

# Kernel Regression

Throughout this course, we have (slowly) come to appreciate how having a *basis* in place helps discretize linear inverse problems that involve *functions* (or mappings) of a continuous variable.

In these notes, we explore a closely related, but ultimately different, concept: *kernels*. There are many different ways we might motivate and interpret the mathematics that comes below; one is that working with a kernel gives you a kind of *data adaptive* basis.

As with our discussion about bases, the concept of a Hilbert space will be at the forefront. Unlike bases, our discussion will benefit tremendously by our ability to work in abstract Hilbert spaces  $\mathcal{H}$  with non-standard inner products.

We start by recalling our central regression problem: we are given pairs of data points  $(\mathbf{t}_m, y_m)$ ,  $m = 1, \dots, M$  with  $\mathbf{t}_m \in \mathbb{R}^D$  and  $y_m \in \mathbb{R}$ , and we want to find a mapping  $\mathbf{f} : \mathbb{R}^D \rightarrow \mathbb{R}$  such that

$$f(\mathbf{t}_m) \approx y_m, \quad m = 1, \dots, M.$$

As we have seen, making this problem well-posed relies critically on having some kind of *model* for  $\mathbf{f}$ .

When we use linear regression, our model is that  $f(\mathbf{t})$  can be written as a linear combination of entries of  $\mathbf{t}$ . That is, there exists<sup>1</sup>  $\mathbf{w} \in \mathbb{R}^D$  such that

$$f(\mathbf{t}) = \langle \mathbf{t}, \mathbf{w} \rangle = \mathbf{w}^T \mathbf{t} = w_1 t[1] + w_2 t[2] + \cdots + w_D t[D].$$

## Regression using a finite dimensional basis

When we use nonlinear regression with a basis, our model is that  $\mathbf{f}$  lives in a linear subspace of dimension  $N$ :

$$\mathbf{f} \in \text{Span}(\{\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_N\}),$$

i.e.

$$f(\mathbf{t}) = \sum_{n=1}^N w_n \psi_n(\mathbf{t}), \quad \forall \mathbf{t} \in \mathbb{R}^D.$$

Another way of interpreting regression in a basis is that we take a data point  $\mathbf{t} \in \mathbb{R}^D$ , map it into  $\mathbb{R}^N$  using  $\boldsymbol{\Psi} : \mathbb{R}^D \rightarrow \mathbb{R}^N$ , where

$$\boldsymbol{\Psi}(\mathbf{t}) = \begin{bmatrix} \psi_1(\mathbf{t}) \\ \psi_2(\mathbf{t}) \\ \vdots \\ \psi_N(\mathbf{t}) \end{bmatrix},$$

then take a linear combination of the  $\boldsymbol{\Psi}(\mathbf{t})$ ; that is,  $\mathbf{f}$  can be written as

$$f(\mathbf{t}) = \langle \boldsymbol{\Psi}(\mathbf{t}), \mathbf{w} \rangle,$$

---

<sup>1</sup>Two comments about notation here: 1) We are using  $\mathbf{w}$  here as the weight vector where we previously used  $\mathbf{x}$ . This is simply to keep things more consistent with most of the literature on this topic. 2) We will use  $t[1], \dots, t[D]$  to index the entries of a data point  $\mathbf{t}$ . So the  $i$ th entry of  $\mathbf{t}_m$  is  $t_m[i]$ .

for some  $\mathbf{w} \in \mathbb{R}^N$ . In the end, the computational problem we have to solve is a linear inverse problem, even though the mapping  $\Psi(\cdot)$  can be highly non-linear in itself.

Let's quickly review how we might fit such an  $\mathbf{f}$  given the data  $\{(\mathbf{t}_m, y_m)\}_{m=1}^M$ . One method is through *ridge regression*, where we solve

$$\underset{\mathbf{w} \in \mathbb{R}^N}{\text{minimize}} \sum_{m=1}^M \left( y_m - \sum_{n=1}^N w_n \psi_n(\mathbf{t}_m) \right)^2 + \delta \|\mathbf{w}\|_2^2, \quad (1)$$

which is the same as

$$\underset{\mathbf{w} \in \mathbb{R}^N}{\text{minimize}} \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 + \delta \|\mathbf{w}\|_2^2,$$

where

$$\mathbf{A} = \begin{bmatrix} -\Psi(\mathbf{t}_1)^T - \\ -\Psi(\mathbf{t}_2)^T - \\ \vdots \\ -\Psi(\mathbf{t}_M)^T - \end{bmatrix}$$

