Kernels
Duke Course Notes
Cynthia Rudin

Credits: Bartlett, Schölkopf and Smola, Cristianini and Shawe-Taylor

The *kernel trick* that I’m going to show you applies much more broadly than SVM, but we’ll use it for SVM’s.

Basic idea: If you can’t separate positives from negatives very well in a low-dimensional space using a hyperplane, then map everything to a higher dimensional space where you can separate them better.

The picture looks like this but it’s going to be a little confusing at this point. In any case, you can see that the original data in 2D is not separable but in higher dimensions it is.

The left figure might be confusing because we took the data, mapped it to a higher dimensional space, created a linear boundary in that space, and then went back and drew that boundary in the original space.

Say I want to predict whether a house on the real-estate market will sell today
or not:

\[
x = \begin{bmatrix}
\begin{array}{cccc}
\underbrace{x^{(1)}} & \underbrace{x^{(2)}} & \underbrace{x^{(3)}} & \underbrace{x^{(4)}} & \cdots
\end{array}
\end{bmatrix},
\begin{array}{l}
\text{house's list price} \\
\text{estimated worth} \\
\text{length of time on market} \\
\text{in a good area}
\end{array}
\]

We might want to consider something more complicated than a linear model:

**Example 1:** \([x^{(1)}, x^{(2)}] \rightarrow \Phi \left([x^{(1)}, x^{(2)}]\right) = [x^{(1)2}, x^{(2)2}, x^{(1)}x^{(2)}]

The 2d space gets mapped to a 3d space. We could have the inner product in the 3d space:

\[
\Phi(x)^T \Phi(z) = x^{(1)2}z^{(1)2} + x^{(2)2}z^{(2)2} + x^{(1)}x^{(2)}z^{(1)}z^{(2)}.
\]

**Example 2:**

\([x^{(1)}, x^{(2)}, x^{(3)}] \rightarrow \Phi \left([x^{(1)}, x^{(2)}, x^{(3)}]\right) = [x^{(1)2}, x^{(1)}x^{(2)}, x^{(1)}x^{(3)}, x^{(2)}x^{(1)}, x^{(2)}x^{(2)}, x^{(2)}x^{(3)}, x^{(3)}x^{(1)}, x^{(3)}x^{(2)}, x^{(3)2}]

and we can take inner products in the 9d space, similarly to the last example.

Rather than apply SVM to the original \(x\)'s, apply it to the \(\Phi(x)\)'s instead.

The kernel trick is that if you have an algorithm (like SVM) where the examples appear only in inner products, you can freely replace the inner product with a different one. (And it works just as well as if you designed the SVM with some map \(\Phi(x)\) in mind in the first place, and took the inner product in \(\Phi\)'s space instead.)

Remember the optimization problem for SVM?

\[
\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \leftarrow \text{inner product}
\]

s.t. \(0 \leq \alpha_i \leq C, i = 1, \ldots, n\) and \(\sum_{i=1}^{n} \alpha_i y_i = 0\)

You can replace this inner product with another one, **without even knowing \(\Phi\).**

In fact there can be many different feature maps that correspond to the same
inner product.

In other words, we’ll replace \( \mathbf{x}_i^T \mathbf{x}_i \) (i.e., \( \langle \mathbf{x}_i, \mathbf{x}_i \rangle_{\mathbb{R}^p} \)) with \( k(\mathbf{x}_i, \mathbf{x}_i) \), where \( k \) happens to be an inner product in some feature space, \( k(\mathbf{x}_i, \mathbf{x}_i) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i) \rangle_{\mathcal{H}_k} \). Note that \( k \) is also called the kernel (you’ll see why later).

**Example 3:** We could make \( k(\mathbf{x}, \mathbf{z}) \) the square of the usual inner product:

\[
k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2_{\mathbb{R}^p} = \left( \sum_{j=1}^{p} x^{(j)} z^{(j)} \right)^2 = \sum_{j=1}^{p} \sum_{\ell=1}^{p} x^{(j)} x^{(\ell)} z^{(j)} z^{(\ell)}.
\]

But how do I know that the square of the inner product is itself an inner product in some feature space? We’ll show a general way to check this later, but for now, let’s see if we can find such a feature space.

Well, for \( p = 2 \), if we used \( \Phi \left( [x^{(1)}, x^{(2)}] \right) = [x^{(1)}x^{(2)}, x^{(1)}x^{(2)}, x^{(1)}x^{(2)}] \) to map into a 4d feature space, then the inner product would be:

\[
\Phi(\mathbf{x})^T \Phi(\mathbf{z}) = x^{(1)}z^{(1)} + x^{(2)}z^{(2)} + 2x^{(1)}x^{(2)}z^{(1)}z^{(2)} = \langle \mathbf{x}, \mathbf{z} \rangle^2_{\mathbb{R}^2}.
\]

So we showed that \( k \) is an inner product for \( p = 2 \) because we found a feature space corresponding to it.

For \( p = 3 \) we can also find a feature space, namely the 9d feature space from Example 2 would give us the inner product \( k \).

