Introduction to RKHS, and some simple kernel algorithms

Arthur Gretton

October 25, 2017

1 Outline

In this document, we give a nontechical introduction to reproducing kernel Hilbert spaces (RKHSs), and describe some basic algorithms in RHKS.

1. What is a kernel, how do we construct it?
2. Operations on kernels that allow us to combine them
3. The reproducing kernel Hilbert space
4. Application 1: Difference in means in feature space
5. Application 2: Kernel PCA
6. Application 3: Ridge regression

2 Motivating examples

For the XOR example, we have variables in two dimensions, $x \in \mathbb{R}^2$, arranged in an XOR pattern. We would like to separate the red patterns from the blue, using only a linear classifier. This is clearly not possible, in the original space. If we map points to a higher dimensional feature space

$$\phi(x) = \begin{bmatrix} x_1 & x_2 & x_1 x_2 \end{bmatrix} \in \mathbb{R}^3,$$

it is possible to use a linear classifier to separate the points. See Figure 2.1.

Feature spaces can be used to compare objects which have much more complex structure. An illustration is in Figure 2.2, where we have two sets of documents (the red ones on dogs, and the blue on cats) which we wish to classify. In this case, features of the documents are chosen to be histograms over words (there are much more sophisticated features we could use, eg string kernels [4]). To use the terminology from the first example, these histograms represent a mapping of the documents to feature space. Once we have histograms, we can compare documents, classify them, cluster them, etc.

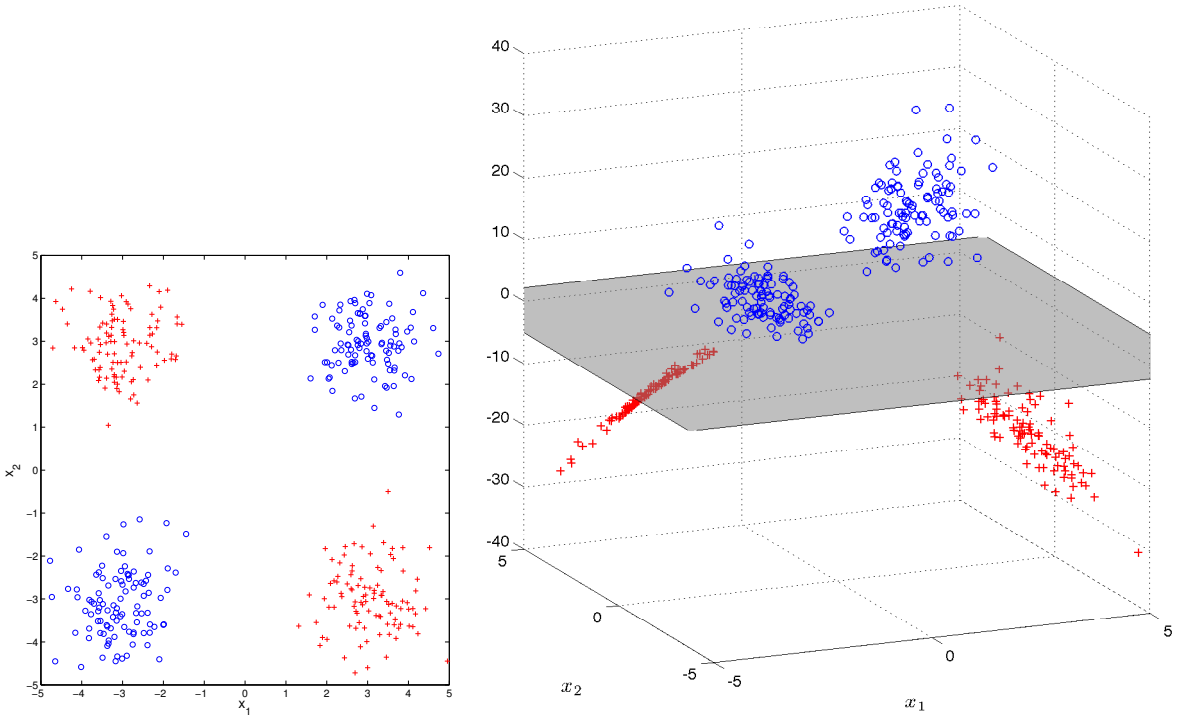

Figure 2.1: XOR example. On the left, the points are plotted in the original space. There is no linear classifier that can separate the red crosses from the blue circles. Mapping the points to a higher dimensional feature space, we obtain linearly separable classes. A possible decision boundary is shown as a gray plane.

