Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

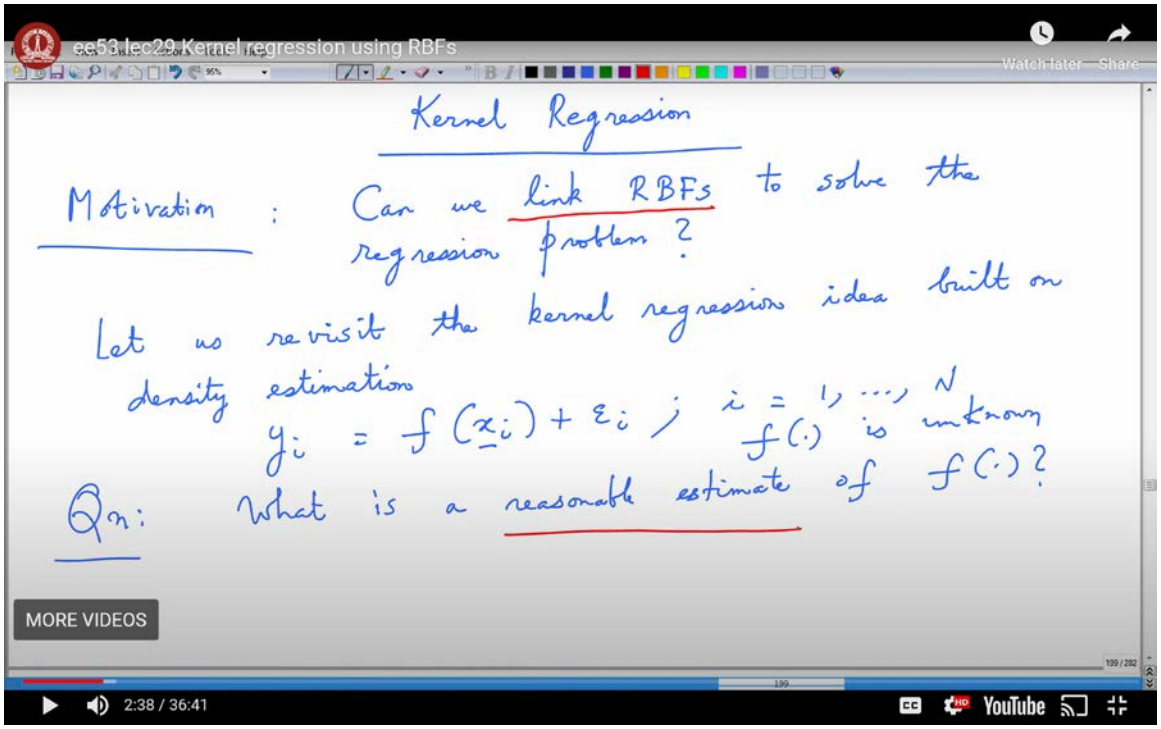
Indian Institute of Science, Bengaluru

Lecture – 29

Kernel Regression using RBFs

Let's dive into the process of estimating the probability density using kernels and explore how this concept applies to kernel regression problems. This approach will also demonstrate how radial basis functions (RBFs) can be effectively utilized in kernel regression. Essentially, we're linking RBFs to solve the regression problem, which serves as a practical application of these functions.

(Refer Slide Time: 02:38)



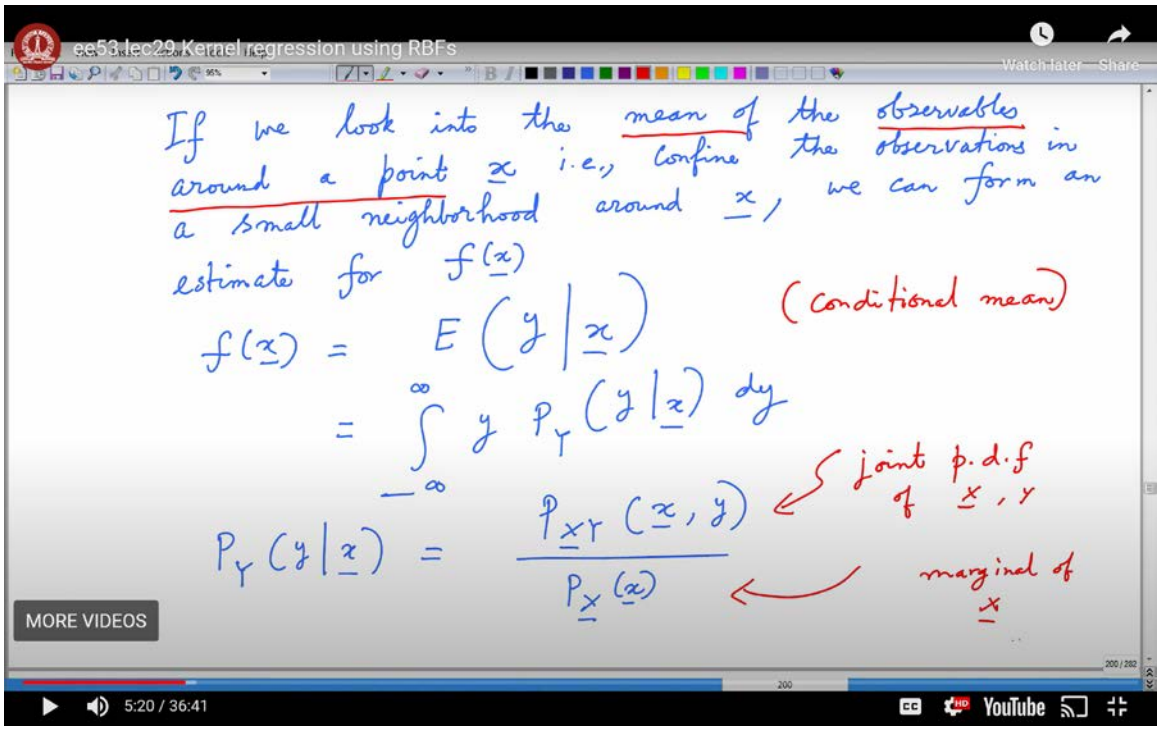
To set the stage, let's revisit the kernel regression concept that we discussed in previous modules. Consider a scenario where the observable y_i is given by the equation $y_i = f(x_i) +$

ϵ_i, where i ranges from 1 to N, representing the number of samples. Here, f is an unknown function, ϵ_i is the random noise associated with the sample x_i, and we assume that this noise is statistically independent and identically distributed across the samples.

