A Journey Through Hilbert Space and Kernel Methods

Hıncal Topçuoğlu

February 22, 2026

1 Introduction

A Hilbert Space is a mathematical concept that generalizes Euclidean space to infinite dimensions, providing a complete inner product space where concepts like length and angle are well-defined.

When working with vectors, Hilbert Space allows us to operate in infinite dimensions. But what does this mean, and what is the benefit?

In Machine Learning problems, it is often difficult to separate data and variables in normal-dimensional systems. Therefore, by mapping data into a Hilbert Space, we make it easier to distinguish between them. In other words, it would be accurate to define Hilbert Space as a space that allows geometry (angle, distance, orthogonality) to work in infinite dimensions and on functions.

2 Definitions of Hilbert Space

2.1 1. Vector Space (The Foundation)

As an example, in $\mathbb{R}^n$ (n-dimensional real numbers), the rule must be: if $u, v \in V$, then $u + v \in V$ and $c \cdot u \in V$.

2.2 2. Inner Product Space

To measure angle, length, and orthogonality in a Vector Space, we define an inner product function: $\langle u, v \rangle$. From this, we derive:

- **Length (Norm):** $\|u\| = \sqrt{\langle u, v \rangle}$
- **Angle and Orthogonality:** If $\langle u, v \rangle = 0$, then u and v are orthogonal (perpendicular) to each other. This structure is known as a pre-Hilbert Space.

2.3 3. Completeness - Closing the Gaps

Mathematically, consider a converging sequence $(x_1, x_2, ..., x_n)$. If this sequence approaches a point, that limit point must also exist within the same space.

- **Cauchy Sequence:** A sequence whose elements become arbitrarily close to each other.
- **Complete Space:** A space where the limit of every Cauchy sequence is also contained within that space.

Therefore, a Hilbert Space can be expressed as:

$$\text{Hilbert Space} = (\text{Vector Space}) + (\text{Inner Product}) + (\text{Completeness})$$

3 The Connection to Machine Learning: Kernels

What does this have to do with the "Kernel" in Machine Learning?

A Kernel is a function that allows us to easily separate data by mapping it to a higher-dimensional space (Hilbert Space) when it is difficult to separate in a lower-dimensional space.

Analogy: Imagine billiard balls on a table; it is hard to separate them with a single straight line. However, you can think of the kernel function as hitting the table and lifting the balls into the air, making it easier to separate the data in 3D space.

Mathematical Definition: Mapping data to a high dimension ($x \to \phi(x)$) and performing operations there ($\langle \phi(x), \phi(y) \rangle$) is computationally very expensive. The Kernel function, $K(x, y)$, gives us the chance to act as if we have gone to that space without ever actually entering it. Here, ϕ is the mapping function to the high dimension, and K is the kernel function. The trick is: ϕ is never calculated; instead, the function $K(x, y)$ is used directly.

Types of Kernel Functions:

- $K(x, y) = x \cdot y \Rightarrow$ Linear
- $K(x, y) = (x \cdot y + c)^d \Rightarrow$ Polynomial

Radial Basis Function (RBF) Kernel: We can say this is the most popular in ML. It maps data to an infinite-dimensional space and finds the similarity between them.

$$K(x, y) = \exp(-\gamma \|x - y\|^2)$$

This kernel function is widely used in SVM and PCA analyses.

4 Functions in Inner Product Space

How does a function provide length, angle, and orthogonality in an inner product space? It is easy to imagine vectors in terms of angle and length, but for functions, this can be difficult and may seem strange at first.

4.1 Transition from Vectors to Functions

$$v = [v_1, v_2, v_3, ..., v_n]$$
$$f(x) = [f(x_1), f(x_2), ..., f(x_n)]$$

The key point to note here is that the "summation" operation performed on vectors turns into "integration" for functions.

Therefore, the inner product is:

- **For Vectors:** $\langle u, v \rangle = u_1 v_1 + u_2 v_2 + ... + u_n v_n$
- **For Functions:** $\langle f, g \rangle = \int_a^b f(x) g(x)\, dx$

It is useful to mention that length is related to the Pythagorean theorem, i.e., $x^2 + y^2 = r^2$. For functions, length is the magnitude of the square root of the area under the graph of that function squared.

$$\|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int_a^b (f(x))^2\, dx}$$

Example: Let $f(x) = 1$ and $g(x) = x$. If we consider these functions in the interval $[0, 1]$:

$$f(x) = \sqrt{\int_0^1 1^2\, dx} = 1$$

$$g(x) = \sqrt{\int_0^1 x^2\, dx} = \sqrt{\left[\frac{x^3}{3}\right]_0^1} = \sqrt{\frac{1}{3}} \approx 0.577$$

Thus, $f(x)$ is a "longer" function than $g(x)$.