The classsification of objects via well chosen features is of course not an unusual approach. What distinguishes kernel methods is that they can (and often do) use *infinitely many features*. This can be achieved as long as our learning algorithms are defined in terms of *dot products* between the features, where these dot products can be computed in closed form. The term "kernel" simply refers to a dot product between (possibly infinitely many) features.

Alternatively, kernel methods can be used to control smoothness of a function used in regression or classification. An example is given in Figure 2.3, where different parameter choices determine whether the regression function overfits, underfits, or fits optimally. The connection between feature spaces and smoothness is not obvious, and is one of the things we'll discuss in the course.

3 What is a kernel and how do we construct it?

3.1 Construction of kernels

The following is taken mainly from [11, Section 4.1].

Definition 1 (Inner product)**.** Let $\mathcal{H}$ be a vector space over $\mathbb{R}$. A function $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is said to be *an inner product* on $\mathcal{H}$ if

1. $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle_{\mathcal{H}} = \alpha_1 \langle f_1, g \rangle_{\mathcal{H}} + \alpha_2 \langle f_2, g \rangle_{\mathcal{H}}$
2. $\langle f, g \rangle_{\mathcal{H}} = \langle g, f \rangle_{\mathcal{H}}$ [1]
3. $\langle f, f \rangle_{\mathcal{H}} \geq 0$ and $\langle f, f \rangle_{\mathcal{H}} = 0$ if and only if $f = 0$.

We can define a norm using the inner product as $\|f\|_{\mathcal{H}} := \sqrt{\langle f, f \rangle_{\mathcal{H}}}$.

A Hilbert space is a space on which an inner product is defined, along with an additional technical condition.[2] We now define a kernel.

Definition 3. Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a **kernel** if there exists an $\mathbb{R}$-Hilbert space and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$,

$$k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} .$$

Note that we imposed almost no conditions on $\mathcal{X}$: we don't even require there to be an inner product defined on the elements of $\mathcal{X}$. The case of documents is an instructive example: you can't take an inner product between two books, but you can take an inner product between features of the text.

[1] If the inner product is complex valued, we have conjugate symmetry, $\langle f, g \rangle_{\mathcal{H}} = \overline{\langle g, f \rangle}_{\mathcal{H}}$.

[2] Specifically, a Hilbert space must contain the limits of all Cauchy sequences of functions:

Definition 2 (Cauchy sequence)**.** A sequence $\{f_n\}_{n=1}^{\infty}$ of elements in a normed space $\mathcal{H}$ is said to be *a Cauchy sequence* if for every $\epsilon > 0$, there exists $N = N(\varepsilon) \in \mathbb{N}$, such that for all $n, m \geq N$, $\|f_n - f_m\|_{\mathcal{H}} < \epsilon$.

