - \bar{g}(x) - f(x)\right)^2$$

So, the squared bias is:

$$\text{Bias}(x) = \bar{g}(x) - f(x)$$

Finally, the expectation of this squared bias over x is computed to give the overall bias in our model.

In conclusion, we arrive at a final expression for the error in regression, which comprises three key components: variance, bias, and the noise term, denoted by σ^2.

Initially, the variance is computed based on the conditional expectation of D given x. Since this variance is a function of x, averaging it over all possible x yields a variance term that depends on the distribution of x. Similarly, the bias, when averaged over all x, provides an additional term in the error analysis. The term σ^2 represents the noise statistic, averaged over both x and D, and reflects the inherent variability in the data.

So, the total error that we compute, averaged over the data points x and the datasets D, consists of these three components: the bias, the variance, and σ^2.

When dealing with this error, if it is fixed, increasing bias will generally reduce variance, and vice versa. This tradeoff is crucial in regression problems, where the goal is to find a

balance that minimizes both bias and variance. Depending on how you fit your model and the functions you choose, this balance may vary.

This bias-variance tradeoff is a fundamental dilemma in regression analysis. While this analysis was conducted with the assumption of constant noise variance across datasets, this assumption is not strictly necessary. If ϵ^2 varies across datasets, we can adjust our calculations accordingly. Specifically, when we average ϵ^2 over all datasets given x, we obtain a variance term that depends on x, and averaging this over all x yields σ^2, consistent with our previous results.

Thus, it's important to be aware that σ^2 does not need to be constant across datasets. Instead, it should be understood as σ^2_x when conditioned on x, and when averaged over all x, it converges to σ^2.

I hope this explanation clarifies the details of the bias-variance dilemma. As you tackle regression problems, carefully consider the balance between bias and variance to optimize your model's performance. This ends our lecture on the topic.