In this setup, y represents the observable quantity, while x is a vector corresponding to our data samples. Each pair (x_i, y_i) represents a data sample, where y_i can also be a vector in a more general context. However, since we often deal with classification problems, y_i typically represents a label associated with the training vector x_i. Therefore, we can think of f as a mapping from some m-dimensional space (corresponding to x_i) to a one-dimensional real space, which corresponds to the observable y_i.

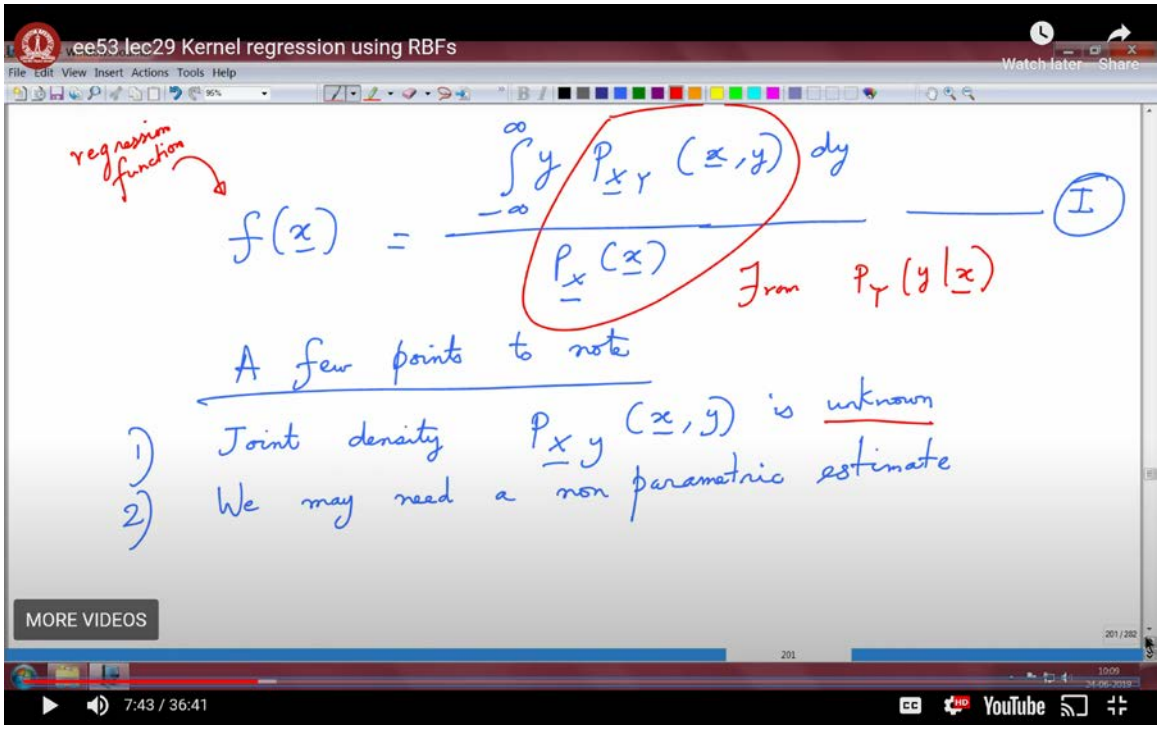
Now, the critical question arises: what constitutes a reasonable estimate of f? To approach this, we can draw from basic statistical principles. If you are given a set of observables y and asked to determine a reasonable estimate for y, which corresponds to f(x), common sense suggests that we should consider the mean. If we're seeking a single value to represent this, calculating the mean serves as a reasonable estimate.

(Refer Slide Time: 05:20)



In this context, we focus on the mean of the observables around the point x. Given the point x, we can examine the observation space within a small neighborhood around x to form an estimate of f(x). Essentially, f(x) represents the expected value of y given x, that is, it is the conditional mean of the observables y given the vector x.

(Refer Slide Time: 07:43)



We can compute this conditional mean by integrating over all possible values of y from minus infinity to plus infinity, multiplying y by the conditional density function P(y|x). This integral gives us a reasonable estimate for f(x).

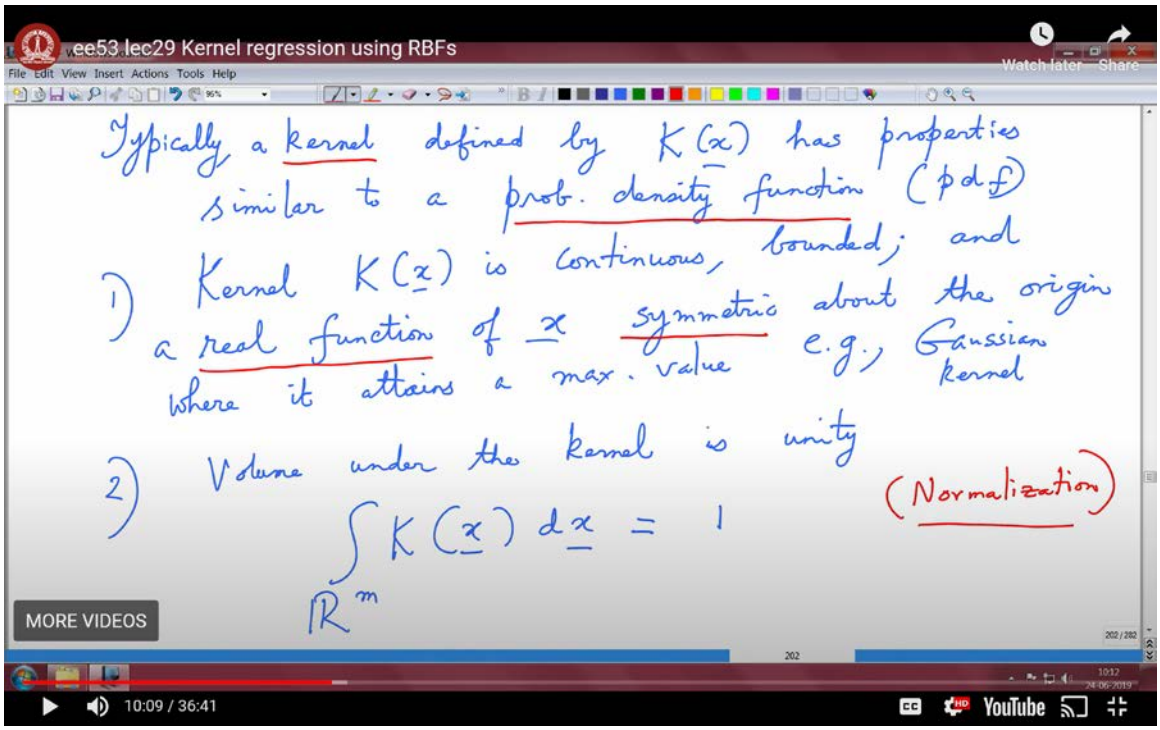
To compute P(y|x), we can utilize the joint probability density function of x and y, as well as the marginal probability density of x. Here, x is treated as a random vector, while y is a scalar. By integrating over these densities, we can arrive at a reasonable estimate for the unknown function f(x).