We know that the solution to this problem is given by,

$$\hat{\mathbf{w}} = \mathbf{A}^T \hat{\boldsymbol{\alpha}}, \quad \text{where} \quad \hat{\boldsymbol{\alpha}} = \left( \mathbf{A}\mathbf{A}^T + \delta \mathbf{I} \right)^{-1} \mathbf{y}.$$

With  $\mathbf{K} = \mathbf{A}\mathbf{A}^T$ , we compute the solution by solving an  $M \times M$  (positive definite) system of equations with the matrix

$$\mathbf{K} + \delta \mathbf{I}, \quad K_{i,j} = \langle \Psi(\mathbf{t}_j), \Psi(\mathbf{t}_i) \rangle, \quad (2)$$

and then take

$$\hat{\mathbf{w}} = \sum_{m=1}^M \hat{\alpha}_m \Psi(\mathbf{t}_m). \quad (3)$$

Note in particular that the optimal weights  $\hat{\mathbf{w}}$  are a linear combination of the *mapped* data points  $\Phi(\mathbf{t}_m)$ .

The vector  $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^M$  also provides a parameterization for our “best fit” function  $\hat{f}$ . Given an arbitrary  $\mathbf{t}$ , we can compute

$$\hat{f}(\mathbf{t}) = \langle \Psi(\mathbf{t}), \hat{\mathbf{w}} \rangle = \sum_{m=1}^M \hat{\alpha}_m \langle \Psi(\mathbf{t}), \Psi(\mathbf{t}_m) \rangle.$$

Notice that all the real work is being done in  $\mathbb{R}^M$  above. This allows us to take  $N$  very large, as the following example illustrates.

### **Example: Multidimensional Polynomials.**

Consider the mapping

$$\Psi(\mathbf{t}) = \begin{bmatrix} 1 \\ \sqrt{2} t[1] \\ \vdots \\ \sqrt{2} t[D] \\ t[1]^2 \\ \vdots \\ t[D]^2 \\ \sqrt{2} t[1]t[2] \\ \vdots \\ \sqrt{2} t[D-1]t[D] \end{bmatrix}$$

This corresponds to using a basis that can build up all linear and quadratic functions on  $\mathbb{R}^D$ . The number of basis functions we need to do this is  $N = 1 + D + D + D(D-1)/2 = (D^2 + 3D + 2)/2$ . So unlike the one-dimensional case ( $D = 1$ ) where we only need three coefficients to specify a quadratic, this value of  $N$  can be quite large.

Fortunately, we do not need to work in  $\mathbb{R}^N$  at all to solve the ridge regression problem. A quick calculation shows that

$$(\langle \mathbf{t}_i, \mathbf{t}_j \rangle + 1)^2 = (\mathbf{t}_j^T \mathbf{t}_i)^2 + 2\mathbf{t}_j^T \mathbf{t}_i + 1 = \langle \Psi(\mathbf{t}_i), \Psi(\mathbf{t}_j) \rangle.$$

That is, the inner products we need to compute in  $\mathbb{R}^N$  are simple functions of inner products in the much smaller dimensional space  $\mathbb{R}^D$ .

This effect gets even more dramatic when we move to higher order polynomials. To use a basis that can build up all polynomial functions on  $\mathbb{R}^D$  of at most  $p$ , the mapping becomes

$$\Psi(\mathbf{t}) = \begin{bmatrix} \vdots \\ \vdots \\ \sqrt{\binom{p'}{j_1, \dots, j_D}} t[1]^{j_1} \cdots t[D]^{j_D} \\ \vdots \\ \vdots \end{bmatrix}$$

with entries for all values of  $p' \leq p$  and  $(j_1, \dots, j_D)$  such that  $j_1 + \cdots + j_D = p'$ . This makes the dimension  $N \sim D^p$ . The constant in front of the  $t_1^{j_1} \cdots t_D^{j_D}$  is the multinomial coefficient

$$\binom{p'}{j_1, \dots, j_D} = \frac{p'!}{j_1! j_2! \cdots j_D!}$$

But again, we are saved from this enormous  $N$  by a trick for computing the necessary inner products. Using the multinomial theorem<sup>2</sup>, we can show that

$$(\langle \mathbf{t}_i, \mathbf{t}_j \rangle + 1)^p = \langle \Phi(\mathbf{t}_i), \Phi(\mathbf{t}_j) \rangle.$$

---

<sup>2</sup>The Wikipedia article on this is decent, [https://en.wikipedia.org/wiki/Multinomial\\_theorem](https://en.wikipedia.org/wiki/Multinomial_theorem).