That is,

\[
\Phi(\mathbf{x}) = (x^{(1)}x^{(2)}, ..., x^{(3)}), \quad \text{and} \quad \Phi(\mathbf{z}) = (z^{(1)}z^{(2)}, ..., z^{(3)}),
\]

\[
\langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle_{\mathbb{R}^9} = \langle \mathbf{x}, \mathbf{z} \rangle^2_{\mathbb{R}^3}.
\]

That’s nice.

We can even add a positive constant \( c \geq 0 \), so that \( k \) is the inner product plus a constant squared.
**Example 4:**

\[
k(x, z) = (x^T z + c)^2 = \left( \sum_{j=1}^{p} x^{(j)} z^{(j)} + c \right) \left( \sum_{\ell=1}^{p} x^{(\ell)} z^{(\ell)} + c \right)
= \sum_{j=1}^{p} \sum_{\ell=1}^{p} x^{(j)} x^{(\ell)} z^{(j)} z^{(\ell)} + 2c \sum_{j=1}^{p} x^{(j)} z^{(j)} + c^2
= \sum_{j,\ell=1}^{p} (x^{(j)} x^{(\ell)})(z^{(j)} z^{(\ell)}) + \sum_{j=1}^{p} (\sqrt{2cx^{(j)}})(\sqrt{2cz^{(j)})} + c^2,
\]

and in \( p = 3 \) dimensions, one possible feature map is:

\[
\Phi(x) = [x^{(1)^2}, x^{(1)}x^{(2)}, \ldots, x^{(3)^2}, \sqrt{2cx^{(1)}}, \sqrt{2cx^{(2)}}, \sqrt{2cx^{(3)}}, c]
\]

and \( c \) controls the relative weight of the linear and quadratic terms in the inner product.

Even more generally, if you wanted to, you could choose the kernel to be any higher power of the regular inner product.

**Example 5:** For any integer \( d \geq 2 \)

\[
k(x, z) = (x^T z + c)^d,
\]

where the feature space \( \Phi(x) \) will contain all monomials up to and including degree \( d \).

The decision boundary in the feature space (of course) is a hyperplane, whereas in the input space it’s a polynomial of degree \( d \). Now do you understand that figure at the beginning of the lecture?

Because these kernels give rise to polynomial decision boundaries, they are called *polynomial kernels*. They are very popular.

Here’s the crescent full moon dataset:
Here is a polynomial kernel that fully separates it. I’ve shown a side view so you can see how it works.

Beyond these examples, it is possible to construct very complicated kernels, even ones that have infinite dimensional feature spaces, that provide a lot of modeling power. I’ll show an example of the spiral data with a gaussian kernel below, which has an infinite-dimensional feature space.
SVMs with these kinds of fancy kernels are really cool.

But the solution to the optimization problem is still a simple linear combination, even if the feature space is very high dimensional.

**How do I evaluate \( f(x) \) for a test example \( x \) then?**

Recall that the SVM derivation started with a linear model: \( f(x) = \sum_j \lambda_j^* x(j) + \lambda_0^* \). We found that \( \lambda^* = \sum_i \alpha_i^* y_i x_i \) in the linear case, and that \( \lambda_0^* \) can be obtained from any of the positive support vectors (call it \( i_{sv} \)) by \( \lambda_0^* = 1 - \lambda^* T x_{isv} \). If we use a plain linear kernel, this means:

\[
    f(x^{\text{new}}) = \sum_j \lambda_j^* x^{\text{new}}(j) + \lambda_0^*.
\]

But when we’re using a kernel, we actually want to work with the \( \alpha_i^* \)’s. So let’s change that back:

\[
    f(x^{\text{new}}) = \sum_j \sum_i \alpha_i^* y_i x_i^{(j)} x^{\text{new}}(j) + \lambda_0^* \\
    = \sum_i \alpha_i^* y_i x_i \cdot x^{\text{new}} + \lambda_0^*.
\]

Let’s replace the inner product with the new one!

\[
    = \sum_i \alpha_i^* y_i k(x_i, x^{\text{new}}) + \lambda_0^*.
\]

Then we need to do just a little more work on \( \lambda_0^* \):

\[
    \lambda_0^* = 1 - \lambda^* T x_{isv} = 1 - (\sum_i \alpha_i^* y_i x_i) T x_{isv} = 1 - \sum_i \alpha_i^* y_i x_i^T x_{isv},
\]

which happily turns into the following when we replace the inner product with our kernel:

\[
    \lambda_0^* = 1 - \sum_i \alpha_i^* y_i k(x_i, x_{isv}).
\]

So we can evaluate \( f(x) \) without having to know what the map is to the new feature space.
If we’re going to replace $\mathbf{x}_i^T \mathbf{x}_k$ everywhere with some function of $\mathbf{x}_i$ and $\mathbf{x}_k$ that is hopefully an inner product from some other space (a kernel), we need to ensure that it really is an inner product. More generally, we’d like to know how to construct functions that are guaranteed to be inner products in some space. We need to know some functional analysis to do that.

