Neural Networks for Signal Processing-I

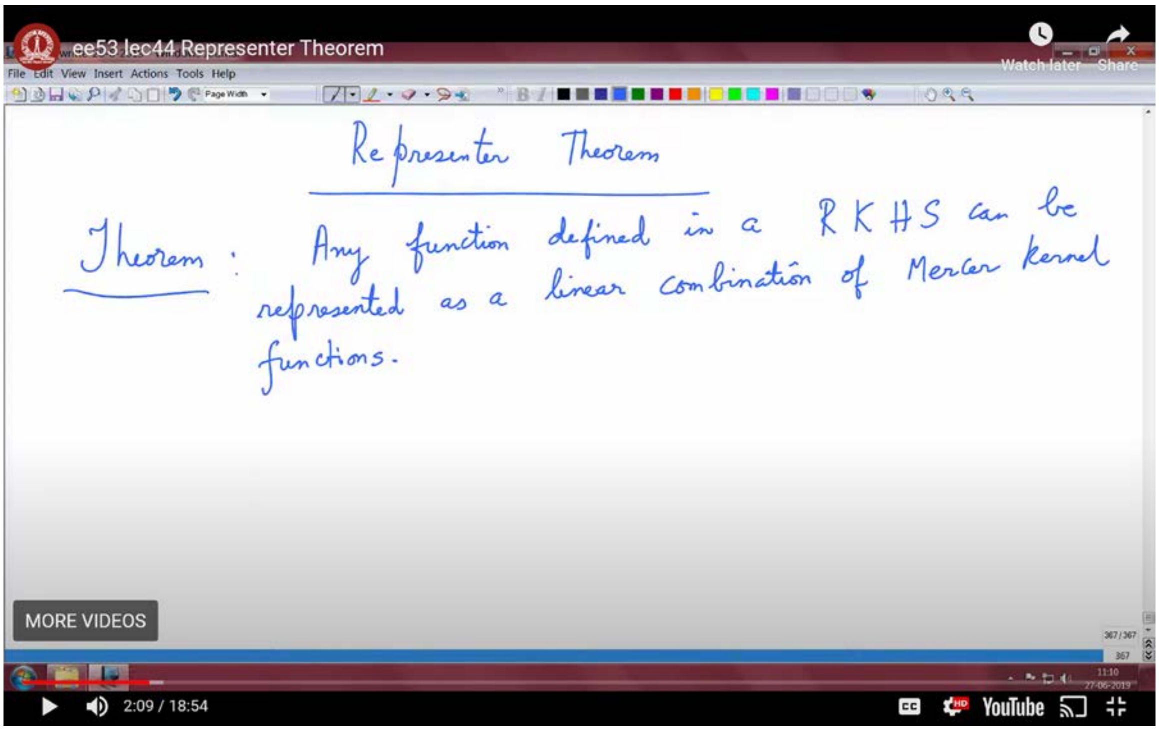
Prof. Shayan Srinivasa Garani
Department of Electronic System Engineering
Indian Institute of Science, Bengaluru

Lecture – 44

Representer Theorem

Earlier, we explored some foundational concepts related to Reproducing Kernel Hilbert Spaces (RKHS). Now, let's dive into the details of the representer theorem, which asserts that any function within an RKHS can be expressed as a linear combination of Mercer kernel functions. I'll now state this important theorem: Any function defined in a Reproducing Kernel Hilbert Space can indeed be represented as a linear combination of Mercer kernel functions.

