1 Reproducing Kernel Hilbert Space (RKHS)

Let $L_2(\mathcal{X})$ be the set of all functions $f : \mathcal{X} \rightarrow \mathbb{R}$ that are square-integrable; that is, $\int_{\mathcal{X}} |f(x)|^2 dx < \infty$. $\mathcal{X}$ is the data-space, usually $\mathbb{R}^d$. In short, RKHS is a subset of $L_2(\mathcal{X})$.

More specifically, if we think of functions as a continuous vector, then RKHS is a set of functions with a special inner product, and this inner product is associated with a kernel.

We will first define a kernel and then define RKHS.

**Definition 1.** A Kernel is a function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that

1. it is symmetric: $K(x, y) = K(y, x)$.

2. positive semi-definite (often just referred to as “positive definite”): $\forall x_1, \ldots, x_n \in \mathcal{X}$, the $n \times n$ matrix $K_{i,j} = K(x_i, x_j)$ is positive semi-definite.

Note that this definition of positive semi-definiteness is equivalent to saying that $\int_{\mathcal{X}} \int_{\mathcal{X}} K(x, y) f(x) f(y) dx dy \geq 0$ for all square-integrable function $f$.

Defining RKHS is tricky; we will start out with an initial set of special functions and then add more functions to the initial set in a process called *completion* to get the final RKHS:

**Definition 2.** Let $K_{x_j}$ denote a function $\mathcal{X} \rightarrow \mathbb{R}$ such that $K_{x_j}(x) = K(x_j, x)$.

$$\mathcal{H}_0 = \{ f = \sum_{j=1}^{k} \alpha_j K_{x_j} | x_j \in \mathcal{X}, \alpha_j \in \mathbb{R}, k \in \mathbb{N} \}$$

Let $f = \sum_{j=1}^{k} \alpha_j K_{x_j}$ and let $g = \sum_{j=1}^{m} \beta_j K_{y_j}$, then define a new inner product:

$$\langle f, g \rangle_K = \sum_{i=1}^{k} \sum_{j=1}^{m} \alpha_i \beta_j K(x_i, y_j)$$
The norm (distance) induced by this inner product is:

$$
\|f\|_K = \sqrt{\sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_i \alpha_j K(x_i, x_j)}
$$

A Cauchy sequence is an infinite sequence of functions \( \{f_1, f_2, \ldots \} \subset \mathcal{H}_0 \) such that \( \|f_n - f_{n+1}\|_K \) goes to 0 as \( n \) goes to infinity. All such sequences have a limit in \( L_2(\mathcal{X}) \), although the limit might not be in \( \mathcal{H}_0 \).

To complete the space, we will include the limits of all the Cauchy sequences. There are technical issues:

1. \( f \in \mathcal{H}_0 \) can be represented in multiple ways, we must ensure \( \langle f, g \rangle \) will not change value if we change representation of \( f \).
2. We will have expand definition of inner product to handle Cauchy-sequence limits

An important characteristics of RKHS is the Reproducing Property:

**Proposition 3. (Reproducing Property)**

- \( \langle K_x, K_y \rangle_K = K(x, y) \)
- Let \( f \in \mathcal{H}_0 \), then \( \langle f, K_x \rangle_K = f(x) \)

1.1 Connection to Discrete Vector Space

If we think of a function \( f \) as a continuous vector where \( f(x) \) is accessing the \( x \)-th position of the vector, then a positive semi-definite Kernel is similar to a positive semi-definite matrix.

In the following example, we will denote \( u, v \in \mathbb{R}^p \) and \( M \in \mathbb{R}^{p \times p} \). \( Mv \) is a vector and \( (Mv)(i) = \sum_j M_{i,j}v_j \). Similarly, \( g(y) = \int_x K(x, y)f(x)dx \) is a function.

**However**, we cannot stretch out the analogy too far; the inner product for discrete vector space \( \langle u, v \rangle = \sum_i u(i)v(i) \) has a special form the relates to matrix multiplication. The inner product we define for RKHS is more abstract; it is not at all similar to \( \langle u, v \rangle \) and it is not directly related to “continuous matrix multiplication”.

2 Kernel As a Measure of Similarity

We will now present Kernels in a different way - the way that you probably first learned it.

Given a data point \( x \in \mathcal{X} \), we can define a feature map \( \Phi : \mathcal{X} \rightarrow \mathcal{F} \) where \( \mathcal{F} \) is the feature space, a discrete possibly infinite-dimensional vector space. We call \( \Phi(x) \) the feature vector.

For example, suppose a data \( x = (x_1, x_2, x_3) \) is a 3-dimensional vector, then we can define a polynomial feature map:

\[
\Phi(x) = (x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_2x_3, \sqrt{2}x_1x_3)
\]

The feature space is 9-dimensional, and comprises monomials of degree at most 2.