Let's delve into the concept of the regression function, denoted as f(x). It's crucial to understand that in this context, f(x) always refers to the regression function. On the other hand, when we use the notation p(x) with an underscore, it represents the probability

density. This distinction is vital, which is why we use different notations, p for the probability density and f for the unknown regression function.

Now, the numerator in our formulation can be computed using the conditional mean, as we've discussed. This is divided by p(x), which represents the probability density at x. However, let's correct a potential confusion here, p(x) is not the density in this particular context. Instead, what we need to focus on is the integral over the entire space, which brings us to the correct formulation.

(Refer Slide Time: 10:09)



So, let's clarify: f(x) is our regression function, and we compute the integral from minus infinity to plus infinity using the conditional density p(y | x). This conditional density is crucial because it links the observable y given x to our regression function. By substituting p(y | x) using the joint and marginal densities, we can express the integral more clearly.

However, there are a couple of important points to note. First, the joint density is often unknown because we don't have an explicit closed-form expression for it. Second, we may need to rely on non-parametric estimates to proceed, especially when the joint density is

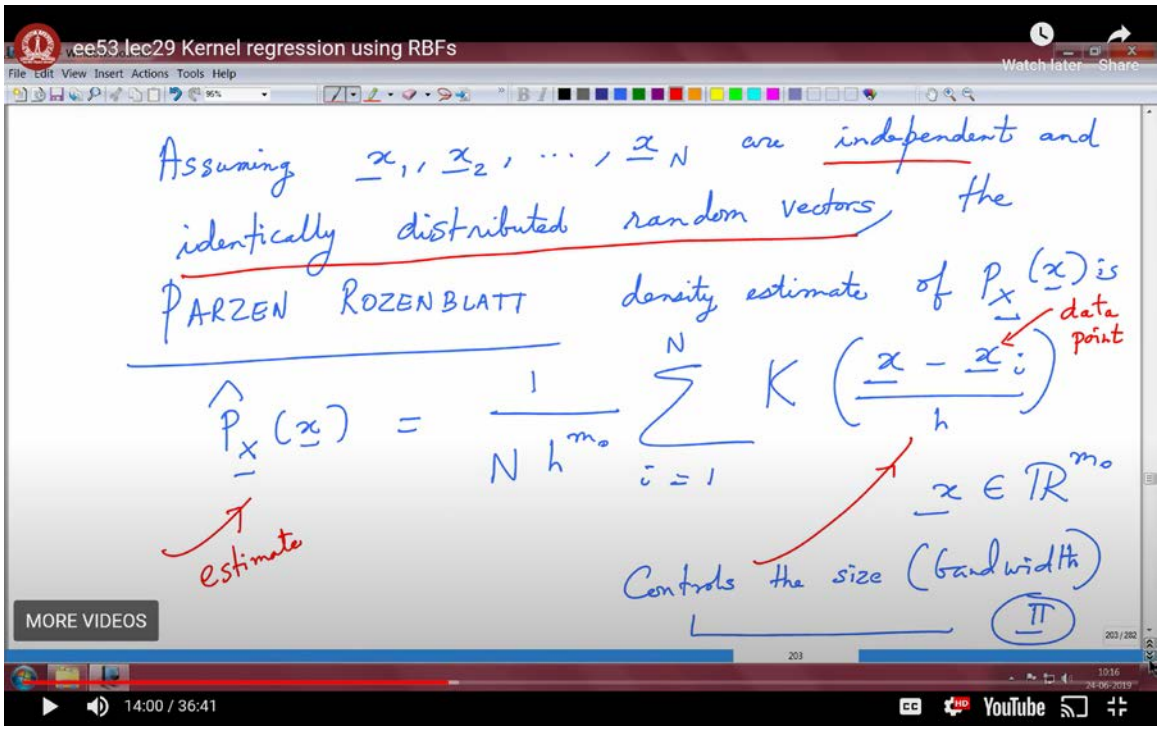
not explicitly known. This brings us to the next step: simplifying the problem using kernel density estimation.

To tackle this, we introduce the idea of kernel density estimation. But first, what exactly is a kernel? A kernel, denoted as k(x), has properties similar to a probability density function. There are two fundamental properties that any kernel must satisfy:

1. Real-Valued and Symmetric: The kernel must be a real-valued function of x and symmetric around the origin. Typically, it should also attain its maximum value at the origin.

2. Normalization: The area under the curve of the kernel, or more accurately, the volume under its surface, must sum to unity. This property ensures that the kernel behaves like a probability density function.

(Refer Slide Time: 14:00)