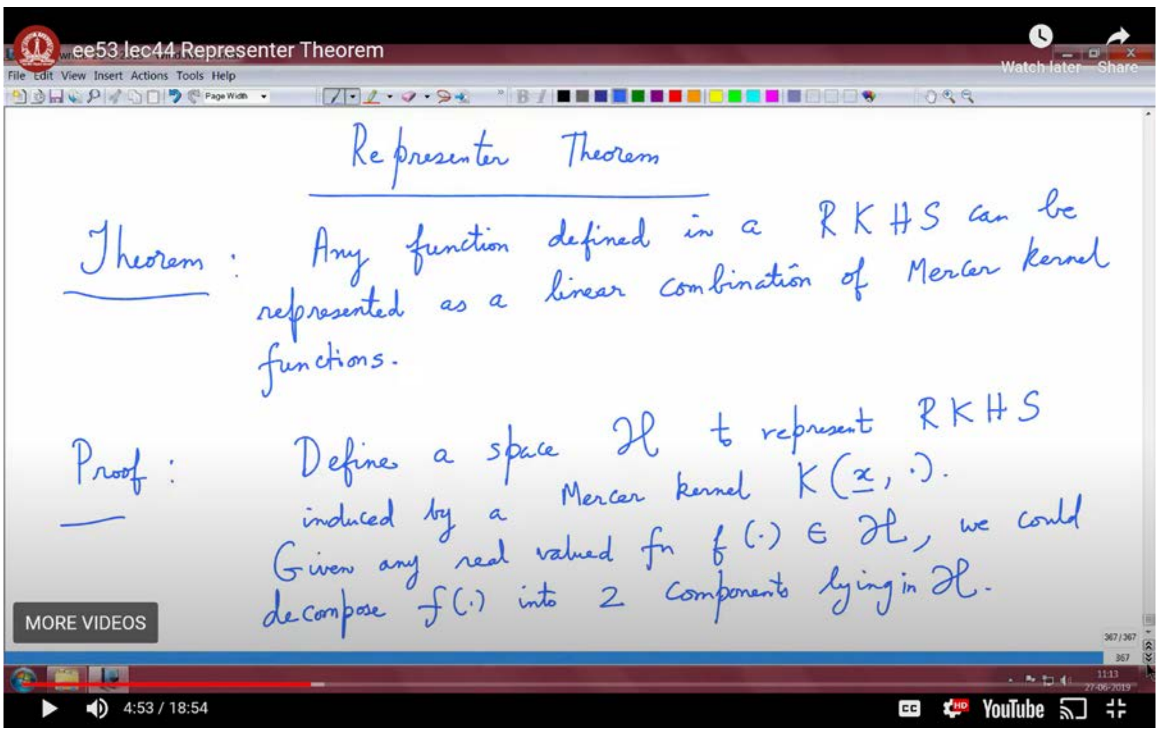
(Refer Slide Time: 02:09)



To provide a sketch of the proof, we begin by defining the space $\mathcal{H}$, which represents the RKHS induced by a Mercer kernel. For any real-valued function, this function can be decomposed into two distinct components: one that lies within the span of the kernel functions and another that belongs to the orthogonal complement.

The process involves taking the inner product of this function with the kernel and simplifying using the reproducing kernel property. That's the basic idea, so let's delve into the details.

(Refer Slide Time: 04:53)



The proof starts by defining the space $\mathcal{H}$, corresponding to the RKHS induced by a Mercer kernel. Let's denote this kernel as $k(x, x_i)$, where the dot notation indicates that x_i is a specific data point.

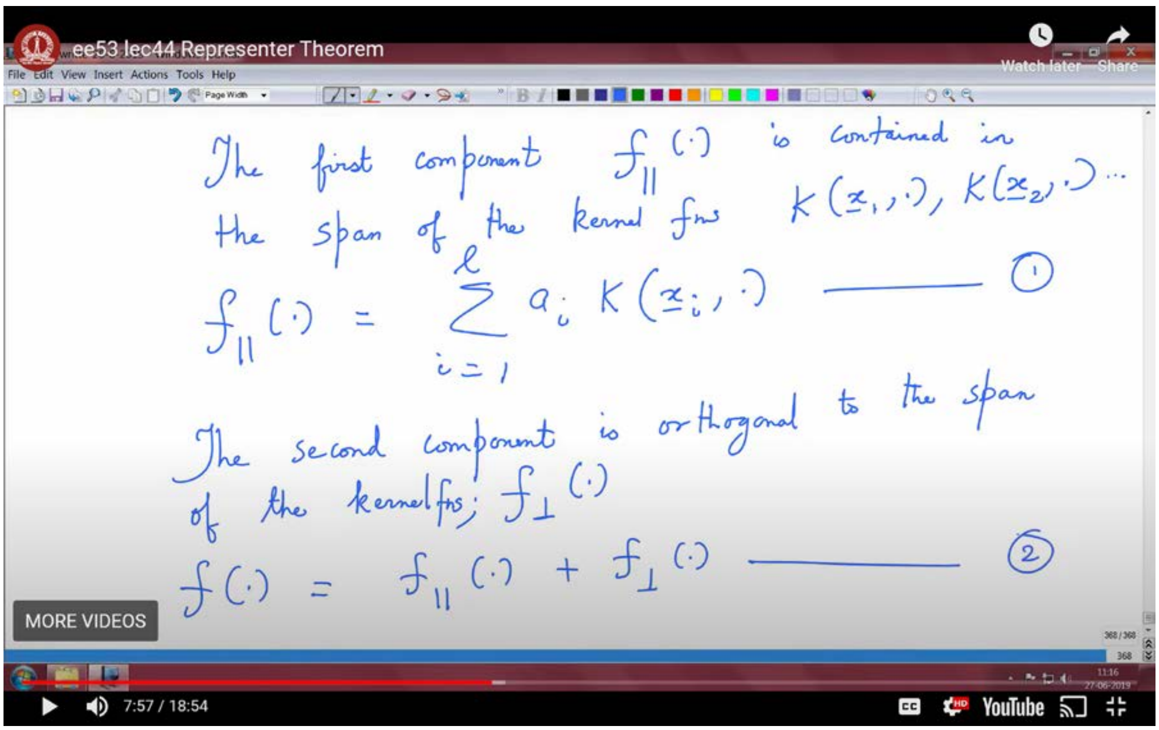
The key idea here is that for any real-valued function f belonging to this space $\mathcal{H}$, we can decompose f into two orthogonal components within the space. To be precise, the first component, denoted as $f_{\|}$, lies in the span of the kernel functions $k(x_1), k(x_2)$, …. We can express $f_{\|}$ acting on a data vector as:

