Bases and Kernels: Two Modeling Paradigms

Let's review three things we have seen in this class: linear regression, nonlinear regression using a basis, and kernel regression. We will use a slightly different point of view (and slightly different notation) than when we first encountered them so that we can suss out the relationships between them.

Again, this is all in the context of our fundamental data-fitting problem: we are given pairs of data points $(\boldsymbol{t}_m, y_m)$, $m = 1, \ldots, M$ with $\boldsymbol{t}_m \in \mathbb{R}^D$ and $y_m \in \mathbb{R}$, and we want to find a mapping $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}$ such that

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

We approach this data-fitting problem using least-squares; we search in the Hilbert space $\mathcal{S}$ (with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$), solving

$$\underset{\boldsymbol{f} \in \mathcal{S}}{\text{minimize}} \ \sum_{m=1}^{M} |y_m - f(\boldsymbol{t}_m)|^2 + \delta \|\boldsymbol{f}\|^2. \tag{1}$$

We have seen two different techniques to discretize this problem, and turn the search over the ∞-dimensional space $\mathcal{S}$ into a finite-dimensional least-squares problem; each of these techniques using a different model.

Regression using a finite dimensional basis

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

$$\boldsymbol{f} \in \text{span}\left(\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N\}\right),$$

i.e.

$$f(\boldsymbol{t}) = \sum_{n=1}^{N} w_n \psi_n(\boldsymbol{t}), \quad \forall \boldsymbol{t} \in \mathbb{R}^D. \tag{2}$$

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

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

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

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

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

To actually fit such an $\boldsymbol{f}$ given the data $\{(\boldsymbol{t}_m, y_m)\}_{m=1}^M$, we can use *ridge regression*. By plugging (2) into (1), we are now solving

$$\underset{\boldsymbol{w} \in \mathbb{R}^N}{\text{minimize}} \ \sum_{m=1}^{M} \left(y_m - \sum_{n=1}^{N} w_n \psi_n(\boldsymbol{t}_m) \right)^2 + \delta \left\| \sum_{n=1}^{N} w_n \boldsymbol{\psi}_n \right\|^2, \tag{3}$$

which is the same as

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

where

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

and $\boldsymbol{G}$ is the Gram matrix for the basis, $G_{i,j} = \langle \boldsymbol{\psi}_j, \boldsymbol{\psi}_i \rangle$. A quick calculation (taking the gradient and setting it equal to zero) shows that the solution to this problem is given by

$$\hat{\boldsymbol{w}} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \delta \boldsymbol{G})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$$

In general, since we have $\boldsymbol{G}$ in place of $\mathbf{I}$, $\hat{\boldsymbol{w}}$ will not be in the row space of $\boldsymbol{A}$. However, when the $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N\}$ are orthonormal, it is: we have $\boldsymbol{G} = \mathbf{I}$ and the procedure mirrors exactly what we called ridge regression previously in the lecture notes, and we can take

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

It is also true that since there are a finite number of the $\boldsymbol{\psi}_n$ and they are linearly independent, then we can always re-define the inner product on $\mathcal{S}$ so that they are orthonormal — the only place the inner product is coming into play in the solution is in the computation of the $\boldsymbol{G}$ matrix. This re-definition changes the model we are using for $\boldsymbol{f}$, and so of course we will get a slightly different answer. And in fact, this is what is typically done, we just assume/decree that we are working in a Hilbert space where the $\boldsymbol{\psi}_n$ are orthonormal; it is an straightforward exercise to show that such a space always exists. We will proceed, then, with $\boldsymbol{G} = \mathbf{I}$.

To compute the $\hat{w}$, we need to form the matrix $\boldsymbol{K} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$, and so

$$K_{i,j} = \langle \boldsymbol{\Psi}(\boldsymbol{t}_j), \boldsymbol{\Psi}(\boldsymbol{t}_i) \rangle. \tag{4}$$

This results in an $M \times M$ (positive definite) system of equations, which we solve and then take

$$\hat{\boldsymbol{w}} = \boldsymbol{A}^{\mathrm{T}}\hat{\boldsymbol{\alpha}} = \sum_{m=1}^{M} \hat{\alpha}_m \boldsymbol{\Psi}(\boldsymbol{t}_m). \tag{5}$$

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

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

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

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

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

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

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

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 uses a basis that can build up all polynomial functions on $\mathbb{R}^D$ of at most p, the mapping becomes

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

with entries for all values of $p' \leq p$ and $(j_1, \ldots, j_D)$ 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

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

But again, we are saved from this enormous N by a trick for computing the necessary inner products. Using the multinomial theorem[1], we can show that

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

[1]The Wikipedia article on this is decent, https://en.wikipedia.org/wiki/Multinomial_theorem.