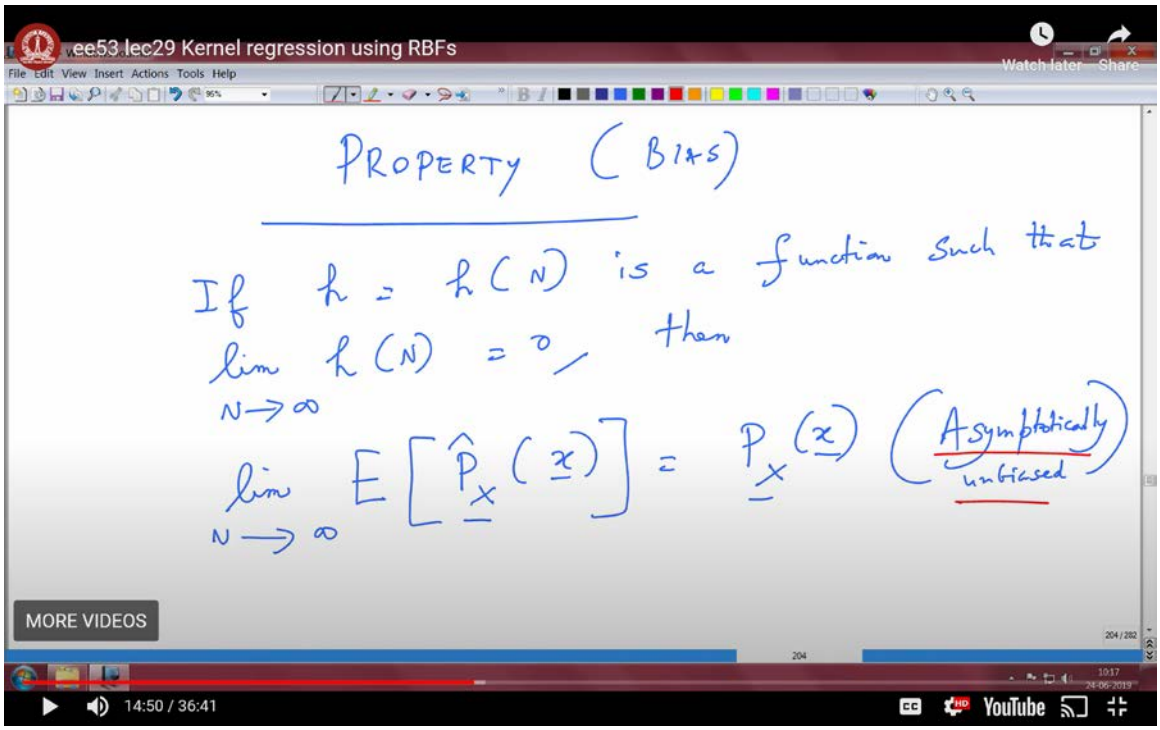


Additionally, we want our kernels to be continuous and bounded for them to be well-behaved. Simply being real and symmetric about the origin isn't always sufficient. Some

classic examples of well-behaved kernels include the Gaussian kernel, cosine kernel, and parabolic kernel. Even a triangle-shaped function can serve as a kernel, although it may lack smoothness at its peak.

Thus, for a function to be a good kernel, it must exhibit symmetry and normalization, akin to the properties of a probability density function. All functions that meet these criteria can be considered suitable kernels. The efficiency of one kernel compared to a reference kernel can be evaluated based on bandwidth and certain statistical properties.

(Refer Slide Time: 14:50)



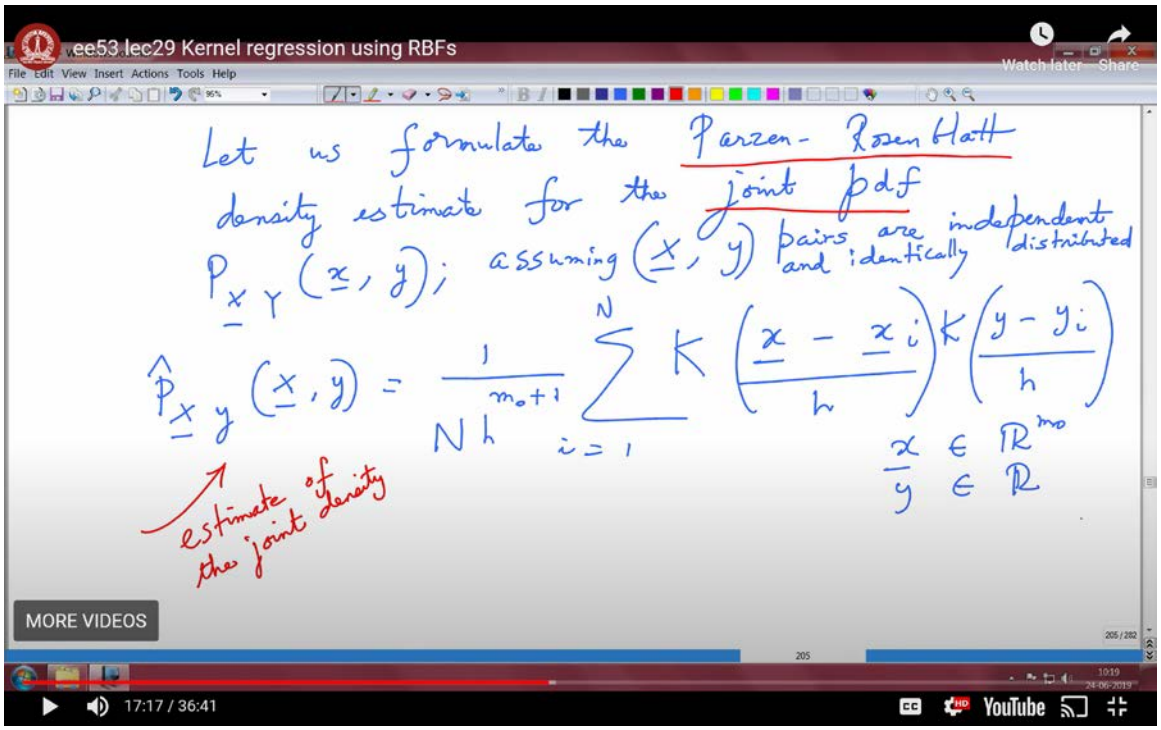
Now, let's assume we have training samples x_1, x_2, …, x_n. These samples are assumed to be independent and identically distributed (i.i.d.) random vectors. When sampling data points from a process, we treat them as random vectors because the presence of observation noise typically introduces randomness. Even the feature vectors themselves can be random, depending on the nature of the data.

The Parsons-Rosenblatt density estimation of $p_x(x)$, denoted as $\widehat{p_x}(x)$, is essentially an estimate of the probability density function (PDF). It is given by the equation:

$$\widehat{p_x}(x) = \frac{1}{nh^{m_0}} \sum_{i=1}^{N} k\left(\frac{x - x_i}{h}\right)$$

Here, x_i represents the data points, and x is a continuous random variable. The parameter h is crucial as it controls the size of the kernel, which is commonly referred to as the bandwidth. Both x and x_i are vectors residing in an m_0-dimensional space, meaning that this PDF estimate is inherently multivariate. The parameter h is a scalar that directly influences the bandwidth of the kernel.

(Refer Slide Time: 17:17)



Now, a pertinent question arises: what is the role of h, and how do we determine its optimal value? Typically, h can be made as small as possible, leading to a more refined and detailed estimate. However, if h is large, the estimate becomes more accommodating, encompassing more data points. This introduces a natural bias-variance tradeoff in the estimation of the density. If h approaches a delta function, bias would be introduced, but the variance would diminish. On the other hand, incorporating some standard deviation σ would introduce

variance. Hence, the parameter h plays a crucial role in managing the bias-variance tradeoff.