$$f_{\parallel}(x) = \sum_{i=1}^{l} a_i k(x_i, x)$$

Let’s refer to this as Equation 1.

The second component is orthogonal to the span of the kernel functions, meaning that it does not lie within this span. We denote this orthogonal component as $f_{\perp}$. Essentially, $f_{\perp}$ is orthogonal to the function obtained from the span of the kernel functions.

(Refer Slide Time: 07:57)



Thus, the function f can be expressed as:

$$f(x) = f_{\parallel}(x) + f_{\perp}(x)$$

You can visualize these components as functions residing in spaces that are orthogonal complements of each other. Let’s call this Equation 2.

Let's substitute Equation 1 into Equation 2 and then simplify. Essentially, the function f can be expressed as:

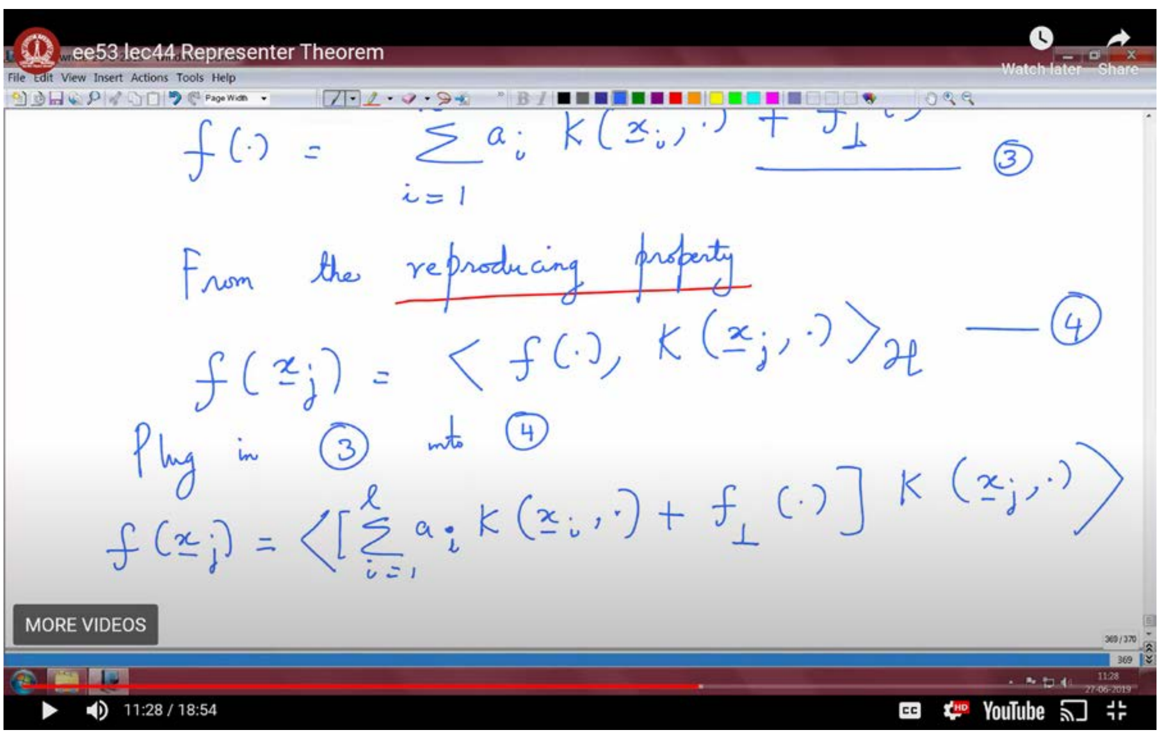
$$f(x) = \sum_{i=1}^{l} a_i k(x_i, x) + f_{\perp}(x)$$