4.2 Finding the Angle

In vectors, the angle (θ) shows how similar the values are to each other. $\langle u, v \rangle = \|u\| \cdot \|v\| \cdot \cos(\theta)$. Let's apply this to functions:

$$\cos(\theta) = \frac{\langle f, g \rangle}{\|f\| \cdot \|g\|} = \frac{\int f(x)g(x)\,dx}{\sqrt{\int f^2\,dx} \cdot \sqrt{\int g^2\,dx}}$$

- If $\cos(\theta) = 1 \rightarrow$ All functions are in the same direction.
- If $\cos(\theta) = -1 \rightarrow$ Opposite direction.
- If $\cos(\theta) = 0 \rightarrow$ The functions are **orthogonal** (perpendicular).

Orthogonality: If the integral of the product of two functions is zero, these functions are orthogonal.

$$\int_a^b f(x)g(x)\,dx = 0$$

This means there is a state of complete independence between the functions. The product of the functions balances out so well in positive and negative areas that the total result is zero, and they do not interact with each other. In other words, when decomposing a signal, we are actually trying to find the "orthogonality" within that signal. The purpose of finding orthogonality is to separate two classes orthogonally from each other and to classify them correctly.

5 Geometric Classification Graph

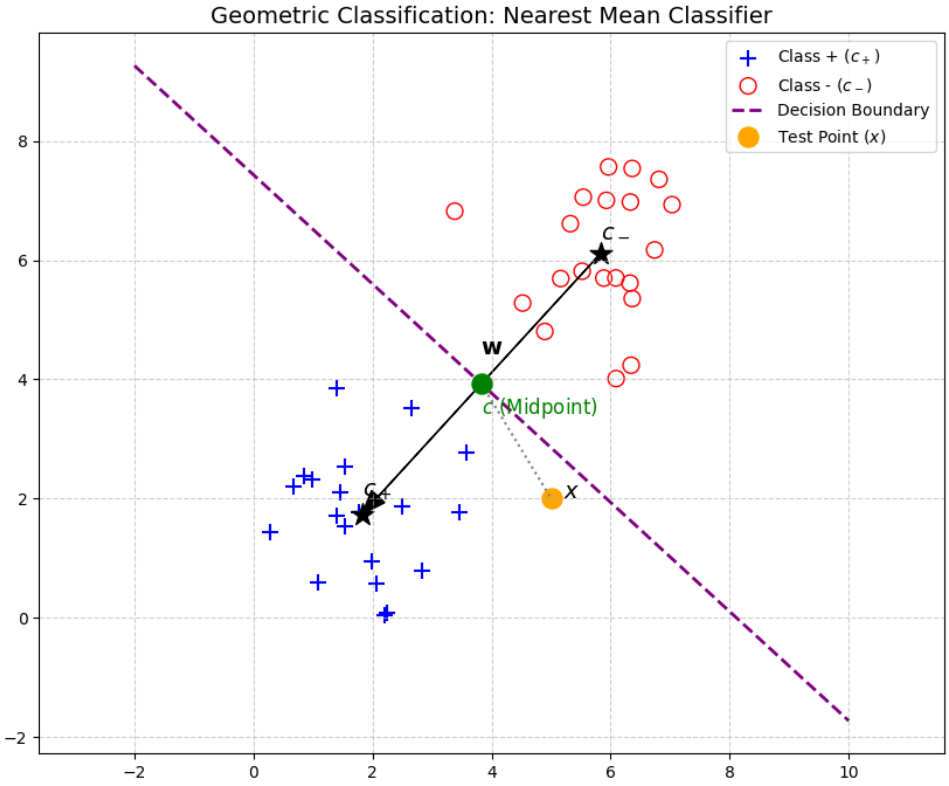

Figure 1: Geometric Classification: Nearest Mean Classifier

6 Missing Conditions in Hilbert Space

Note: The three conditions of Hilbert Space → Vector Space + Inner Product + Completeness create the best possible example space that can be defined geometrically.

Question: What happens if one of these conditions is missing?

6.1 Example: Banach Space

If a space has vectors, length can be measured, and the space is complete, but **angles cannot be measured** (no inner product), this is called a **Banach Space**.

What is its characteristic?