So again, all the computations for creating the $M \times M$ matrix $\boldsymbol{K}$ can take place in $\mathbb{R}^D$. The trick above makes the cost of solving (3) 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 $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_M \in \mathbb{R}^D$ 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 $\phi : \mathbb{R}^D \to \mathbb{R}$ be the so-called "Gaussian radial basis function"[2]

$$\phi(\boldsymbol{s}) = \left(\frac{1}{\sigma\sqrt{\pi}}\right)^{D/2} \exp\left(-\frac{\|\boldsymbol{s}\|_2^2}{2\sigma^2}\right),$$

where σ is a width parameter that we get to choose. Now, let $\mathcal{T}$ be some region in $\mathbb{R}^D$ where we can safely restrict $\boldsymbol{t}_m \in \mathcal{T}$ — $\mathcal{T}$ 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 $\mathcal{S}$ be the **finite linear span** of all shifts $\phi(\boldsymbol{s} - \boldsymbol{t})$ for $\boldsymbol{t} \in \mathcal{T}$. That is, $\mathcal{S}$ contains all functions $h : \mathbb{R}^D \to \mathbb{R}$ such that

$$h(\boldsymbol{s}) = \sum_{k=1}^{K} \alpha_k \, \phi(\boldsymbol{s} - \boldsymbol{t}_k), \tag{6}$$

[2]The constant $(\sigma\sqrt{pi})^{-D/2}$ is there simply a normalization factor, so that $\|\boldsymbol{\phi}\|_{L_2} = 1$.

for any arbitrary K and $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K \in \mathcal{T}$. It is a fact that for any distinct set of shifts $\{\boldsymbol{t}_k\}$, the $\{\phi(\boldsymbol{s} - \boldsymbol{t}_k)\}$ will be linearly independent. As K is arbitrary, this means that this space is **infinite dimensional** (even if $\mathcal{T}$ is bounded).

$\mathcal{S}$ is a subspace of $L_2(\mathbb{R}^D)$, and it naturally inherits the standard inner product. Given two functions h as in(6) and g written as

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

we can write this inner product[3] as

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

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

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

At this point we should mention that for $\mathcal{S}$ to be a Hilbert space, we really need to take the *closure* of the finite linear span, which essentially means that all members of $\mathcal{S}$ can be approximated arbitrarily well by functions of the form (6). This is an important detail, but does not really affect how we think about this space qualitatively.

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

$$\boldsymbol{\Phi} : \mathbb{R}^D \to \mathcal{S}, \quad \boldsymbol{\Phi}(\boldsymbol{t}) = \phi(\boldsymbol{s} - \boldsymbol{t}).$$

[3] Again, the $\langle \cdot, \cdot \rangle_{\mathcal{S}}$ 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 $\{(\boldsymbol{t}_m, y_m)\}$, our goal is to write the y_m as a linear functional evaluated at $\boldsymbol{\Phi}(\boldsymbol{t}_m)$; that is[4], we want to find a $\boldsymbol{w} \in \mathcal{S}$ such that

$$\langle \boldsymbol{\Phi}(\boldsymbol{t}_m), \boldsymbol{w} \rangle_{\mathcal{S}} \approx y_m, \quad m = 1, \ldots, M.$$

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

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

$$\underset{\boldsymbol{w} \in \mathcal{S}}{\text{minimize}} \quad \sum_{m=1}^{M} \left(y_m - \langle \boldsymbol{\Phi}(\boldsymbol{t}_m), \boldsymbol{w} \rangle_{\mathcal{S}} \right)^2 + \delta \|\boldsymbol{w}\|_{\mathcal{S}}^2. \tag{7}$$

We again know how to solve this problem, thanks to the Representor Theorem; we know that there exists $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^M$ such that

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

is the solution to (7), and moreover we know that to compute these $\hat{\boldsymbol{\alpha}}$, we form the $M \times M$ matrix $\boldsymbol{K}$ with entries

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

and then compute

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

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

[4] As we have seen, the Riesz Representation Theorem says that every (continuous) linear functional on a Hilbert space can be written as an inner product with a fixed vector from that space.