Here, the dot indicates that the vector x can replace x_i. Now, leveraging the reproducing property of Mercer kernels, $f(x_j)$ can be written as the inner product of the function at some data point x_j with the kernel $k(x_j, x)$. This inner product, taken within the Hilbert space, essentially reproduces the function $f(x_j)$.

Let's call this Equation 3. Now, let's substitute Equation 3 into Equation 4. Thus, $f(x_j)$ becomes a function of x_j, which we expand in terms of the two components:

$$f(x_j) = \sum_{i=1}^{l} a_i k(x_i, x) + f_{\perp}(x)$$

(Refer Slide Time: 11:28)

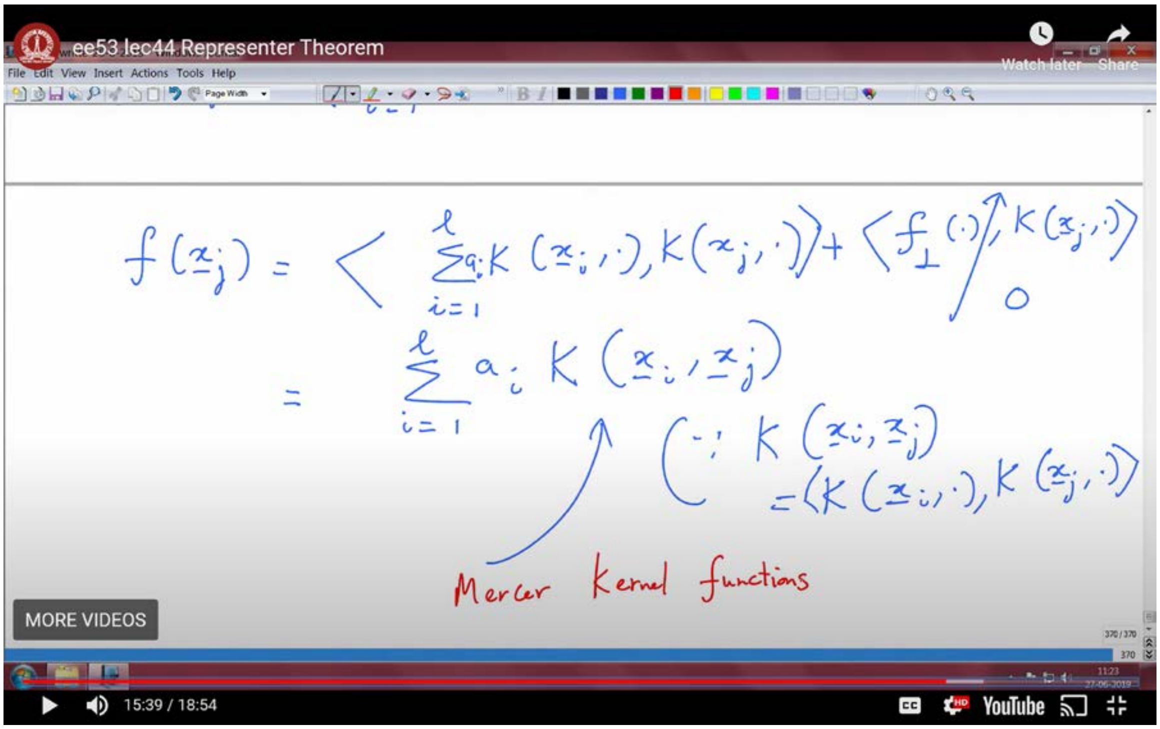


This is because $f(x_j)$ lies in the span of the kernel functions. Now, let's clarify the notation here, $f_{\parallel}$ should correspond to the first component, and $f_{\perp}$ to the second component, which is orthogonal to the kernel function, consistent with Equation 3.