So again, all the computations for creating the  $M \times M$  matrix  $\mathbf{K}$  can take place in  $\mathbb{R}^D$ . The trick above makes the cost of solving (1) depend only on the dimension  $D$  and the number of observations  $M$ , and not at all on the dimension of our model.

## Regression using an infinite dimensional mapping

In the examples above, we saw how we could map data points  $\mathbf{t}_1, \dots, \mathbf{t}_M \in \mathbb{R}^D$  into  $\mathbb{R}^N$ , where  $N$  could be much, much larger than both  $D$  and  $M$ . In this section, we take this idea to the extreme by mapping each data point into an infinite dimensional Hilbert space. Like the examples above, if we do this carefully, the fact that we are working in infinite dimensions will not affect the computational resources we need to estimate  $f$ .

Let's start by considering a particular example. Let  $\phi : \mathbb{R}^D \rightarrow \mathbb{R}$  be the so-called “Gaussian radial basis function”

$$\phi(\mathbf{s}) = \exp\left(-\frac{\|\mathbf{s}\|_2^2}{2\sigma^2}\right).$$

Now, let  $\mathcal{T}$  be some region in  $\mathbb{R}^D$  where we can safely restrict  $\mathbf{t}_m \in \mathcal{T}$  —  $\mathcal{T}$  could be a bounded region like  $[-10, 10]^D$  or even all of  $\mathbb{R}^D$ , the particular choice does not affect our discussion below. Let  $\mathcal{H}$  be the **finite linear span** of all shifts  $\phi(\mathbf{s} - \mathbf{t})$  for  $\mathbf{t} \in \mathcal{T}$ . That is,  $\mathcal{H}$  contains all functions  $h : \mathbb{R}^D \rightarrow \mathbb{R}$  such that

$$h(\mathbf{s}) = \sum_{k=1}^K \alpha_k \phi(\mathbf{s} - \mathbf{t}_k), \quad (4)$$

for any arbitrary  $K$  and  $\mathbf{t}_1, \dots, \mathbf{t}_K \in \mathcal{T}$ . It is a fact that for any distinct set of shifts  $\{\mathbf{t}_k\}$ , the  $\{\phi(\mathbf{s} - \mathbf{t}_k)\}$  will be linearly independent.

As  $K$  is arbitrary, this means that this space is **infinite dimensional** (even if  $\mathcal{T}$  is bounded).

$\mathcal{H}$  is a subspace of  $L_2(\mathbb{R}^D)$ , and it naturally inherits the standard inner product. Given two functions  $h$  as in (4) and  $g$  written as

$$g(\mathbf{s}) = \sum_{\ell=1}^L \beta_{\ell} \phi(\mathbf{s} - \mathbf{t}_{\ell}),$$

we can write this inner product<sup>3</sup> as

$$\langle \mathbf{h}, \mathbf{g} \rangle_{\mathcal{H}} = \sum_{k=1}^K \sum_{\ell=1}^L \alpha_k \beta_{\ell} \langle \phi(\mathbf{s} - \mathbf{t}_k), \phi(\mathbf{s} - \mathbf{t}_{\ell}) \rangle_{\mathcal{H}} = \boldsymbol{\alpha}^T \mathbf{G} \boldsymbol{\beta},$$

where  $\mathbf{G}$  is the  $K \times L$  matrix with entries

$$G_{k,\ell} = \langle \phi(\mathbf{s} - \mathbf{t}_k), \phi(\mathbf{s} - \mathbf{t}_{\ell}) \rangle_{\mathcal{H}} = \int_{\mathbf{s} \in \mathbb{R}^D} \phi(\mathbf{s} - \mathbf{t}_k) \phi(\mathbf{s} - \mathbf{t}_{\ell}) \, d\mathbf{s}.$$

At this point we should mention that for  $\mathcal{H}$  to be a Hilbert space, we really need to take the *closure* of the finite linear span, which essentially means that all members of  $\mathcal{H}$  can be approximated arbitrarily well by functions of the form (4). This is an important detail, but does not really affect how we think about this space qualitatively.