---

**Roadmap**

1. make some definitions (inner product, Hilbert space, kernel)
2. give some intuition by doing a calculation in a space with a finite number of states
3. design a general Hilbert space whose inner product is the kernel
4. show it has a reproducing property - now it’s a Reproducing Kernel Hilbert space
5. create a totally different representation of the space, which is a more intuitive to express the kernel (similar to the finite state one)
6. prove a cool representer theorem for SVM-like algorithms
7. show you some nice properties of kernels, and how you might construct them

**Definitions**

An *inner product* takes two elements of a vector space $\mathcal{X}$ and outputs a number. An inner product could be a usual dot product: $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v} = \sum_i u^{(i)} v^{(i)}$, or it could be something fancier. An inner product $\langle \cdot, \cdot \rangle$ must satisfy the following conditions:

1. Symmetry
   
   $\langle u, v \rangle = \langle v, u \rangle \; \forall u, v \in \mathcal{X}$

2. Bilinearity

   $\langle \alpha u + \beta v, w \rangle = \alpha \langle u, w \rangle + \beta \langle v, w \rangle \; \forall u, v, w \in \mathcal{X}, \forall \alpha, \beta \in \mathbb{R}$
3. Strict Positive Definiteness

\[ \langle u, u \rangle \geq 0 \quad \forall x \in \mathcal{X} \]

\[ \langle u, u \rangle = 0 \iff u = 0. \]

An inner product space (or pre-Hilbert space) is a vector space together with an inner product.

A Hilbert space is a complete inner product space. (‘Complete’ means sequences converge to elements of the space - there aren’t any “holes” in the space.)

Examples of Hilbert spaces include:

- The vector space \( \mathbb{R}^p \) with \( \langle u, v \rangle_{\mathbb{R}^p} = u^T v \), the vector dot product of \( u \) and \( v \).

- The space \( \ell_2 \) of square summable sequences, with inner product \( \langle u, v \rangle_{\ell_2} = \sum_{i=1}^{\infty} u_i v_i \)

- The space \( L_2(\mathcal{X}, \mu) \) of square integrable functions, that is, functions \( f \) such that \( \int f(x)^2 d\mu(x) < \infty \), with inner product \( \langle f, g \rangle_{L_2(\mathcal{X}, \mu)} = \int f(x)g(x)d\mu(x) \).

As a preview of what is coming later, a reproducing kernel Hilbert space (RKHS) has an inner product and a special function \( k \) that work together to provide a special property – the reproducing property. Specifically, \( k(\cdot, \cdot) \) is a reproducing kernel of Hilbert space \( \mathcal{H} \) if for all \( f \in \mathcal{H} \),

\[ f(x) = \langle k(x, \cdot), f(\cdot) \rangle_{\mathcal{H}}, \]

where \( \langle \cdot, \cdot \rangle_{\mathcal{H}} \) is the inner product in the Hilbert space. If we set \( f_z(\cdot) = k(\cdot, z) \), then the reproducing property will give us that \( k \) is exactly the inner product: \( f_z(x) = \langle k(x, \cdot), k(\cdot, z) \rangle_{\mathcal{H}} \) and from the definition of \( f_z(\cdot) \), we get that \( f_z(x) = k(x, z) \). So we will see that \( k(x, z) \) is the inner product of functions \( k(x, \cdot) \) and \( k(\cdot, z) \), which are going to be exactly the feature maps \( \Phi(x) \) and \( \Phi(z) \). So to put the preview all together, we’ll see that \( k(x, z) \) is the inner product of \( \Phi(x) \) and \( \Phi(z) \), as long as everything is defined properly.

Now that I’ve gotten way ahead of myself, let’s back up and think about what properties are absolutely necessary when we define function \( k \).
A Finite World

Say we have a finite input space \{x_1, \ldots, x_m\}. So there’s only \( m \) possible states for the \( x_i \)’s. You might think this is unrealistic because even in 2D, there are an infinite number of points. While that’s completely true, as it turns out, assuming a finite state space is actually not unrealistic at all! Think about discrete feature spaces rather than continuous ones. Try thinking of all features being binary so there are \( m = 2^p \) possible vectors for \( x_i \). Think about predicting stroke from a combination of age, gender, whether someone has past history of stroke, whether they take blood thinner, whether they have congestive heard failure, and whether they have hypertension. Given that there are at most about 120 age values, 2 gender values (putting non-binary genders into the riskier group which in this case is females), 2 values each for each of the other four features, the whole feature space is of size \( 120 \times 2 \times 2 \times 2 \times 2 \times 2 \), which is definitely finite! We will get a lot of intuition about general properties using a finite space.