An important property of this density estimation is related to the behavior of h as the number of samples n increases. Specifically, if h_n, a function of n, approaches 0 as n approaches infinity, the expected value of the density estimate converges to the true density p(x). In other words, if the limit:

$$\lim_{n\to\infty} E\left[\widehat{p_x}(x)\right] - p(x) = 0$$

is satisfied, then the estimate is asymptotically unbiased. For this condition to hold, h_n must diminish to 0 as n increases.

Now, let's move forward and apply the Parsons-Rosenblatt density estimation to formulate the joint PDF. Recall that to compute the conditional mean, we required knowledge of the conditional density, specifically the probability of Y given X. To derive this, we needed the joint density over X and Y, divided by the marginal density of X.

Let's explore how kernel density estimation can be used for this purpose. We assume that the pairs (x, y) are independent and identically distributed (i.i.d.). Under this assumption, the estimate of the joint density of x and y is given by:

$$\widehat{p_{x,y}}(x,y) = \frac{1}{nh^{m_0+1}} \sum_{i=1}^{N} k_x\left(\frac{x-x_i}{h}\right) k_y\left(\frac{y-y_i}{h}\right)$$

Here, the exponent $m_0 + 1$ appears because x is an m_0-dimensional vector, while y is a scalar in R. The dimension scaling is crucial, hence the inclusion of $m_0 + 1$ in the exponent. This summation captures the contribution of the kernel function acting on both x and y, providing an estimate of the joint density.

Let's delve into this process step by step. Consider the expression $\frac{x-x_i}{h}$ where the kernel function k acts on this, and similarly, k acts on $\frac{y-y_i}{h}$. Here, y is a scalar, y_i represents the observable data point, x_i is your feature vector, and x is a continuous variable. We can now

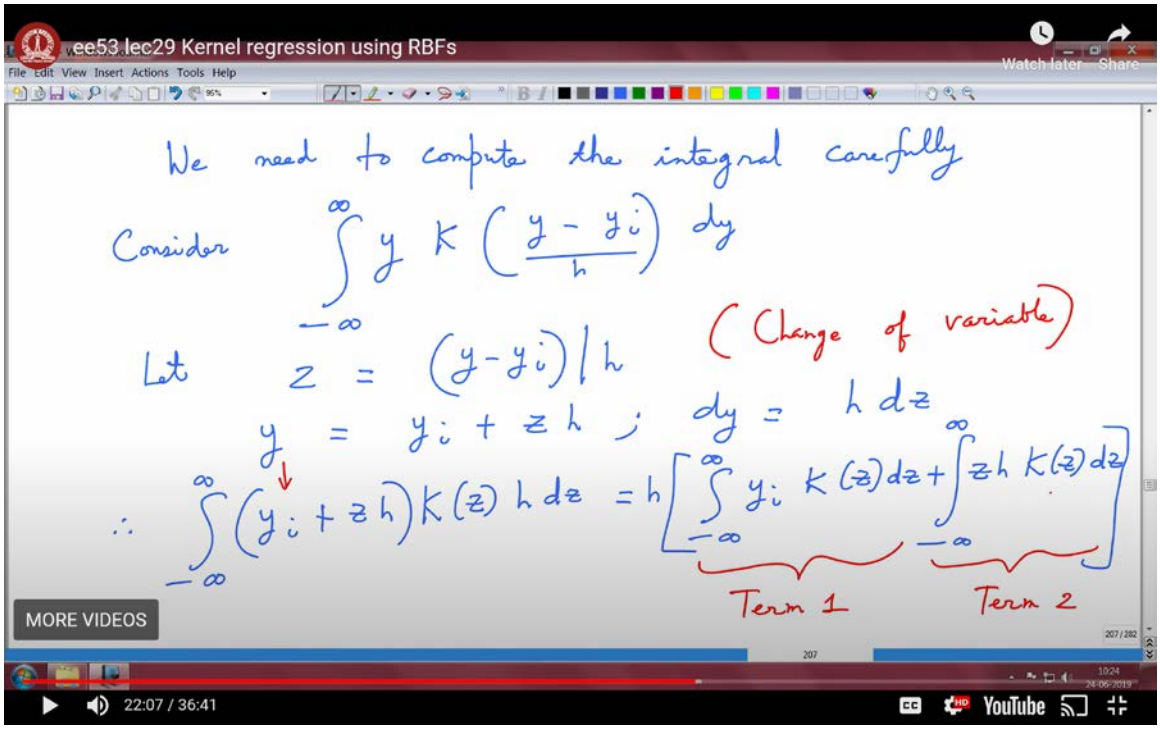
estimate the joint density in this form, and since we've already discussed how to estimate the marginal density, we're close to simplifying the problem at hand.

The next task is to compute the integral from minus infinity to plus infinity of y times the estimated joint density. This approach aligns with the starting point of our earlier equation, Equation 1. To recall, Equation 1 involved the integral of this quantity divided by P(x), the marginal density of x. Now, with the joint density estimate $P_{x,y}$, as derived in the previous step, we substitute it into this expression.

This simplifies to the integral calculation of the sum:

$$\frac{1}{n \cdot h^{m_0+1}} \sum_{i=1}^{N} k\left(\frac{x - x_i}{h}\right) \int_{-\infty}^{\infty} y \cdot k\left(\frac{y - y_i}{h}\right) dy$$

(Refer Slide Time: 22:07)



Although this expression still seems complex, we can simplify it further. To do this, let's define a new variable $z = \frac{y - y_i}{h}$. Consequently, y can be expressed as y_i + zh, and dy becomes $h \cdot dz$ since the derivative of y_i is 0.

Now, we substitute these into our integral:

$$\int_{-\infty}^{\infty} y \cdot k\left(\frac{y - y_i}{h}\right) dy = \int_{-\infty}^{\infty} (y_i + zh) \cdot k(z) h \, dz$$

We can factor out the h and expand the integral into two separate terms:

$$h \cdot \left[y_i \int_{-\infty}^{\infty} k(z) dz + h \int_{-\infty}^{\infty} z \cdot k(z) dz \right]$$

Here, the first term involves y_i multiplied by the integral of k(z) over all z, and the second term involves h multiplied by the integral of $z \cdot k(z)$.

Now, let's analyze these two terms:

1. Term 1: Normalization , y_i is a constant and can be pulled out of the integral. The integral of k(z) over all z is equal to 1 due to the normalization property of the kernel function.