- **Orthogonality:** We cannot say that "two data sets are orthogonal to each other."
- **Projection:** Consequently, a projection cannot be found, and the best approximation cannot be obtained. This is because the best approximation is where the error is "orthogonal" to the data. For example, the Pythagorean theorem does not work here.
- **Machine Learning Context:** An algorithm that operates in this space is **Lasso Regression** (L1 Regularization).

6.2 Condition 2: What if Completeness is missing?

In this space, which we can call a **Pre-Hilbert Space**, if the completeness condition is not met:

- An optimization method used, such as **Gradient Descent**, might reduce the error, but since the determined limit does not exist within that space, it enters an infinite loop.
- In other words, it is the situation where the function intended to converge does not exist in that space.

If the length condition is also missing, vector calculations cannot be performed at all.

6.3 Recap

In a classification example, let's consider the function:

$$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

Here:

- ϕ (phi) is the function that maps "x" vectors to the Hilbert Space (Feature Mapping Function).
- From this movement, if we define a similarity measure, we reach the functional structure referred to as the "Kernel" (k).

The Advantage: The purpose of using a kernel is to allow us to build algorithms entirely within a **"dot product" space!**

7 Detailed Analysis of Kernel Methods

7.1 Kernels

We are given data, $(x_1, y_1), ..., (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$. X are predictors and y is target. $i, j \in [n]$ where $[n] := \{1, ..., n\}$.

Here, we do not make any assumption on X. Do not forget that in the learning step, we want to be able to generalize "unseen" data points.

In the case of binary classification, we want to predict $y \in \{\pm 1\}$. So we actually want to find y such that (x, y) is in some sense **similar** to training examples.

To measure similarity in X in $\{\pm 1\}$, we prove that these values are identical so it requires a function that $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $(x, x') \mapsto k(x, x')$. So we will search for $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$. Here Φ maps product space to $\mathcal{H}$, Hilbert space or feature map space. The "k" is called as "**kernel**".

We will use this "kernel" function as similarity means to construct algorithms in dot product spaces.

7.2 Simple Classification Algorithm

In a simple classification algorithm, the purpose is to compute means of two classes in that feature space.

So $y = \{\pm 1\}$,

$$c_+ = \frac{1}{n_+} \sum_{\{i:y_i=+1\}} \Phi(x_i)$$

and

$$c_- = \frac{1}{n_-} \sum_{\{i:y_i=-1\}} \Phi(x_i)$$

Here n_+ and n_- are number of samples due to each class.

If we assign a new point $\Phi(x)$ to the class whose mean closer to it, this leads a prediction rule,

$$y = \text{sgn}(\langle \Phi(x), c_+ \rangle - \langle \Phi(x), c_- \rangle + b)$$

b is bias term which is $b = \frac{1}{2}(\|c_-\|^2 - \|c_+\|^2)$.

7.3 Substituting the Means

If we substitute $c_\pm$, it follows:

$$y = \text{sgn}\left(\frac{1}{n_+} \sum_{\{i:y_i=+1\}} \underbrace{\langle \Phi(x), \Phi(x_i) \rangle}_{k(x,x_i)} - \frac{1}{n_-} \sum_{\{i:y_i=-1\}} \underbrace{\langle \Phi(x), \Phi(x_i) \rangle}_{k(x,x_i)} + b \right)$$

Meaning: When a new "x" point arrives, we look at whether it makes more similarity with the positive class or the negative class; we look at the "**dot product**" it makes. Whichever one it matches with more, it belongs to that class.

In other words:

- I multiply the new incoming point x with all the positive values I have.
- I take the average of this product.
- This tells me: "How much does this new point resemble the positives?"
- The same case applies to the negative class.

7.4 The Decision Moment

The decision moment is the "subtraction" term in the formula. It compares these two forces.

Here, in the Hilbert space, "feature space", if we want to write two probability distributions, prediction is usually done with the "kernel" function:

$$P_+(x) := \frac{1}{n_+} \sum_{\{i:y_i=+1\}} k(x, x_i)$$

$$P_-(x) := \frac{1}{n_-} \sum_{\{i:y_i=-1\}} k(x, x_i)$$

This classifier forms the basis of the **SVM algorithm**.

References

[1] Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). *Kernel Methods in Machine Learning.* The Annals of Statistics, 36(3), 1171-1220.

[2] Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press.

[3] Kreyszig, E. (1978). *Introductory Functional Analysis with Applications.* John Wiley & Sons.

[4] Amari, S. (2016). *Information Geometry and Its Applications*. Springer.

[5] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.