I want to be able to take inner products between any two elements of the space using function \( k \) as the inner product. Inner products by definition are symmetric, so \( k(x_i, x_l) = k(x_l, x_i) \). In other words, in order for us to even consider \( k \) as a valid kernel function, the matrix:

\[
K = \begin{bmatrix}
1 & l & m \\
1 & k(x_i, x_l) & 1 \\
1 & l & m
\end{bmatrix}
\]

needs to be symmetric. This means we can diagonalize it, and the eigendecomposition takes this form:

\[
K = V \Lambda V'
\]

where \( V \) is an orthogonal matrix where the columns of \( V \) are eigenvectors, \( v_t \), and \( \Lambda \) is a diagonal matrix with eigenvalues \( \lambda_t \) on the diagonal. This fact (that real symmetric matrices can always be diagonalized) isn’t difficult to prove, but requires some linear algebra.
For now, assume all \( \lambda_t \)'s are nonnegative, and consider this feature map:

\[
\Phi(x_i) = [\sqrt{\lambda_1}v_1^{(i)}, ..., \sqrt{\lambda_l}v_l^{(i)}, ..., \sqrt{\lambda_m}v_m^{(i)}].
\]

(writing it for \( x_l \) too):

\[
\Phi(x_l) = [\sqrt{\lambda_1}v_1^{(l)}, ..., \sqrt{\lambda_l}v_l^{(l)}, ..., \sqrt{\lambda_m}v_m^{(l)}].
\]

With this choice, \( k \) is just a dot product in \( \mathbb{R}^m \):

\[
\langle \Phi(x_i), \Phi(x_l) \rangle_{\mathbb{R}^m} = \sum_{i=1}^{m} \lambda_t v_t^{(i)} v_t^{(l)} = (V\Lambda V')_{il} = K_{il} = k(x_i, x_l).
\]

Why did we need the \( \lambda_t \)'s to be nonnegative? Say \( \lambda_s < 0 \). Form a point in feature space that is a special linear combination of the \( \Phi(x_i)'s:\n
\[
z = \sum_{i=1}^{m} v_s^{(i)} \Phi(x_i). \quad \text{(coeffs are elements of eigenvector } v_s)\]

Then calculate

\[
\|z\|_2^2 = \langle z, z \rangle_{\mathbb{R}^m} = \sum_i \sum_l v_s^{(i)} \Phi(x_i)^T \Phi(x_l) v_s^{(l)} = \sum_i \sum_l v_s^{(i)} K_{il} v_s^{(l)}
\]

\[
= v_s^T K v_s = \lambda_s < 0
\]

which conflicts with the geometry of the feature space.

We just showed that if \( k \) has any chance of being an inner product in a feature space, then matrix \( K \) needs to be positive semi-definite (needs to have non-negative eigenvalues).

In fact, we’ll just define kernels in the first place to have positive semi-definite Gram matrices, since they can’t be inner products without that. In the infinite state case, we can’t write out a Gram matrix (like \( K \) in the finite state case) for the whole space, because the \( x \)'s can take infinitely many values. We’ll just get to pick \( m \) observations - we want to make sure the Gram matrix for those observations is positive semi-definite, no matter what observations we get!

Let us officially define a kernel. A function \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \) is a kernel if
\begin{itemize}
\item $k$ is symmetric: $k(x, z) = k(z, x)$.
\item $k$ gives rise to a positive semi-definite “Gram matrix,” i.e., for any number
of states $m \in \mathbb{N}$ and any set of states $x_1, \ldots, x_m$ chosen from $\mathcal{X}$, the Gram
matrix $K$ defined by $K_{ii} = k(x_i, x_i)$ is positive semidefinite.
\end{itemize}

Another way to show that a matrix $K$ is positive semi-definite is to show that
\begin{equation}
\forall c \in \mathbb{R}^m, c^T K c \geq 0.
\end{equation}
(This is equivalent to all the eigenvalues being nonnegative, which again is not
hard to show but requires some calculations.)

Here are some nice properties of $k$:
\begin{itemize}
\item $k(u, u) \geq 0$ (Think about the Gram matrix of $m = 1$.)
\item $k(u, v) \leq \sqrt{k(u, u)k(v, v)}$ (This is the Cauchy-Schwarz inequality.)
\end{itemize}

The second property is not hard to show for $m = 2$. The Gram matrix
\begin{equation}
K = \begin{pmatrix} k(u, u) & k(u, v) \\
                   k(v, u) & k(v, v) \end{pmatrix}
\end{equation}
is positive semi-definite whenever $c^T K c \geq 0 \ \forall c$. Choose in particular
\begin{equation}
c = \begin{bmatrix} k(v, v) \\
                        -k(u, v) \end{bmatrix}.
\end{equation}

Then since $K$ is positive semi-definite,
\begin{equation}
0 \leq c^T K c = [k(v, v)k(u, u) - k(u, v)^2]k(v, v)
\end{equation}
(where I skipped a little bit of simplifying in the equality) so we must then have
$k(v, v)k(u, u) \geq k(u, v)^2$. That’s it!