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

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

$$\langle \boldsymbol{\Phi}(t_m), \boldsymbol{\Phi}(t_\ell) \rangle_{\mathcal{S}} \quad \text{for all} \;\; m, \ell = 1, \ldots, M.$$

2. Solve the symmetric positive-definite system of equations

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

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

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

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

Note that all of the steps above depend only on computing **inner products** in the Hilbert space $\mathcal{S}$. 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:

$$\begin{aligned} \langle \boldsymbol{\Phi}(\boldsymbol{t}_m), \boldsymbol{\Phi}(\boldsymbol{t}_\ell) \rangle &= \left(\frac{1}{\sigma \sqrt{\pi}} \right)^D \int_{\boldsymbol{s} \in \mathbb{R}^D} e^{-\|\boldsymbol{s} - \boldsymbol{t}_m\|_2^2 / 2\sigma^2} e^{-\|\boldsymbol{s} - \boldsymbol{t}_\ell\|_2^2 / 2\sigma^2} \; \mathrm{d}\boldsymbol{s} \\ &= e^{-\|\boldsymbol{t}_m - \boldsymbol{t}_\ell\|_2^2 / 4\sigma^2}. \end{aligned}$$

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 $\boldsymbol{\Phi}(\cdot)$ from data space into an abstract Hilbert space, and then an efficient way of computing inner products between arbitrary $\boldsymbol{\Phi}(\boldsymbol{t})$ and $\boldsymbol{\Phi}(\boldsymbol{t}')$ 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 $\langle \boldsymbol{\Phi}(\boldsymbol{t}), \boldsymbol{\Phi}(\boldsymbol{t}') \rangle$ can be interpreted as measuring the "similarity" or "likeness" of the data points $\boldsymbol{t}$ and $\boldsymbol{t}'$ — different nonlinear mappings $\boldsymbol{\Phi}$ give different notions of likeness.

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

1. Create matrix $\boldsymbol{K}$ by computing
$$K_{\ell,m} = k(\boldsymbol{t}_m, \boldsymbol{t}_\ell).$$
2. Solve the $M \times M$ system
$$(\boldsymbol{K} + \delta \mathbf{I})\hat{\boldsymbol{\alpha}} = \boldsymbol{y}.$$
3. The solution $\hat{\boldsymbol{\alpha}}$ now parameterizes your regression function $\hat{f}$. Given an arbitrary $\boldsymbol{t} \in \mathbb{R}^D$, we can evaluate $\hat{f}$ at $\boldsymbol{t}$ with
$$\hat{f}(\boldsymbol{t}) = \sum_{m=1}^{M} \hat{\alpha}_m \, k(\boldsymbol{t}, \boldsymbol{t}_m).$$

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 (7) for some $\mathcal{S}$ and some mapping $\boldsymbol{\Phi}(\cdot)$ into $\mathcal{S}$.

We have the natural question:

What kinds of functions $k(\boldsymbol{u},\boldsymbol{t})$ correspond to inner products between nonlinear mappings of $\boldsymbol{u}$ and $\boldsymbol{t}$ 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[5].

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

1. $k(\boldsymbol{u},\boldsymbol{t}) = k(\boldsymbol{t},\boldsymbol{u})$ for all $\boldsymbol{u},\boldsymbol{t} \in \mathbb{R}^D$,
2. for all $M \in \mathbb{N}$ and all $\boldsymbol{t}_1,\ldots,\boldsymbol{t}_M \in \mathbb{R}^D$, the matrix $\boldsymbol{K}$ with entries
$$K_{\ell,m} = k(\boldsymbol{t}_m,\boldsymbol{t}_\ell)$$
is positive semidefinite, meaning $\boldsymbol{z}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{z} \geq 0$ for all $\boldsymbol{z} \in \mathbb{R}^M$.