Our model, now, is that  $f(\mathbf{t})$  can be computed by mapping  $\mathbf{t}$  into  $\mathcal{H}$  and then applying a linear functional. For our Gaussian radial basis function example, we use the mapping

$$\Phi : \mathbb{R}^D \rightarrow \mathcal{H}, \quad \Phi(\mathbf{t}) = \phi(\mathbf{s} - \mathbf{t}).$$

---

<sup>3</sup>Again, the  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  inner product here is really just the standard  $L_2(\mathbb{R}^D)$  inner product — in this particular case, we are only writing it this way to emphasize that we are only considering functions in this subspace.

Given the data points  $\{(\mathbf{t}_m, y_m)\}$ , our goal is to write the  $y_m$  as a linear functional evaluated at  $\Phi(\mathbf{t}_m)$ ; that is<sup>4</sup>, we want to find a  $\mathbf{w} \in \mathcal{H}$  such that

$$\langle \Phi(\mathbf{t}_m), \mathbf{w} \rangle_{\mathcal{H}} \approx y_m, \quad m = 1, \dots, M.$$

Note that  $\mathbf{w}$  really is a point in  $\mathcal{H}$ , it is not some finite list of weights as in the examples in the previous section.

To fit the best  $\mathbf{w}$ , we use the Hilbert space analog of ridge regression. We solve the optimization program

$$\underset{\mathbf{w} \in \mathcal{H}}{\text{minimize}} \quad \sum_{m=1}^M (y_m - \langle \Phi(\mathbf{t}_m), \mathbf{w} \rangle_{\mathcal{H}})^2 + \delta \|\mathbf{w}\|_{\mathcal{H}}^2. \quad (5)$$

Compare (5) to (1). At first glance, this optimization problem appears to be very hard to solve, as we are optimizing over an infinite number of variables (we are considering all  $\mathbf{w} \in \mathcal{H}$  which is infinite dimensional). But there is a critical fact that saves us. Just as in finite dimensions (recall (3)), the optimal  $\mathbf{w}$  will lie in the linear span of  $\Phi(\mathbf{t}_1), \dots, \Phi(\mathbf{t}_M)$ . That is, there exists  $\hat{\alpha} \in \mathbb{R}^M$  such that

$$\hat{\mathbf{w}} = \sum_{m=1}^M \hat{\alpha}_m \Phi(\mathbf{t}_m)$$

is the solution to (5). We have seen why this is true for finite-dimensional  $\mathcal{H}$ . That is true for  $\infty$ -dimensional Hilbert spaces is another fundamental result from functional analysis; it often goes by the name of *the Representer Theorem*.

---

<sup>4</sup>As we have mentioned before, there is a major result in functional analysis that says that every (continuous) linear functional on a Hilbert space can be written as an inner product with a fixed vector from that space. This is known as the Riesz Representation Theorem.



Moreover, the vector  $\hat{\boldsymbol{\alpha}}$  that parameterizes the solution is computed in an analogous way to the finite-dimensional case. As in (2), we form the  $M \times M$  matrix  $\mathbf{K}$  with entries

$$K_{\ell,m} = \langle \Phi(t_m), \Phi(t_\ell) \rangle_{\mathcal{H}},$$

and then compute

$$\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \delta \mathbf{I})^{-1} \mathbf{y}.$$

If we have an efficient method for constructing the  $\mathbf{K}$  matrix (i.e. computing the inner products above), then this is no more expensive than computing the solution in finite dimensions.

Let's recap the steps we have taken to solve the problem above. Given the data points  $\{(\mathbf{t}_m, y_m)\}_{m=1}^M$ , we

1. Create the  $\mathbf{K}$  matrix by computing

$$\langle \Phi(t_m), \Phi(t_\ell) \rangle_{\mathcal{H}} \quad \text{for all } m, \ell = 1, \dots, M.$$

2. Solve the symmetric positive-definite system of equations

$$(\mathbf{K} + \delta \mathbf{I}) \hat{\boldsymbol{\alpha}} = \mathbf{y}.$$

This completely defines our best-fit function  $\hat{f}$ .

3. Given another  $\mathbf{t} \in \mathbb{R}^D$ , we can use  $\hat{\boldsymbol{\alpha}}$  to evaluate

$$\hat{f}(\mathbf{t}) = \sum_{m=1}^M \hat{\alpha}_m \langle \Phi(\mathbf{t}), \Phi(\mathbf{t}_m) \rangle_{\mathcal{H}}$$

Note that all of the steps above depend only on computing **inner products** in the Hilbert space  $\mathcal{H}$ . If this can be done efficiently, then we are home-free.