**Building a Rather Special Hilbert Space**

Define $\mathbb{R}^\mathcal{X} := \{ f : \mathcal{X} \to \mathbb{R}\}$, the space of functions that map $\mathcal{X}$ to $\mathbb{R}$. Let’s define
the feature map $\Phi : \mathcal{X} \to \mathbb{R}^\mathcal{X}$ so that it maps $x$ to $k(\cdot, x)$:
\begin{equation}
\Phi : x \mapsto k(\cdot, x).
\end{equation}

So, $\Phi(x)$ is a function, and for a point $z \in \mathcal{X}$, the function assigns $k(z, x)$ to it.
So we turned each $x$ into a function on the domain $\mathcal{X}$. Those functions could be thought of as infinite dimensional vectors. These functions will be elements of our Hilbert space. Think about $\Phi(x_i)$ as being a bump centered on $x_i$. Keep in mind here that $\Phi(x_i)$ is a function, so it has an additional argument. I guess you could notate it $\Phi(x_i)(\cdot)$ to show that for your choice of $z$, you could compute $\Phi(x_i)(z)$ and get a number. That number would be $k(z, x_i)$.

We want to have $k$ be an inner product in feature space. To do this we need to:

1. Define feature map $\Phi : \mathcal{X} \to \mathbb{R}^\mathcal{X}$, which we’ve done already. Then we need to turn the image of $\Phi$ into a vector space.

2. Define an inner product $\langle \cdot, \cdot \rangle_{H_k}$.

3. Show that the inner product satisfies:

   $$k(x, z) = \langle \Phi(x), \Phi(z) \rangle_{H_k}.$$ 

That’ll make sure that the feature space is a pre-Hilbert space.

Let’s do the rest of Step 1. Elements in the vector space will look like this, they’ll be in the span of the $\Phi(x_i)$’s:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \leftarrow \text{“vectors”}$$
where \( m, \alpha_i \) and \( x_1, \ldots, x_m \in \mathcal{X} \) can be anything. (We have addition and multiplication, so it’s a vector space.) The vector space is:

\[
\text{span} \left( \{ \Phi(x) : x \in \mathcal{X} \} \right) = \left\{ f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) : m \in \mathbb{N}, x_i \in \mathcal{X}, \alpha_i \in \mathbb{R} \right\}.
\]

For Step 2, let’s grab functions \( f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \) and \( g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j) \), and define the inner product:

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

Is it well-defined?

Well, it’s symmetric, since \( k \) is symmetric:

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

It’s also bilinear, look at this:

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

so we have

\[
\langle f_1 + f_2, g \rangle_{H_k} = \sum_{j=1}^{m'} \beta_j \left( f_1(x'_j) + f_2(x'_j) \right) \quad \text{(use the expression above)}
\]

\[
= \sum_{j=1}^{m'} \beta_j f_1(x'_j) + \sum_{j=1}^{m'} \beta_j f_2(x'_j)
\]

\[
= \langle f_1, g \rangle_{H_k} + \langle f_2, g \rangle_{H_k} \quad \text{(use the expression above again)}.
\]

(We can do the same calculation to show \( \langle f, g_1 + g_2 \rangle_{H_k} = \langle f, g_1 \rangle_{H_k} + \langle f, g_2 \rangle_{H_k} \)).

So it’s bilinear.

It’s also positive semi-definite, since \( k \) gives rise to positive semi-definite Gram matrices. To see this, for function \( f \),

\[
\langle f, f \rangle_{H_k} = \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) = \mathbf{a}^T \mathbf{K} \mathbf{a}
\]
and we said earlier in (1) that since $K$ is positive semi-definite, it means that for any $\alpha_i$’s we choose, the sum above is $\geq 0$.

So, $k$ is almost a inner product! (What’s missing?) We’ll get to that in a minute.

For Step 3, something totally cool happens, namely that the inner product of $k(\cdot, x)$ and $f$ is just the evaluation of $f$ at $x$.

$$\langle k(\cdot, x), f \rangle_{H_k} = \sum \alpha_i k(x_i, x) = f(x).$$

And then this is a special case:

$$\langle k(\cdot, x), k(\cdot, x') \rangle_{H_k} = k(x, x').$$

This is why $k$ is called a reproducing kernel, and the Hilbert space is a RKHS.

______________________________

Oops! We forgot to show that one last thing, which is that $\langle \cdot, \cdot \rangle_{H_k}$ is strictly positive definite, so that it’s an inner product. Let’s use the reproducing property. For any $x$,

$$|f(x)|^2 = |\langle k(\cdot, x), f \rangle_{H_k}|^2 \leq \langle k(\cdot, x), k(\cdot, x) \rangle_{H_k} \cdot \langle f, f \rangle_{H_k}$$

which means that $\langle f, f \rangle_{H_k} = 0 \Rightarrow f = 0$ for all $x$. For the inequality the line above, we used Cauchy-Schwarz, which I didn’t prove for this case, I only proved it for the case of a single kernel. If Cauchy-Schwarz doesn’t hold, our space is not an inner product space, so Cauchy-Schwarz must hold.

(An interesting thing to think about is whether the norm $\langle f, f \rangle_{H_k}$ can be 0 when the Gram matrix is positive semi-definite but not positive definite. In other words, is it possible that $\langle f, f \rangle_{H_k} = 0$ but the $\alpha_i$’s that are used in $f$ are not 0? I think it is possible, when some of the rows of the Gram matrix be negatives of each other and the $\alpha_i$’s of these two rows are both 1 perhaps. It is important that such cases exist or we would have had to declare the Gram matrices to be positive definite in the first place!)
Ok, let us conclude that our definition for the inner product is positive definite, and thus, a valid inner product.