To simplify further, consider:

$$f(x_j) = \left\langle \sum_{i=1}^{l} a_i k(x_i, x)\, , k(x_j, x) \right\rangle + \langle f_\perp, k(x_j, x)$$

(Refer Slide Time: 15:39)



Here, the first term represents the inner product of the function lying within the span of the kernel functions with the kernel $k(x_j, x)$. The second term is the inner product of the orthogonal component $f_\perp$ with the kernel. Since $f_\perp$ is orthogonal to the kernel functions, this second term evaluates to zero.

Therefore, what remains is the inner product of the parallel component:

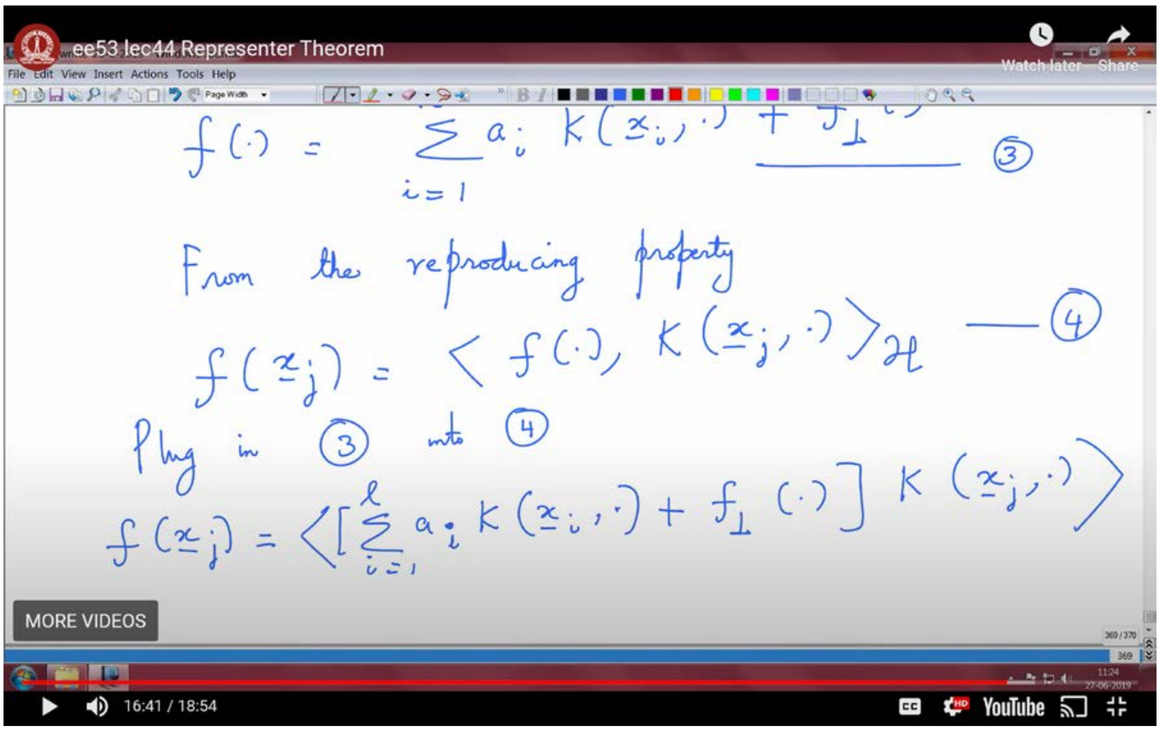
$$f(x_j) = \sum_{i=1}^{l} a_i k(x_i, x_j)$$

According to our definition, this can be written as:

$$f(x_j) = \sum_{i=1}^{l} a_i k(x_i, x_j)$$

Here, $k(x_i, x_j)$ is defined as the inner product of the data vectors x_i and x_j with respect to the kernel.