The Kernel then is defined to be \( K(x, y) = \langle \Phi(x), \Phi(y) \rangle \). Intuitively, we think of \( K(x, y) \) as a measure of similarity between data \( x \) and \( y \). In SVM, using feature map and kernels allow you to create non-linear decision
boundaries.

All Feature Maps induce PSD kernels but the feature map is impractical if the kernel is not easy to compute. Conversely, all PSD kernels also define a feature map.

**Theorem 4. (Mercer’s Theorem)** Suppose \( K \) is a symmetric positive semi-definite Kernel. Then there exist a set of orthonormal eigen-functions \( \{\psi_j : \mathcal{X} \to \mathbb{R}\}_{j=1,\ldots,N} \) (\( N \) possibly infinity) and a set of eigenvalues \( \lambda_j > 0 \) such that

- \( \sum_{j=1}^{N} \lambda_j < \infty \)
- \( K(x,y) = \sum_{j=1}^{N} \lambda_j \psi_j(x)\psi_j(y) \)

**Definition 5.** Let \( K \) be a symmetric positive semi-definite Kernel with eigenvalues \( \lambda_1 \geq \lambda_2 \geq \ldots \lambda_N \) and eigenfunctions \( \{\psi_j\}_{j=1,\ldots,N} \) (\( N \) again could be infinity).

Then define a Feature Map \( \Phi : \mathcal{X} \to \mathbb{R}^N \) as

\[
\Phi(x) = (\sqrt{\lambda_1} \psi_1(x), \sqrt{\lambda_2} \psi_2(x), \ldots, \sqrt{\lambda_N} \psi_N(x))
\]

Using the standard inner product on \( \mathbb{R}^N \), we see that

\[
\langle \Phi(x), \Phi(y) \rangle = \sum_{j=1}^{N} \lambda_j \psi_j(x)\psi_j(y) = K(x,y)
\]

### 2.1 Support Vector Machine

Recall that in SVM, the dual optimization is:

\[
\max_{\alpha_1,\ldots,\alpha_n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle
\]

The kernelized version is:

\[
\max_{\alpha_1,\ldots,\alpha_n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle = \max_{\alpha_1,\ldots,\alpha_n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
\]

Recall that with optimal \( \alpha_i \)'s, the resulting decision function is of the form

\[
f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i K(x, x_i) - b \right)
\]

Optimizing the kernelized SVM is equivalent to searching in the corresponding RKHS for a function to use as classifier.
Notice that \( \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) = z^T K z \) where \( z_i = \alpha_i y_i \). Since \( K \) is positive semi-definite, it is easy to show that the optimization in \( \alpha_i \) is convex.

However, if we used a generic similarity function \( S(x, y) \) that is not symmetric positive semi-definite, then the resulting optimization need not be convex.

To summarize:

- Every feature map defines a PSD Kernel and every PSD Kernel defines a feature map
- We can think of Kernels as similarity functions but the PSD property separates them from generic similarity functions and makes them more useful.
- Performing the kernel trick is similar to working in RKHS.

### 2.2 Examples

- **Homogenous Polynomial Kernel**
  \( K(x, y) = \langle x, y \rangle^r \)
  Feature map \( \Phi(x) \) all monomials of degree \( r \) formed by coordinates of \( x \)

- **Inhomogeneous Polynomial Kernel**
  \( K(x, y) = (\langle x, y \rangle + 1)^r \)
  Feature map \( \Phi(x) \) all monomials of degree \( r \) or less formed by coordinates of \( x \)

- **Radial Basis Kernel**
  \( K(x, y) = \exp(-\frac{|x-y|^2}{\sigma^2}) \)
  Feature map \( \Phi(x) \) basis polynomials of all degrees (infinite dimensional)

- **String Kernel**

### 3 Representer Theorem

A seemingly different way to motivate Kernels is regularized risk minimization. The key is the representer theorem:

**Theorem 6.** (Representer Theorem) Let \( (X_1, Y_1), \ldots, (X_n, Y_n) \) be \( n \) data. Let \( c : (X \times Y)^n \to \mathbb{R} \) be an arbitrary loss function. Let \( \Omega : [0, \infty) \to \mathbb{R} \) be a strictly monotonically increasing function.

Let \( \mathcal{H}_K \) be a RKHS with PSD kernel \( K \), then

\[
\arg\min_{f \in \mathcal{H}_K} c((f(X_1), Y_1), \ldots, (f(X_n), Y_n)) + \Omega(||f||_K)
\]

has the form \( f = \sum_{i=1}^n \alpha_i K x_i \).

Hence, as in the case with SVM, to optimize over RKHS, we only need to optimize over the \( \alpha_i \)'s.