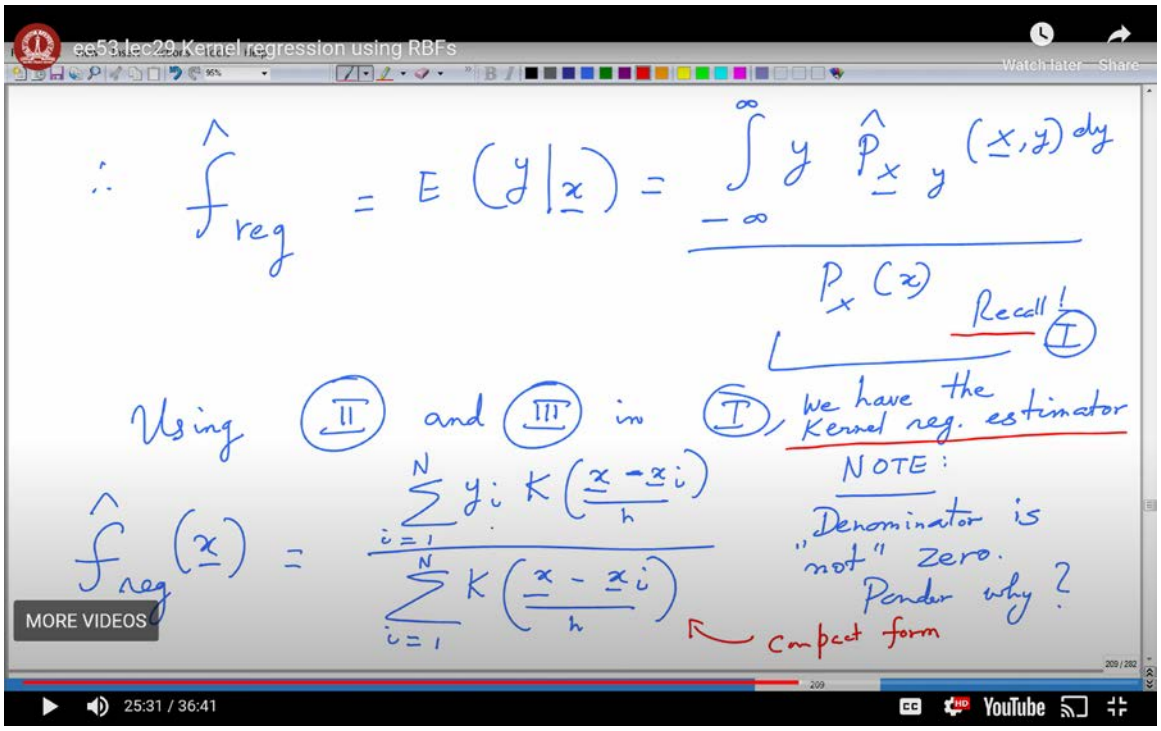
2. Term 2: Symmetricity , $z \cdot k(z)$ is symmetric about the origin, and because the kernel is symmetric, the mean of this kernel is 0. Therefore, the integral of $z \cdot k(z)$ evaluates to 0.

So, the first term simplifies to y_i, as the normalization property ensures the integral equals 1. The second term simplifies to 0 due to the symmetricity of the kernel, and we assume it has a zero mean.

So, this integral evaluates to y_i, owing to the kernel's normalization and symmetricity properties, and hence simplifies the process significantly.

Let's simplify the integral, which runs from minus infinity to plus infinity of y multiplied by the joint density. Here's how it breaks down: we have the term $\frac{1}{n \cdot h^{m_0+1}}$. After substituting the values, we find that Term 1 evaluates to 1, and Term 2 evaluates to 0. This leaves us with a scaling factor h that remains from Term 1.

(Refer Slide Time: 25:31)



Incorporating this scaling factor h and y, which also comes out of the integral, we obtain the following expression:

$$\frac{1}{n \cdot h^{m_0+1}} \cdot h \cdot \sum_{i=1}^{N} y_i \cdot k\left(\frac{x - x_i}{h}\right)$$

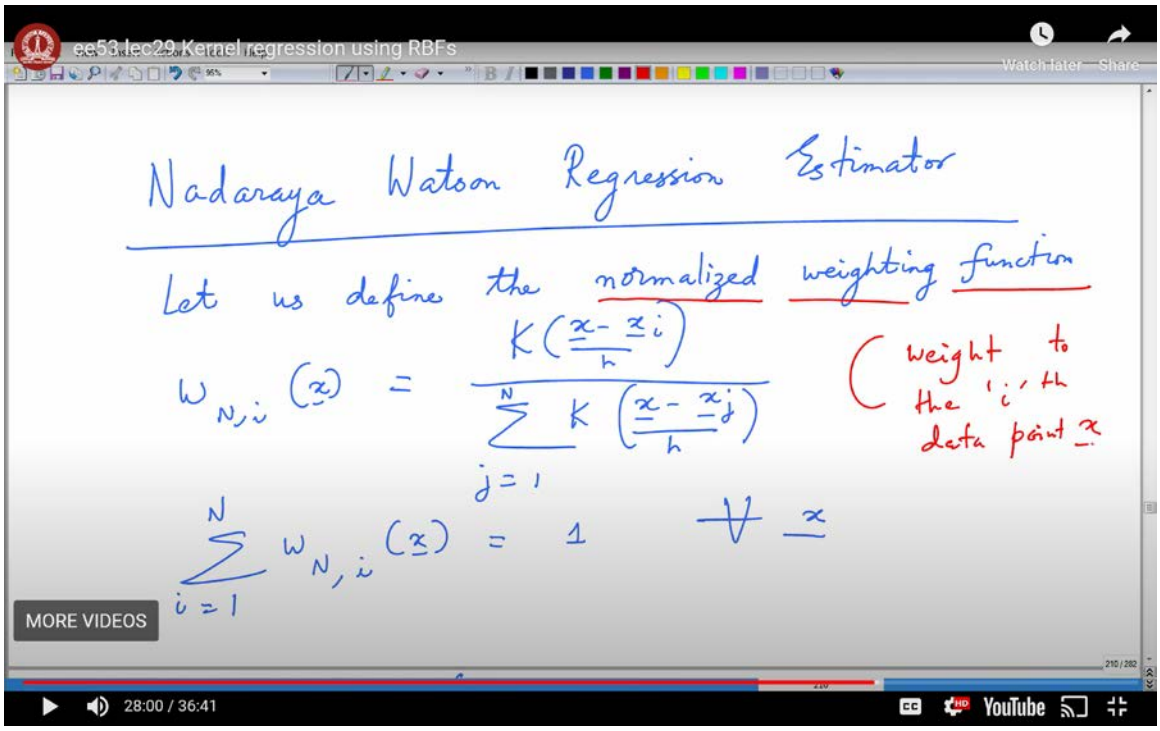
Here, x is a vector, and x_i represents the corresponding feature vectors. This formula represents our regression function, which is evaluated as the expected value of y given x, i.e., the conditional mean. By substituting the estimate for $P_{x,y}$ (the joint density) and P_x (the marginal density of x), we arrive at this compact form:

$$\frac{\sum_{i=1}^{N} y_i \cdot k\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^{N} k\left(\frac{x - x_i}{h}\right)}$$

This is the streamlined form of the regression function. It requires the data points x_i and their corresponding observables y_i. The denominator does not evaluate to zero because it

is a valid density function, meaning it is non-zero for the given points x and x_i. Thus, everything in the derived regression expression is valid.