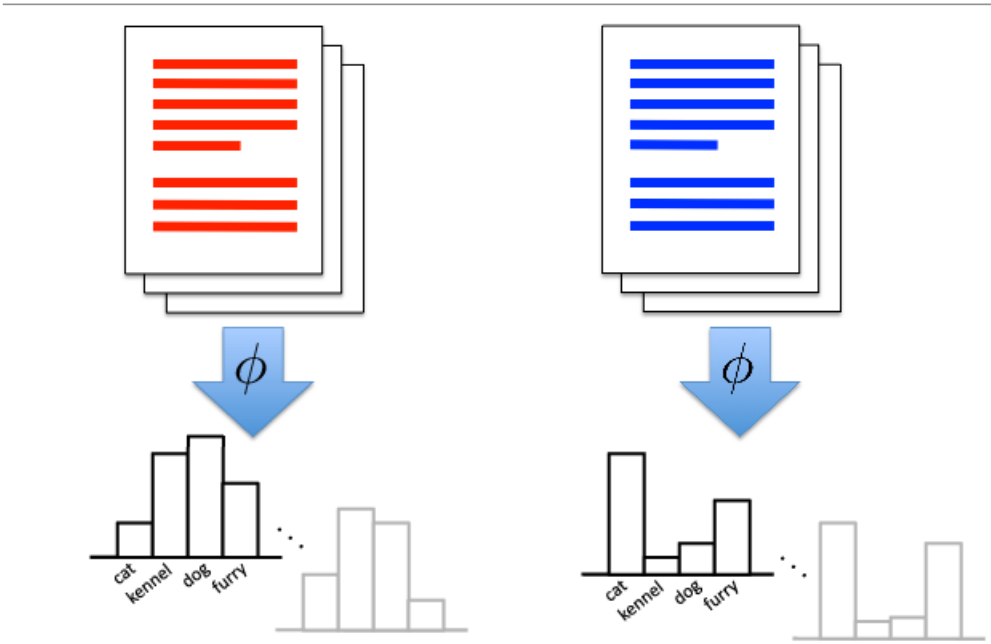

Figure 2.2: Document classification example: each document is represented as a histogram over words.

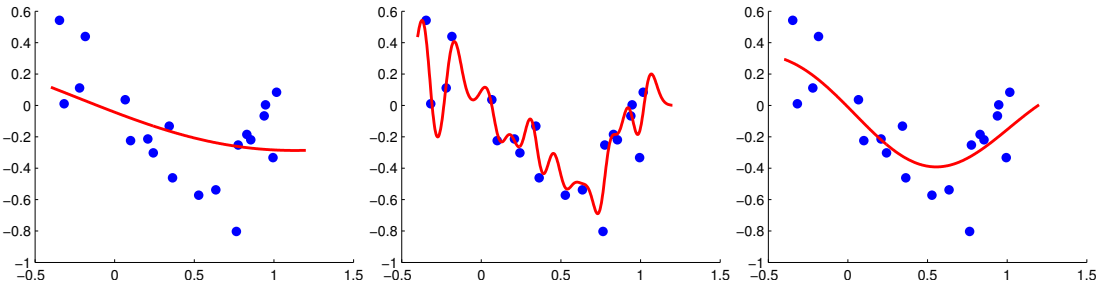

Figure 2.3: Regression: examples are shown of underfitting, overfitting, and a good fit. Kernel methods allow us to control the smoothness of the regression function.

A single kernel can correspond to multiple sets of underlying features. A trivial example for $\mathcal{X} := \mathbb{R}$:

$$\phi_1(x) = x \qquad \text{and} \qquad \phi_2(x) = \begin{bmatrix} x/\sqrt{2} \\ x/\sqrt{2} \end{bmatrix}$$

Kernels can be combined and modified to get new kernels:

Lemma 4 (Sums of kernels are kernels)**.** *Given $\alpha > 0$ and k, k_1 and k_2 all kernels on $\mathcal{X}$, then αk and $k_1 + k_2$ are kernels on $\mathcal{X}$.*

To easily prove the above, we will need to use a property of kernels introduced later, namely *positive definiteness*. We provide this proof at the end of Section 3.2. A difference of kernels may not be a kernel: if $k_1(x, x) - k_2(x, x) < 0$, then condition 3 of Definition 1 is violated.