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

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

[5]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.

with

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

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 $\mathcal{S}$ is finite- or infinite-dimensional. Here some examples of kernels that are used in machine learning algorithms:

- Polynomial kernels. These come in two varieties,

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

  and

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

  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 $f(\boldsymbol{t})$ that is a pth order polynomial in the entries of $\boldsymbol{t}$. 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

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

  for some function of one variable $r : \mathbb{R} \to \mathbb{R}$. Above, we looked carefully at the case where $r(t) = e^{-t^2/4\sigma^2}$. You can also take r to be a B-spline, or a sinc function, etc — the

only condition is that the D-dimensional Fourier transform of $r(\|\boldsymbol{t}\|_2)$ is real-valued and positive (a sufficient condition for this is if the one dimensional Fourier transform of $r(t)$ is real-valued and positive). This means that there is a function $\tilde{r}$ such that

$$r(\|\boldsymbol{t}-\boldsymbol{t}'\|_2) = \int_{\boldsymbol{s}\in\mathbb{R}^D} \tilde{r}(\|\boldsymbol{s}-\boldsymbol{t}\|_2)\,\tilde{r}(\|\boldsymbol{s}-\boldsymbol{t}'\|_2)\ \mathrm{d}\boldsymbol{s}.$$

The Hilbert space you are mapping into is the (closure of the) span of all different shifts of $\tilde{r}(\|\boldsymbol{s}\|_2)$

$$\mathcal{S} = \mathrm{span}\left(\left\{\tilde{r}(\|\boldsymbol{s}-\boldsymbol{t}\|_2),\ \boldsymbol{t}\in\mathbb{R}^D\right\}\right).$$

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

- Sigmoid kernel. The kernel function is

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

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 $h_{\boldsymbol{t}}(\boldsymbol{s}) = k(\boldsymbol{s}, \boldsymbol{t})$. As we vary $\boldsymbol{t}$, this function changes.

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

$$h(\boldsymbol{s}) = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{s}, \boldsymbol{t}_i) \tag{8}$$

for some natural number N, some sequence of vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_N \in \mathbb{R}^D$, and some sequence of scalars $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$. A simpler way to say this is that [6]

$$\mathcal{S} = \text{span}\left\{k(\cdot, \boldsymbol{t}),\ \boldsymbol{t} \in \mathbb{R}^D\right\}.$$

Now for two such functions $\boldsymbol{h}, \boldsymbol{g} \in \mathcal{S}$,

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

we consider the following candidate for the inner product:

$$\langle \boldsymbol{h}, \boldsymbol{g} \rangle_{\mathcal{S}} = \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j\, k(\boldsymbol{t}_i, \boldsymbol{u}_j). \tag{9}$$

If we take $\boldsymbol{\Phi} : \mathbb{R}^D \to \mathcal{S}$ as

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

[6] Again, we should really be taking the closure of this finite linear span.

then

$$\langle \boldsymbol{\Phi}(\boldsymbol{t}'), \boldsymbol{\Phi}(\boldsymbol{t}) \rangle_{\mathcal{S}} = k(\boldsymbol{t}', \boldsymbol{t}),$$

since for $k(\cdot, \boldsymbol{t})$ we can take $N = 1, \alpha_1 = 1$ in the representation (8), and similarly for $k(\cdot, \boldsymbol{t}')$. It still remains to show that this is indeed a valid inner product on $\mathcal{S}$.

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

$$\langle \boldsymbol{h}, \boldsymbol{g} \rangle = \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j \, k(\boldsymbol{t}_i, \boldsymbol{u}_j) = \sum_{j=1}^{M} \beta_j h(\boldsymbol{u}_j) = \sum_{i=1}^{N} \alpha_i g(\boldsymbol{t}_i). \quad (10)$$

Now we verify that (9) meets the three criteria for an inner product when $k(\boldsymbol{s}, \boldsymbol{t})$ is symmetric positive semi-definite kernel.

Symmetry follows immediately from the symmetry of $k(\boldsymbol{s}, \boldsymbol{t})$. The linearity property is also straightforward to verify given the form of (9). For the third property, that $\langle \boldsymbol{h}, \boldsymbol{h} \rangle = 0 \Leftrightarrow \boldsymbol{h} = \boldsymbol{0}$, consider that for a fixed $\boldsymbol{h} \in \mathcal{S}$,

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

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

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

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

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

where the second equality follows from the fact that $k(\cdot, \boldsymbol{s}) \in \mathcal{S}$ and (10). 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

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

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

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 $\mathcal{S}$ obeying*

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

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

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

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

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

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

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

This is the same as saying that

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

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

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

■