Now we have an inner product space. In order to make it a Hilbert space, we need to make it complete. The completion is actually not a big deal, we just create a norm:

\[ \|f\|_{H_k} = \sqrt{\langle f, f \rangle_{H_k}} \]

and just add to \( H_k \) the limit points of sequences that converge in that norm. Once we do that,

\[ H_k = \{ f : f = \sum_i \alpha_i k(\cdot, x_i) \} \]

is a Reproducing Kernel Hilbert space (RKHS).

Here’s a formal (simplified) definition of RKHS:

We are given a (compact) \( \mathcal{X} \subseteq \mathbb{R}^d \) and a Hilbert space \( \mathcal{H} \) of functions \( f : \mathcal{X} \rightarrow \mathbb{R} \). We say \( \mathcal{H} \) is a Reproducing Kernel Hilbert Space if there exists a \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \), such that

1. \( k \) has the reproducing property, i.e., \( f(x) = \langle f(\cdot), k(\cdot, x) \rangle_\mathcal{H} \)
2. \( k \) spans \( \mathcal{H} \), that is, \( \mathcal{H} = \text{span}\{k(\cdot, x) : x \in \mathcal{X}\} \)

So when we do the kernel trick, we could think that we’re implicitly mapping \( x \) to \( k(\cdot, x) \) in the RKHS \( H_k \). Neat, huh?

The RKHS we described is the one from the Moore-Aronszajn Theorem (1950) that states that for every positive definite function \( k(\cdot, \cdot) \) there exists a unique RKHS.

What’s not a RKHS?

If you are wondering about a Hilbert space that is not a RKHS, you can think about \( L_2 \), the space of square integrable functions. The function that gives us the reproducing property is the Dirac delta function.

\[ f(x) = \int \delta(x - z)f(z)dz. \]
But the Dirac delta function is not actually in $L_2$, because it is not square-integrable:

$$\int \delta(z)^2 dz \not\rightarrow \text{not finite.}$$

We could stop there, but we won’t. We’re going to define another Hilbert space that has a one-to-one mapping (an isometric isomorphism) to the first one.

**Mercer’s Theorem**

The inspiration of the name “kernel” comes from the study of integral operators, studied by Hilbert and others. Function $k$ which gives rise to an operator $T_k$ via:

$$(T_kf)(x) = \int_{\mathcal{X}} k(x, x') f(x') dx'$$

is called the *kernel* of $T_k$.

Think about an operator $T_k : L_2(\mathcal{X}) \rightarrow L_2(\mathcal{X})$. If you don’t know this notation, don’t worry about it. Just think about $T_k$ eating a function and spitting out another one. Think of $T_k$ as an infinite matrix, which maps infinite dimensional vectors to other infinite dimensional vectors. Remember earlier we analyzed the finite state case? It’s just like that.\(^1\)

And, just like in the finite state case, because $T_k$ is going to be positive semi-definite, it has an eigen-decomposition into eigenfunctions and eigenvalues that are nonnegative.

The following theorem from functional analysis is very similar to what we proved in the finite state case. This theorem is important – it helps people construct kernels (though I admit myself I find the other representation more helpful).

Basically, the theorem says that if we have a kernel $k$ that is positive (defined somehow), we can expand $k$ in terms of eigenfunctions and eigenvalues of a positive operator that comes from $k$. The theorem essentially provides a basis for

\[^1\text{If you know the notation, you’ll see that I’m missing the measure on } L_2 \text{ in the notation. Feel free to put it back in as you like.}\]
the RKHS.

**Mercer’s Theorem (Simplified)** Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^n$. Suppose $k$ is a continuous symmetric function such that the integral operator $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ defined by

$$(T_k f)(\cdot) = \int_{\mathcal{X}} k(\cdot, x) f(x) dx$$

is positive; which here means $\forall f \in L_2(\mathcal{X})$,

$$\int_{\mathcal{X} \times \mathcal{X}} k(x, z) f(x) f(z) dx dz \geq 0,$$

then we can expand $k(x, z)$ in a uniformly convergent series in terms of $T_k$’s eigenfunctions $\psi_j \in L_2(\mathcal{X})$, normalized so that $\|\psi\|_{L_2} = 1$, and positive associated eigenvalues $\lambda_j \geq 0$,

$$k(x, z) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(z).$$

The definition of positive semi-definite here is equivalent to the ones we gave earlier.²

This looks like what we did in the finite case when we decomposed the Gram matrix, using the fact that it is symmetric; there we also saw the Gram matrix $K$ as a matrix multiplication of eigenvectors and eigenvalues. We could define a feature map – as in the finite case – this way:

$$\Phi(x) = [\sqrt{\lambda_1} \psi_1(x), \ldots, \sqrt{\lambda_j} \psi_j(x), \ldots].$$

So that’s a cool property of the RKHS that we can define a feature space according to Mercer’s Theorem. Now for another cool property of SVM problems, having to do with RKHS.