(Refer Slide Time: 28:00)



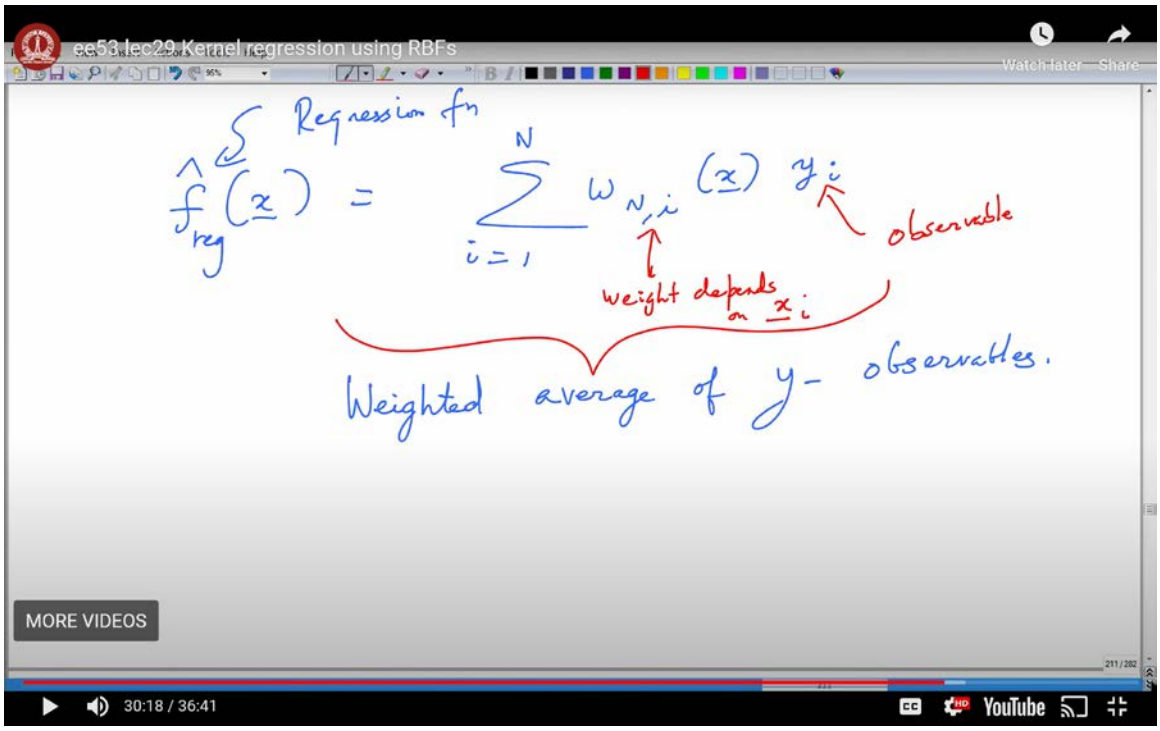
Additionally, there's a well-known estimator in the estimation theory community called the Adaraya-Watson regression estimator.

To define this, we introduce the normalized weighting function $W_{n,i}(x)$, where n represents the number of data points and i indexes the i-th sample. The normalized weighting function is given by:

$$W_{n,i}(x) = \frac{k\left(\frac{x - x_i}{h}\right)}{\sum_{j=1}^{n} k\left(\frac{x - x_j}{h}\right)}$$

You can straightforwardly verify that the sum of $W_{n,i}(x)$ over all i equals 1. This is because the denominator is constant, and summing the numerator and denominator across all i cancels out, leading to normalization. Hence, it's called a normalized weighting function, which assigns a weight to the i-th data point x_i.

(Refer Slide Time: 30:18)



Once we have the normalized weighting function, we can compute the regression function. The regression function $f_{\text{reg}}(x)$ is given by:

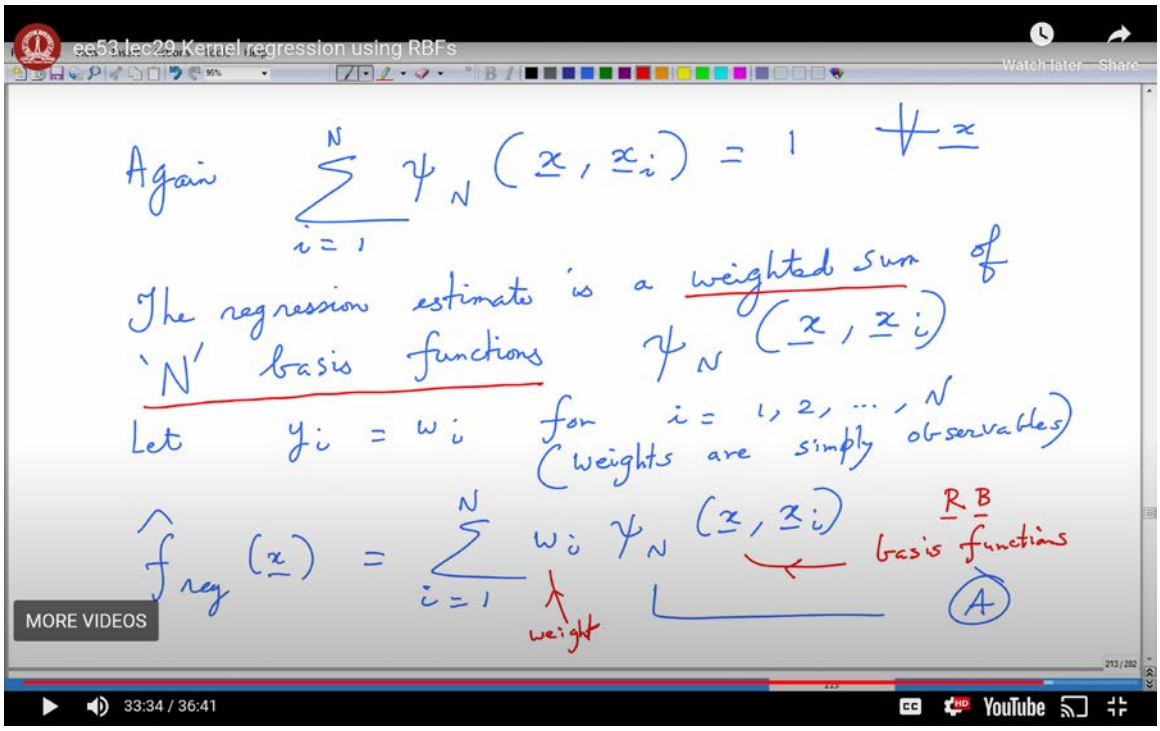
$$f_{\text{reg}}(x) = \sum_{i=1}^{N} y_i \cdot W_{n,i}(x)$$