For our Gaussian radial-basis function example, we actually have a closed form expression for the inner products:

$$\begin{aligned}\langle \Phi(\mathbf{t}_m), \Phi(\mathbf{t}_\ell) \rangle &= \int_{\mathbf{s} \in \mathbb{R}^D} e^{-\|\mathbf{s}-\mathbf{t}_m\|_2^2/2\sigma^2} e^{-\|\mathbf{s}-\mathbf{t}_\ell\|_2^2/2\sigma^2} d\mathbf{s} \\ &= (\sigma\sqrt{\pi})^D e^{-\|\mathbf{t}_m-\mathbf{t}_\ell\|_2^2/4\sigma^2}.\end{aligned}$$

So computing the required inner products takes about the same amount of time as computing all the pairwise distances between the data points.

## Kernels

Our discussion above revolved around having a mapping  $\Phi(\cdot)$  from data space into an abstract Hilbert space, and then an efficient way of computing inner products between arbitrary  $\Phi(\mathbf{t})$  and  $\Phi(\mathbf{t}')$  in this Hilbert space. In the end, the infinite dimensional Hilbert space is just there for us to think about ... it comes into the actual computation of the solution only through these inner products. Loosely speaking, the inner products  $\langle \Phi(\mathbf{t}), \Phi(\mathbf{t}') \rangle$  can be interpreted as measuring the “similarity” or “likeness” of the data points  $\mathbf{t}$  and  $\mathbf{t}'$  — different nonlinear mappings  $\Phi$  give different notions of likeness.

So let’s skip the middleman. Suppose we have a “similarity” measure defined by a function  $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ , then simply replace the  $\langle \mathbf{t}_\ell, \mathbf{t}_m \rangle$  (in linear regression) or  $\langle \Psi(\mathbf{t}_\ell), \Psi(\mathbf{t}_m) \rangle$  (in finite-dimensional basis regression) or  $\langle \Phi(\mathbf{t}_\ell), \Phi(\mathbf{t}_m) \rangle$  (in infinite-dimensional Hilbert space regression) with  $k(\mathbf{t}_\ell, \mathbf{t}_m)$ . Then every different  $k(\cdot, \cdot)$  gives us a new regression technique; given data  $\{(\mathbf{t}_m, y_m)\}$ , we follow the steps

1. Create matrix  $\mathbf{K}$  by computing

$$K_{\ell,m} = k(\mathbf{t}_m, \mathbf{t}_\ell).$$

2. Solve the  $M \times M$  system

$$(\mathbf{K} + \delta \mathbf{I}) \hat{\boldsymbol{\alpha}} = \mathbf{y}.$$

3. The solution  $\hat{\boldsymbol{\alpha}}$  now parameterizes your regression function  $\hat{f}$ . Given an arbitrary  $\mathbf{t} \in \mathbb{R}^D$ , we can evaluate  $\hat{f}$  at  $\mathbf{t}$  with

$$\hat{f}(\mathbf{t}) = \sum_{m=1}^M \hat{\alpha}_m k(\mathbf{t}, \mathbf{t}_m).$$

That is great, but as you must suspect, we need  $k(\cdot, \cdot)$  to obey certain properties for this procedure to make sense. What is critical is that  $k(\cdot, \cdot)$  mimics the (valid) inner product of the data mapped into some abstract Hilbert space. If it does, then we know we are solving (5) for some  $\mathcal{H}$  and some mapping  $\Phi(\cdot)$  into  $\mathcal{H}$ .

We have the natural question:

*What kinds of functions  $k(\mathbf{u}, \mathbf{t})$  correspond to inner products between nonlinear mappings of  $\mathbf{u}$  and  $\mathbf{t}$  into some Hilbert space?*

There is a succinct answer to this question: when  $k(\cdot, \cdot)$  is a symmetric positive-definite kernel, or “kernel” for short<sup>5</sup>.

---

<sup>5</sup>The word “kernel” comes from linear operator theory, where functions like these are used to define linear operators that act on points in a Hilbert space. It is a cool word, so we will keep it.