²To see the equivalence, for this direction $\Leftarrow$, choose $f$ as a weighted sum of $\delta$ functions at each $x$. For this direction $\Rightarrow$, say that $\int_{\mathcal{X} \times \mathcal{X}} k(x, z) f(x) f(z) dx dz \geq 0$ doesn’t hold for some $f$ and show the contrapositive. Approximate the integral with a finite sum over a mesh of inputs $\{x_1, \ldots, x_m\}$ chosen sufficiently finely, then let $v$ be the values of $f$ on the mesh, and as long as the mesh is fine enough, we’ll have $v^T K v < 0$, so $K$ isn’t positive semi-definite.
Representer Theorem

Recall that the SVM optimization problem can be expressed as follows:

\[ f^* = \arg\min_{f \in H_k} R^{\text{train}}(f) \]

where

\[ R^{\text{train}}(f) := \sum_{i=1}^{n} \text{hingeloss}(f(x_i), y_i) + C\|f\|^2_{H_k}. \]

Or, you could even think about using a generic loss function:

\[ R^{\text{train}}(f) := \sum_{i=1}^{n} \ell(f(x_i), y_i) + C\|f\|^2_{H_k}. \]

You can pick whatever loss you’d like. Also, let’s use a generic regularization term:

\[ R^{\text{train}}(f) := \sum_{i=1}^{n} \ell(f(x_i), y_i) + \Omega(\|f\|^2_{H_k}), \]

where \( \Omega : \mathbb{R} \rightarrow \mathbb{R} \) is nondecreasing function, so that it can regularize \( \|f\|^2_{H_k} \).

The following theorem is a big surprise. It says that the solutions to any problem of this type - no matter what the loss function is! - all have a solution in a rather nice form.

**Representer Theorem** *(Kimeldorf and Wahba, 1971, Simplified)* Fix a set \( \mathcal{X} \), a kernel \( k \), and let \( H_k \) be the corresponding RKHS. Let \( \Omega : \mathbb{R} \rightarrow \mathbb{R} \) be a nondecreasing function. For any function \( \ell : \mathbb{R}^2 \rightarrow \mathbb{R} \), the solutions of the optimization problem:

\[ f^* \in \arg\min_{f \in H_k} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \Omega(\|f\|^2_{H_k}) \]

can all be expressed in the form:

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

This shows that to solve the SVM optimization problem, we need only to solve for the \( \alpha_i \), which agrees with the solution from the Lagrangian formulation for
SVM. It says that even if we’re trying to solve an optimization problem in an infinite dimensional space \( H_k \), where an arbitrary loss centered on arbitrary \( x'_i \)‘s, then the solution lies in the span of the \( n \) kernels centered on the \( x_i \)’s.

I hope you grasp how cool this is!

Proof. Suppose we project \( f \) onto the subspace:

\[
\text{span}\{k(x_i, \cdot) : 1 \leq i \leq n\}
\]

obtaining \( f_s \) (the component along the subspace) and \( f_\perp \) (the component perpendicular to the subspace). We have:

\[
f = f_s + f_\perp \Rightarrow \|f\|_{H_k}^2 = \|f_s\|_{H_k}^2 + \|f_\perp\|_{H_k}^2 \geq \|f_s\|_{H_k}^2.
\]

This implies that \( \|f\|_{H_k} \), and thus \( \Omega(\|f\|_{H_k}^2) \), is minimized if \( f \) lies in the subspace.

\[
\Omega(\|f\|_{H_k}^2) \geq \Omega(\|f_s\|_{H_k}^2).
\]

Furthermore, since the kernel \( k \) has the reproducing property, we have for each \( i \):

\[
f(x_i) = \langle f, k(x_i, \cdot) \rangle_{H_k} = \langle f_s, k(x_i, \cdot) \rangle_{H_k} + \langle f_\perp, k(x_i, \cdot) \rangle_{H_k} = \langle f_s, k(x_i, \cdot) \rangle_{H_k} = f_s(x_i).
\]

So,

\[
\sum_{i=1}^{n} \ell(f(x_i), y_i) = \sum_{i=1}^{n} \ell(f_s(x_i), y_i).
\]

In other words, the loss depends only on the component of \( f \) lying in the subspace. To minimize, we can just let \( \|f_\perp\|_{H_k} = 0 \) and we can express the minimizer as:

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

That’s it! ■

**Constructing Kernels**

Let’s construct new kernels from previously defined kernels. Suppose we have \( k_1 \) and \( k_2 \). Then the following are also valid kernels:
1. \( k(x, z) = \alpha k_1(x, z) + \beta k_2(x, z) \) for \( \alpha, \beta \geq 0 \)

**Proof.** \( k_1 \) has its feature map \( \Phi_1 \) and inner product \( \langle \cdot, \cdot \rangle_{H_{k_1}} \) and \( k_2 \) has its feature map \( \Phi_2 \) and inner product \( \langle \cdot, \cdot \rangle_{H_{k_2}} \). By linearity, we can have:

\[
\alpha k_1(x, z) = \langle \sqrt{\alpha} \Phi_1(x), \sqrt{\alpha} \Phi_1(z) \rangle_{H_{k_1}} \quad \text{and} \quad \beta k_2(x, z) = \langle \sqrt{\beta} \Phi_2(x), \sqrt{\beta} \Phi_2(z) \rangle_{H_{k_2}}.
\]

Then:

\[
k(x, z) = \alpha k_1(x, z) + \beta k_2(x, z) = \langle \sqrt{\alpha} \Phi_1(x), \sqrt{\alpha} \Phi_1(z) \rangle_{H_{k_1}} + \langle \sqrt{\beta} \Phi_2(x), \sqrt{\beta} \Phi_2(z) \rangle_{H_{k_2}}
\]

and that means that \( k(x, z) \) can be expressed as an inner product, and we have given its feature map.

