The Representer Theorem

**Theorem 4** Given: a p.d. kernel $k$ on $\mathcal{X} \times \mathcal{X}$, a training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathbb{R}$, a strictly monotonic increasing real-valued function $\Omega$ on $[0, \infty[$, and an arbitrary cost function $c : (\mathcal{X} \times \mathbb{R}^2)^m \rightarrow \mathbb{R} \cup \{\infty\}$

Any $f \in \mathcal{F}$ minimizing the regularized risk functional

$$c((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))) + \Omega(\|f\|) \quad (3)$$

admits a representation of the form

$$f(.) = \sum_{i=1}^{m} \alpha_i k(x_i, .).$$
Mercer’s Theorem

If $k$ is a continuous kernel of a positive definite integral operator on $L_2(X)$ (where $X$ is some compact space),

$$
\int_X k(x, x') f(x) f(x') \, dx \, dx' \geq 0,
$$

it can be expanded as

$$
k(x, x') = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x')
$$

using eigenfunctions $\psi_i$ and eigenvalues $\lambda_i \geq 0$ [41].
The Mercer Feature Map

In that case

$$\Phi(x) := \begin{pmatrix} \sqrt{\lambda_1} \psi_1(x) \\ \sqrt{\lambda_2} \psi_2(x) \\ \vdots \end{pmatrix}$$

satisfies $$\langle \Phi(x), \Phi(x') \rangle = k(x, x').$$

Proof:

$$\langle \Phi(x), \Phi(x') \rangle = \left\langle \begin{pmatrix} \sqrt{\lambda_1} \psi_1(x) \\ \sqrt{\lambda_2} \psi_2(x) \\ \vdots \end{pmatrix}, \begin{pmatrix} \sqrt{\lambda_1} \psi_1(x') \\ \sqrt{\lambda_2} \psi_2(x') \\ \vdots \end{pmatrix} \right\rangle$$

$$= \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x') = k(x, x')$$
Positive Definite Kernels

It can be shown that (modulo some details) the admissible class of kernels coincides with the one of positive definite (pd) kernels: kernels which are symmetric, and for

- any set of training points $x_1, \ldots, x_m \in \mathcal{X}$ and
- any $a_1, \ldots, a_m \in \mathbb{R}$

satisfy

$$\sum_{i,j} a_i a_j K_{ij} \geq 0, \text{ where } K_{ij} := k(x_i, x_j).$$
Elementary Properties of PD Kernels

Kernels from Feature Maps.
If $\Phi$ maps $\mathcal{X}$ into a dot product space $\mathcal{H}$, then $\langle \Phi(x), \Phi(x') \rangle$ is a pd kernel on $\mathcal{X} \times \mathcal{X}$.

Positivity on the Diagonal.
$k(x, x) \geq 0$ for all $x \in \mathcal{X}$

Cauchy-Schwarz Inequality.
$k(x, x')^2 \leq k(x, x)k(x', x')$ (Hint: compute the determinant of the Gram matrix)

Vanishing Diagonals.
$k(x, x) = 0$ for all $x \in \mathcal{X} \implies k(x, x') = 0$ for all $x, x' \in \mathcal{X}$
The Feature Space for PD Kernels \[4, 1, 48\]

- define a feature map
  \[
  \Phi : \mathcal{X} \rightarrow \mathbb{R}^\mathcal{X}
  \]
  \[
  x \mapsto k(., x).
  \]

E.g., for the Gaussian kernel:

Next steps:
- turn $\Phi(\mathcal{X})$ into a linear space
- endow it with a dot product satisfying
  \[
  \langle k(., x_i), k(., x_j) \rangle = k(x_i, x_j)
  \]
- complete the space to get a reproducing kernel Hilbert space
Endow it With a Dot Product

\[ \langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j) \]

\[ = \sum_{i=1}^{m} \alpha_i g(x_i) = \sum_{j=1}^{m'} \beta_j f(x'_j) \]