**Definition:** We say that  $k(\cdot, \cdot) : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$  is a **symmetric positive semi-definite kernel** (PSD kernel) if the following hold:

1.  $k(\mathbf{u}, \mathbf{t}) = k(\mathbf{t}, \mathbf{u})$  for all  $\mathbf{u}, \mathbf{t} \in \mathbb{R}^D$ ,
2. for all  $M \in \mathbb{N}$  and all  $\mathbf{t}_1, \dots, \mathbf{t}_M \in \mathbb{R}^D$ , the matrix  $\mathbf{K}$  with entries

$$K_{\ell, m} = k(\mathbf{t}_m, \mathbf{t}_\ell)$$

is positive semidefinite, meaning  $\mathbf{z}^T \mathbf{K} \mathbf{z} \geq 0$  for all  $\mathbf{z} \in \mathbb{R}^M$ .

The following is typically referred to as *Mercer's Theorem*:

Given a kernel function  $k(\mathbf{u}, \mathbf{t})$ , there exists a Hilbert space  $\mathcal{H}$  with associated inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  and mapping  $\Phi : \mathbb{R}^D \rightarrow \mathcal{H}$  with

$$\langle \Phi(\mathbf{u}), \Phi(\mathbf{t}) \rangle = k(\mathbf{u}, \mathbf{t}),$$

if and only if  $k$  is a PSD kernel.

We will show why this is true in the Technical Details section below.

The “kernel trick” discussed above works equally well whether  $\mathcal{H}$  is finite- or infinite-dimensional. Here some examples of kernels that are used in machine learning algorithms:

- Polynomial kernels. These come in two varieties,

$$\text{homogeneous: } k(\mathbf{t}, \mathbf{t}') = \langle \mathbf{t}, \mathbf{t}' \rangle^p,$$

and

$$\text{inhomogeneous: } k(\mathbf{t}, \mathbf{t}') = (1 + \langle \mathbf{t}, \mathbf{t}' \rangle)^p.$$

In this case, we have a solid interpretation of the Hilbert space into which we are mapping. As discussed in the finite-dimensional

examples above, using these kernels corresponds to fitting an  $f(\mathbf{t})$  that is a  $p$ th order polynomial in the entries of  $\mathbf{t}$ . The homogeneous kernel uses only terms of order  $p$ , while the inhomogeneous kernel includes terms of order  $p$  or smaller. (As an example, you can expand out the kernels in the two cases above for  $p = 2$ .)

- Radial basis kernels. These are kernels that have the form

$$k(\mathbf{t}, \mathbf{t}') = r(\|\mathbf{t} - \mathbf{t}'\|_2)$$

for some function of one variable  $r : \mathbb{R} \rightarrow \mathbb{R}$ . Above, we looked carefully at the case where  $r(t) = e^{-t^2/2\sigma^2}$ . You can also take  $r$  to be a B-spline, or a sinc function, etc. The Hilbert space you are mapping into is the (closure of the) span of all different shifts of  $r(\|\mathbf{s}\|_2)$

$$\mathcal{H} = \text{Span}(\{r(\|\mathbf{s} - \mathbf{t}\|_2), \mathbf{t} \in \mathbb{R}^D\}).$$

(As before, we could also consider shifts over some region  $\mathbf{t} \in \mathcal{T}$  in place of all of  $\mathbb{R}^D$ . In this case,  $\mathcal{H}$  will be infinite dimensional.

- Sigmoid kernel. The kernel function is

$$k(\mathbf{t}, \mathbf{t}') = \tanh(a\langle \mathbf{t}, \mathbf{t}' \rangle + c),$$

for some  $a > 0$  and  $c \in \mathbb{R}$ . Again, this corresponds to a mapping into an infinite-dimensional Hilbert space (that has, as far as I know, no easy description).

## Technical Details: Proof of Mercer's Theorem

To start, notice that when we fix one of the inputs to the kernel  $k$ , it becomes a function on  $\mathbb{R}^D$ . By convention, we will fix the second argument, writing  $h_{\mathbf{t}}(\mathbf{s}) = k(\mathbf{s}, \mathbf{t})$ . As we vary  $\mathbf{t}$ , this function changes.