Here, y_i is the observable, and $W_{n,i}(x)$ is the weighting function associated with that observable, depending on the data point x_i. We use both x_i and y_i, each data point comprises a vector x_i and its corresponding observable y_i. By applying the kernel function, we formulate the regression estimate.

So far, I've introduced the concept of using kernels to estimate regression functions, but you might be wondering about the role of radial basis functions (RBFs) and why we're discussing them. Radial basis functions often use Gaussian kernels, though other kernels can be employed as well. Gaussian kernels are particularly notable because they exhibit

radial symmetry. This radial symmetry makes RBFs quite useful for regression problems, which we'll explore further shortly.

(Refer Slide Time: 33:34)



The kernel in an RBF has spherical symmetry. Thus, when evaluating the kernel function on $\frac{x-x_i}{h}$, it's equivalent to evaluating the kernel on $\frac{|x-x_i|}{h}$, where $|x - x_i|$ represents the Euclidean norm of the difference between the vectors x and x_i. This norm naturally provides the radial symmetry we need.

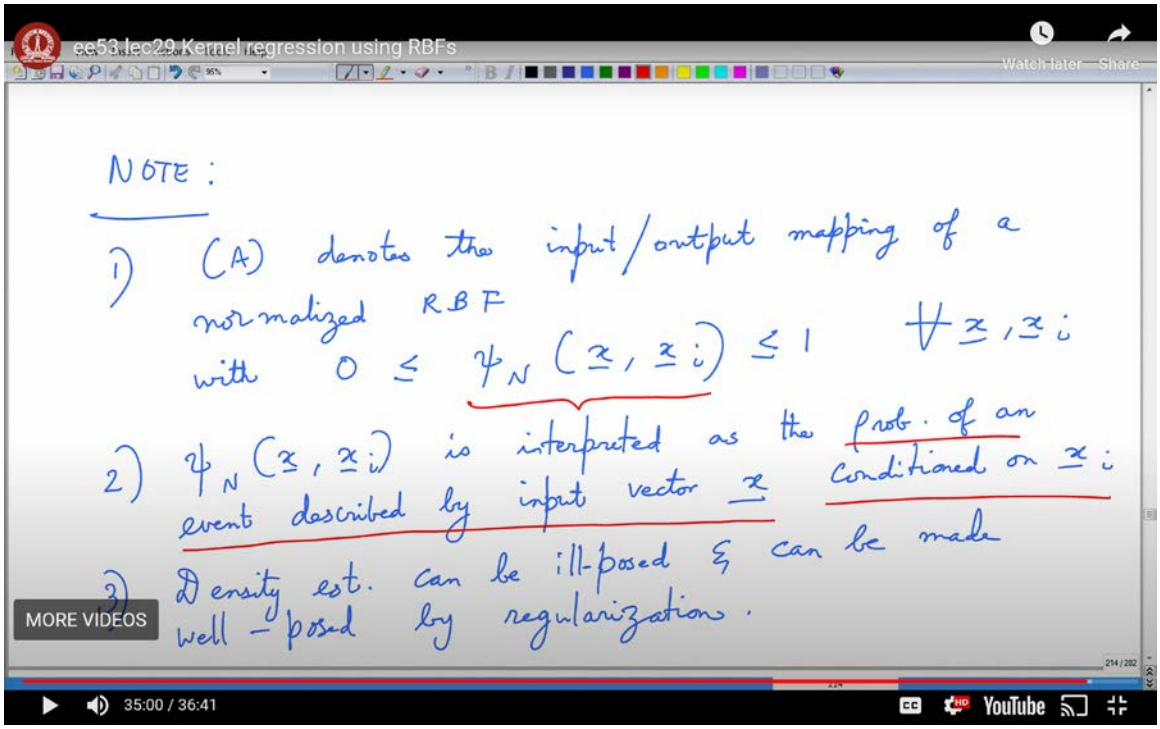
Let's define ψ_n similarly to how we defined the weighting function in previous discussions. Specifically, ψ_n is given by:

$$\psi_n(x) = \frac{k\left(\frac{|x - x_i|}{h}\right)}{\sum_{j=1}^{n} k\left(\frac{|x - x_j|}{h}\right)}$$

Here, $\psi_n(x)$ is akin to the normalized weighting function we discussed in the NWRE (non-parametric weighted regression estimator). When we plug this into our regression function, the normalized weighting function sums to 1, meaning:

$$\sum_{i=1}^{N} \psi_n(x) = 1$$

(Refer Slide Time: 35:00)



This ensures that the regression estimate is essentially a weighted sum of N basis functions defined by ψ_n. The regression function $f_{\text{reg}}(x)$ can be expressed as:

$$f_{\text{reg}}(x) = \sum_{i=1}^{N} w_i \cdot \psi_n(x)$$

Here, w_i represents the observable associated with the i-th data point, and ψ_n are our radial basis functions. Essentially, the regression function is a weighted sum of these basis functions, with the weights being the observables.

This completes our formulation of the regression estimation using radial basis functions. There are a few critical points to note:

1. Normalized Radial Basis Functions: The term $f_{\text{reg}}(x)$ represents the input-output mapping of a normalized radial basis function, where these functions range between 0 and 1. Importantly, $\psi_n(x)$ can be interpreted as the probability of an event described by the input vector x conditioned on x_i.

2. Ill-Posedness: It's important to understand that density estimation problems can sometimes be ill-posed. This raises questions about the uniqueness of the regression solution, the proper choice of weights, and whether our model is underfitting or overfitting. These are complex issues that we'll address in detail when we delve into regularization theory. Regularization helps constrain and optimize parameters to find a unique and effective solution.

We'll cover these concepts more thoroughly in our future discussions on regularization. For now, let's pause here and plan to revisit these topics as we explore regularization theory.