- This is well-defined, symmetric, and bilinear.
- It can be shown that it is also strictly positive definite (hence it is a dot product).
- Complete the space in the corresponding norm to get a Hilbert space \( \mathcal{H}_k \).
The Reproducing Kernel Property

Two special cases:

• Assume

\[ f(.) = k(., x). \]

In this case, we have

\[ \langle k(., x), g \rangle = g(x). \]

• If moreover

\[ g(.) = k(., x'), \]

we have the kernel trick

\[ \langle k(., x), k(., x') \rangle = k(x, x'). \]

\( k \) is called a reproducing kernel for \( \mathcal{H}_k \).
Turn it Into a Linear Space

Form linear combinations

\[ f(.) = \sum_{i=1}^{m} \alpha_i k(., x_i), \]

\[ g(.) = \sum_{j=1}^{m'} \beta_j k(., x'_j) \]

\((m, m' \in \mathbb{N}, \alpha_i, \beta_j \in \mathbb{R}, x_i, x'_j \in X).\)
The Reproducing Kernel Property

Two special cases:

- Assume
  \[ f(\cdot) = k(\cdot, x). \]
  In this case, we have
  \[ \langle k(\cdot, x), g \rangle = g(x). \]

- If moreover
  \[ g(\cdot) = k(\cdot, x'), \]
  we have the kernel trick
  \[ \langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x'). \]

\( k \) is called a reproducing kernel for \( \mathcal{H}_k \).
Recall that the dot product has to satisfy
\[
\langle k(x, .), k(x', .) \rangle = k(x, x').
\]

For a Mercer kernel
\[
k(x, x') = \sum_{j=1}^{N_F} \lambda_j \psi_j(x) \psi_j(x')
\]
(with \(\lambda_i > 0\) for all \(i\), \(N_F \in \mathbb{N} \cup \{\infty\}\), and \(\langle \psi_i, \psi_j \rangle_{L^2(X)} = \delta_{ij}\)), this can be achieved by choosing \(\langle ., . \rangle\) such that
\[
\langle \psi_i, \psi_j \rangle = \delta_{ij}/\lambda_i.
\]
To see this, compute

\[ \langle k(x, .), k(x', .) \rangle = \left\langle \sum_{i} \lambda_i \psi_i(x) \psi_i, \sum_{j} \lambda_j \psi_j(x') \psi_j \right\rangle \]

\[ = \sum_{i, j} \lambda_i \lambda_j \psi_i(x) \psi_j(x') \langle \psi_i, \psi_j \rangle \]

\[ = \sum_{i, j} \lambda_i \lambda_j \psi_i(x) \psi_j(x') \delta_{ij} / \lambda_i \]

\[ = \sum_{i} \lambda_i \psi_i(x) \psi_i(x') \]

\[ = k(x, x'). \]
Some Properties of Kernels [53]

If \( k_1, k_2, \ldots \) are pd kernels, then so are

- \( \alpha k_1 \), provided \( \alpha \geq 0 \)
- \( k_1 + k_2 \)
- \( k_1 \cdot k_2 \)
- \( k(x, x') := \lim_{n \to \infty} k_n(x, x') \), provided it exists
- \( k(A, B) := \sum_{x \in A, x' \in B} k_1(x, x') \), where \( A, B \) are finite subsets of \( \mathcal{X} \)
  (using the feature map \( \tilde{\Phi}(A) := \sum_{x \in A} \Phi(x) \))

Further operations to construct kernels from kernels: tensor products, direct sums, convolutions [28].
Computing Distances in Feature Spaces

Clearly, if $k$ is positive definite, then there exists a map $\Phi$ such that

$$\|\Phi(x) - \Phi(x')\|^2 = k(x, x) + k(x', x') - 2k(x, x')$$

(it is the usual feature map).

This embedding is referred to as a *Hilbert space representation* as a distance. It turns out that this works for a larger class of kernels, called *conditionally positive definite*.

In fact, all algorithms that are translationally invariant (i.e. independent of the choice of the origin) in $\mathcal{H}$ work with cpd kernels [53].