We will take  $\mathcal{H}$  as the space of *functions* with domain  $\mathbb{R}^D$  that can be written as a finite linear combination of “columns” of the kernel  $k(\mathbf{s}, \mathbf{t})$ . More precisely, all  $\mathbf{h} \in \mathcal{H}$  are functions  $h : \mathbb{R}^D \rightarrow \mathbb{R}$  that can be written as

$$h(\mathbf{s}) = \sum_{i=1}^N \alpha_i k(\mathbf{s}, \mathbf{t}_i) \quad (6)$$

for some natural number  $N$ , some sequence of vectors  $\mathbf{t}_1, \dots, \mathbf{t}_N \in \mathbb{R}^D$ , and some sequence of scalars  $\alpha_1, \dots, \alpha_N \in \mathbb{R}$ . A simpler way to say this is that <sup>6</sup>

$$\mathcal{H} = \text{Span} \{k(\cdot, \mathbf{t}), \mathbf{t} \in \mathbb{R}^D\}.$$

Now for two such functions  $\mathbf{h}, \mathbf{g} \in \mathcal{H}$ ,

$$h(\mathbf{s}) = \sum_{i=1}^N \alpha_i k(\mathbf{s}, \mathbf{t}_i), \quad g(\mathbf{s}) = \sum_{j=1}^M \beta_j k(\mathbf{s}, \mathbf{u}_j),$$

we consider the following candidate for the inner product:

$$\langle \mathbf{h}, \mathbf{g} \rangle_{\mathcal{H}} = \sum_{i=1}^N \sum_{j=1}^M \alpha_i \beta_j k(\mathbf{t}_i, \mathbf{u}_j). \quad (7)$$

If we take  $\Phi : \mathbb{R}^D \rightarrow \mathcal{H}$  as

$$\Phi(\mathbf{t}) = k(\cdot, \mathbf{t}),$$

---

<sup>6</sup>Again, we should really be taking the closure of this finite linear span.

then

$$\langle \Phi(\mathbf{t}'), \Phi(\mathbf{t}) \rangle_{\mathcal{H}} = k(\mathbf{t}', \mathbf{t}),$$

since for  $k(\cdot, \mathbf{t})$  we can take  $N = 1, \alpha_1 = 1$  in the representation (6), and similarly for  $k(\cdot, \mathbf{t}')$ . It still remains to show that this is indeed a valid inner product on  $\mathcal{H}$ .

Your first concern might be that the  $N, \{\mathbf{t}_i\}, \{\alpha_i\}, M, \{\mathbf{u}_i\}, \{\beta_i\}$  are not unique, and indeed there could in general be many different choices that lead to the same  $\mathbf{h}$  and  $\mathbf{g}$ . But over all such choices, the inner product above will evaluate to the same thing, as

$$\langle \mathbf{h}, \mathbf{g} \rangle = \sum_{i=1}^N \sum_{j=1}^M \alpha_i \beta_j k(\mathbf{t}_i, \mathbf{u}_j) = \sum_{j=1}^M \beta_j h(\mathbf{u}_j) = \sum_{i=1}^N \alpha_i g(\mathbf{t}_i). \quad (8)$$

Now we verify that (7) meets the three criteria for an inner product when  $k(\mathbf{s}, \mathbf{t})$  is symmetric positive semi-definite kernel.

Symmetry follows immediately from the symmetry of  $k(\mathbf{s}, \mathbf{t})$ . The linearity property is also straightforward to verify given the form of (7). For the third property, that  $\langle \mathbf{h}, \mathbf{h} \rangle = 0 \Leftrightarrow \mathbf{h} = \mathbf{0}$ , consider that for a fixed  $\mathbf{h} \in \mathcal{H}$ ,

$$\langle \mathbf{h}, \mathbf{h} \rangle = \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j k(\mathbf{t}_i, \mathbf{t}_j) = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha},$$

where  $\boldsymbol{\alpha} \in \mathbb{R}^N$  contains the  $\alpha_i$ , and  $\mathbf{K}$  is the  $N \times N$  matrix with entries  $K_{i,j} = k(\mathbf{t}_j, \mathbf{t}_i)$ . Since  $k(\cdot, \cdot)$  is a symmetric positive semidefinite kernel, we know that  $\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \geq 0$ . It is also clear that  $\mathbf{h} = \mathbf{0} \Rightarrow \langle \mathbf{h}, \mathbf{h} \rangle = 0$ .