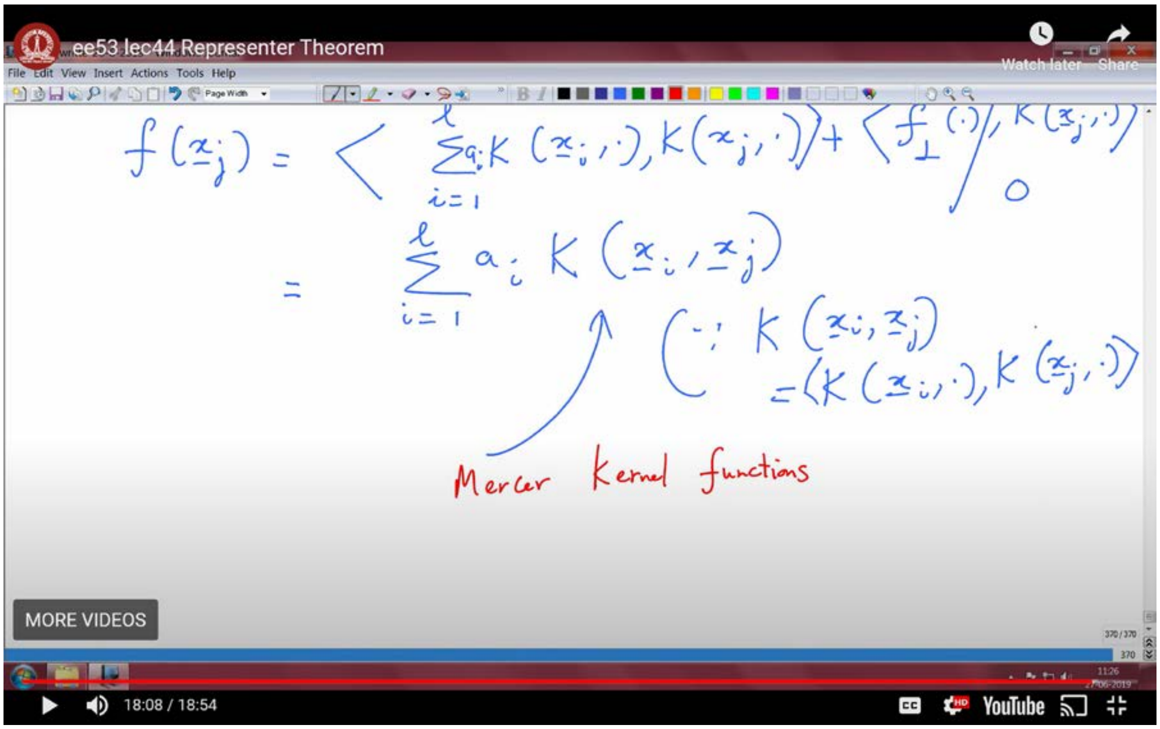
(Refer Slide Time: 16:41)



What this means is that any function in a reproducing kernel Hilbert space can be expressed as a linear combination of Mercer kernel functions. These Mercer kernel functions serve as the building blocks, allowing us to represent any function within the reproducing kernel space. The process we've gone through is quite straightforward, but it's crucial to appreciate the significance of this result.

To summarize: we started by defining a space representing the reproducing kernel Hilbert space, which is induced by a Mercer kernel. We then took a real-valued function within this space and decomposed it into two components. One component lies within the span of the kernel functions, and the other lies in the space orthogonal to the kernel functions, which we refer to as $f_{\perp}$. Therefore, the function f can be expressed as the sum of $f_{\parallel}$ and $f_{\perp}$.

(Refer Slide Time: 18:08)



Next, we invoked the reproducing property of the kernel, which allows us to express f acting on any data point as the inner product of f with the kernel function between different data points. This is the essence of the reproducing property of the kernel.

Now, by substituting f, which has two components, one within the kernel span and the other in its orthogonal complement, into the inner product, and applying the distributive property carefully, we observe that the inner product of $f_{\perp}$ with the kernel is zero. This outcome is due to the way we've constructed our space. What we are left with is a function that can be represented as a linear combination of these Mercer kernel functions.

This result is not only significant but also has intriguing applications, particularly in scenarios where we need to minimize empirical risks under certain regulatory conditions in the Hilbert space. For example, if we are given data samples (x_i, d_i), and we aim to minimize risk, whether it's a classification error or another form of risk, while incorporating these kernel ideas, perhaps within the framework of Support Vector Machines (SVMs) or an extended SVM framework, this theorem becomes extremely useful.

With that, we'll stop here and move on to discuss the broader applicability of the representer theorem.