2. \( k(x, z) = k_1(x, z) k_2(x, z) \)

Proof omitted.

3. \( k(x, z) = k_1(h(x), h(z)) \), where \( h : \mathcal{X} \to \mathcal{X} \).

**Proof.** Since \( h \) is a transformation in the same domain, \( k \) is simply a different kernel in that domain:

\[
k(x, z) = k_1(h(x), h(z)) = \langle \Phi(h(x)), \Phi(h(z)) \rangle_{H_{k_1}} =: \langle \Phi_h(x), \Phi_h(z) \rangle_{H_{\text{new}}}
\]

4. \( k(x, z) = g(x) g(z) \) for \( g : \mathcal{X} \to \mathbb{R} \).

Proof omitted.

5. \( k(x, z) = h(k_1(x, z)) \) where \( h \) is a polynomial with positive coefficients

**Proof.** Since each polynomial term is a product of kernels with a positive coefficient, the proof follows by applying construction principles 1 and 2.

6. \( k(x, z) = \exp(k_1(x, z)) \)

**Proof** Since:

\[
\exp(z) = \lim_{i \to \infty} \left(1 + z + \cdots + \frac{z^i}{i!}\right)
\]

the proof basically follows from 5 and then taking the limit. Think about \( \frac{1}{i!} z^i \) as a term in a polynomial. When you construct the kernel matrix, each element of it is a limit of a sum of these terms. Since each Gram matrix in the sequence is positive semi-definite, so is the limiting one.
7. \( k(\mathbf{x}, \mathbf{z}) = \exp \left( \frac{-\|\mathbf{x} - \mathbf{z}\|_{\ell_2}^2}{\sigma^2} \right) \) “Gaussian kernel”

Proof.

\[
\begin{align*}
k(\mathbf{x}, \mathbf{z}) &= \exp \left( \frac{-\|\mathbf{x} - \mathbf{z}\|_{\ell_2}^2}{\sigma^2} \right) = \exp \left( \frac{-\|\mathbf{x}\|_{\ell_2}^2 - \|\mathbf{z}\|_{\ell_2}^2 + 2\mathbf{x}^T\mathbf{z}}{\sigma^2} \right) \\
&= \left( \exp \left( \frac{-\|\mathbf{x}\|_{\ell_2}^2}{\sigma^2} \right) \right) \exp \left( \frac{-\|\mathbf{z}\|_{\ell_2}^2}{\sigma^2} \right) \exp \left( \frac{2\mathbf{x}^T\mathbf{z}}{\sigma^2} \right) \\
&= g(\mathbf{x})g(\mathbf{z}) \exp(k_1(\mathbf{x}, \mathbf{z})).
\end{align*}
\]

where \( g(\mathbf{x})g(\mathbf{z}) \) is a kernel according to 4, and \( \exp(k_1(\mathbf{x}, \mathbf{z})) \) is a kernel according to 6 (and \( \mathbf{x}^T\mathbf{z} \) is a valid linear kernel, and \( 2/\sigma^2 \) is positive). Finally, according to 2, the product of kernels is a valid kernel.

Note that the Gaussian kernel is translation invariant, since it only depends on \( \mathbf{x} - \mathbf{z} \).

This picture helps with the intuition about Gaussian kernels if you think of the first representation of the RKHS I showed you:

\[
\Phi(\mathbf{x}) = k(\mathbf{x}, \cdot) = \exp \left( \frac{-\|\mathbf{x} - \cdot\|_{\ell_2}^2}{\sigma^2} \right)
\]

So just to summarize, from SVM, we get values of \( f(\mathbf{x}) \) like this for the gaussian kernel:

\[
f(\mathbf{x}) = \sum_i \alpha_i^* y_i k(\mathbf{x}_i, \mathbf{x}) + \lambda_0^*, \quad \text{with} \quad \lambda_0^* = 1 - \sum_i \alpha_i^* y_i k(\mathbf{x}_i, \mathbf{x}_{\text{sv}}).
\]

where

\[
\Phi(\mathbf{x}_i) = k(\mathbf{x}_i, \mathbf{x}) = \exp \left( \frac{-\|\mathbf{x}_i - \mathbf{x}\|_{\ell_2}^2}{\sigma^2} \right).
\]
Gaussian kernels can be very powerful:
Don’t make the $\sigma^2$ too small or you’ll overfit!