It remains to show that  $\langle \mathbf{h}, \mathbf{h} \rangle = 0 \Rightarrow \mathbf{h} = \mathbf{0}$ . Suppose that indeed

$\langle \mathbf{h}, \mathbf{h} \rangle = 0$ , then consider  $\mathbf{h}$  evaluated at an arbitrary  $\mathbf{s} \in \mathbb{R}^D$ :

$$\begin{aligned} |h(\mathbf{s})| &= \left| \sum_{i=1}^N \alpha_i k(\mathbf{s}, \mathbf{t}_i) \right| \\ &= |\langle \mathbf{h}, k(\cdot, \mathbf{s}) \rangle|, \end{aligned}$$

where the second equality follows from the fact that  $k(\cdot, \mathbf{s}) \in \mathcal{H}$  and (8). Using properties of  $\langle \cdot, \cdot \rangle$  we have established so far and the variation of the Cauchy-Schwarz inequality in Lemma 1 below, we have

$$|h(\mathbf{s})|^2 \leq \langle \mathbf{h}, \mathbf{h} \rangle \langle k(\cdot, \mathbf{s}), k(\cdot, \mathbf{s}) \rangle = 0.$$

Thus  $h(\mathbf{s}) = 0$  for all  $\mathbf{s} \in \mathbb{R}^D$ , and so  $\mathbf{h} = \mathbf{0}$ .

Our last ingredient is to say precisely what is meant by “variation of the Cauchy-Schwarz inequality”. Basically, we cannot just apply Cauchy-Schwarz, as this assumes a valid inner product, and that is exactly what we are trying to establish. The following lemma is a more general version which shows that something like CS holds for bilinear forms with the properties we have established so far.

**Lemma 1** *Suppose that  $Q(\cdot, \cdot)$  is a symmetric bilinear function on a real-valued vector space  $\mathcal{H}$  obeying*

1.  $Q(\mathbf{h}, \mathbf{g}) = Q(\mathbf{g}, \mathbf{h})$  for all  $\mathbf{h}, \mathbf{g} \in \mathcal{H}$ ,
2.  $Q(\alpha_1 \mathbf{h} + \alpha_2 \mathbf{h}, \mathbf{g}) = \alpha_1 Q(\mathbf{h}, \mathbf{g}) + \alpha_2 Q(\mathbf{h}, \mathbf{g})$ , and
3.  $Q(\mathbf{h}, \mathbf{h}) \geq 0$ ,

for all  $\mathbf{h}, \mathbf{g}, \mathbf{h} \in \mathcal{H}$  and  $\alpha_1, \alpha_2 \in \mathbb{R}$ . Then

$$|Q(\mathbf{h}, \mathbf{g})|^2 \leq Q(\mathbf{h}, \mathbf{h}) Q(\mathbf{g}, \mathbf{g}).$$



**Proof** For any  $\mathbf{h}, \mathbf{g} \in \mathcal{H}$ , the properties of  $Q(\cdot, \cdot)$  tell us that  $Q(\mathbf{h} + \mathbf{g}, \mathbf{h} + \mathbf{g}) \geq 0$ , and

$$Q(\mathbf{h} + \mathbf{g}, \mathbf{h} + \mathbf{g}) = Q(\mathbf{h}, \mathbf{h}) + 2Q(\mathbf{h}, \mathbf{g}) + Q(\mathbf{g}, \mathbf{g}).$$

Since the expression above holds for *all*  $\mathbf{h}, \mathbf{g} \in \mathcal{H}$ , and  $\mathcal{H}$  is a linear vector space, it must hold for  $\alpha\mathbf{h}$  and  $\beta\mathbf{g}$  for all  $\alpha, \beta \in \mathbb{R}$ , meaning

$$\alpha^2 Q(\mathbf{h}, \mathbf{h}) + 2\alpha\beta Q(\mathbf{h}, \mathbf{g}) + \beta^2 Q(\mathbf{g}, \mathbf{g}) \geq 0, \quad \text{for all } \alpha, \beta \in \mathbb{R}.$$

This is the same as saying that

$$\begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} Q(\mathbf{h}, \mathbf{h}) & Q(\mathbf{h}, \mathbf{g}) \\ Q(\mathbf{h}, \mathbf{g}) & Q(\mathbf{g}, \mathbf{g}) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \geq 0,$$

which means that the  $2 \times 2$  matrix above is symmetric positive semi-definite for all  $\mathbf{h}, \mathbf{g}$ . This means that its determinant is non-negative, and so

$$Q(\mathbf{h}, \mathbf{h})Q(\mathbf{g}, \mathbf{g}) - |Q(\mathbf{h}, \mathbf{g})|^2 \geq 0.$$

■