Lemma 5 (Mappings between spaces)**.** *Let $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ be sets, and define a map $A : \mathcal{X} \to \widetilde{\mathcal{X}}$. Define the kernel k on $\widetilde{\mathcal{X}}$. Then the kernel $k(A(x), A(x'))$ is a kernel on $\mathcal{X}$.*

Lemma 6 (Products of kernels are kernels)**.** *Given k_1 on $\mathcal{X}_1$ and k_2 on $\mathcal{X}_2$, then $k_1 \times k_2$ is a kernel on $\mathcal{X}_1 \times \mathcal{X}_2$. If $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}$, then $k := k_1 \times k_2$ is a kernel on $\mathcal{X}$.*

Proof. The general proof has some technicalities: see [11, Lemma 4.6 p. 114]. However, the main idea can be shown with some simple linear algebra. We consider the case where $\mathcal{H}_1$ corresponding to k_1 is $\mathbb{R}^m$, and $\mathcal{H}_2$ corresponding to k_2 is $\mathbb{R}^n$. Define $k_1 := u^\top v$ for vectors $u, v \in \mathbb{R}^m$ and $k_2 := p^\top q$ for vectors $p, q \in \mathbb{R}^n$.

We will use that the inner product between matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times n}$ is

$$\langle A, B \rangle = \operatorname{trace}(A^\top B). \tag{3.1}$$

Then

$$\begin{aligned} k_1 k_2 &= k_1\left(p^\top q\right) \\ & \quad k_1\left(q^\top p\right) \\ &= k_1 \mathrm{trace}(q^\top p) \\ &= k_1 \mathrm{trace}(pq^\top) \\ &= \mathrm{trace}(p\underbrace{u^\top v}_{k_1} q^\top) \\ &= \langle A, B\rangle, \end{aligned}$$

where we defined $A := up^\top$ and $B := vq^\top$. In other words, the product $k_1 k_2$ defines a valid inner product in accordance with (3.1). □

The sum and product rules allow us to define a huge variety of kernels.

Lemma 7 (Polynomial kernels)**.** *Let $x, x' \in \mathbb{R}^d$ for $d \geq 1$, and let $m \geq 1$ be an integer and $c \geq 0$ be a positive real. Then*

$$k(x, x') := \left(\langle x, x'\rangle + c\right)^m$$

is a valid kernel.

To prove: expand out this expression into a sum (with non-negative scalars) of kernels $\langle x, x'\rangle$ raised to integer powers. These individual terms are valid kernels by the product rule.

Can we extend this combination of sum and product rule to sums with infinitely many terms? It turns out we can, as long as these don't blow up.

Definition 8. The space ℓ_p of p-summable sequences is defined as all sequences $(a_i)_{i\geq 1}$ for which

$$\sum_{i=1}^{\infty} a_i^p < \infty.$$

Kernels can be defined in terms of sequences in ℓ_2.

Lemma 9. *Given a non-empty set $\mathcal{X}$, and a sequence of functions $(\phi_i(x))_{i\geq 1}$ in ℓ_2 where $\phi_i : \mathcal{X} \to \mathbb{R}$ is the ith coordinate of the feature map $\phi(x)$. Then*

$$k(x, x') := \sum_{i=1}^{\infty} \phi_i(x)\phi_i(x') \tag{3.2}$$

is a well-defined kernel on $\mathcal{X}$.

Proof. We write the norm $\|a\|_{\ell_2}$ associated with the inner product (3.2) as

$$\|a\|_{\ell_2} := \sqrt{\sum_{i=1}^{\infty} a_i^2},$$

where we write a to represent the sequence with terms a_i. The Cauchy-Schwarz inequality states

$$|k(x,x')| = \left|\sum_{i=1}^{\infty} \phi_i(x)\phi_i(x')\right| \leq \|\phi(x)\|_{\ell_2} \|\phi(x')\|_{\ell_2} < \infty,$$

so the kernel in (3.2) is well defined for all $x, x' \in \mathcal{X}$. □

Taylor series expansions may be used to define kernels that have infinitely many features.

Definition 10. [Taylor series kernel] [11, Lemma 4.8] Assume we can define the Taylor series

$$f(z) = \sum_{n=0}^{\infty} a_n z^n \qquad |z| < r,\ z \in \mathbb{R},$$

for $r \in (0, \infty]$, with $a_n \geq 0$ for all $n \geq 0$. Define $\mathcal{X}$ to be the $\sqrt{r}$-ball in $\mathbb{R}^d$. Then for $x, x' \in \mathbb{R}^d$ such that $\|x\| < \sqrt{r}$, we have the kernel

$$k(x,x') = f\left(\langle x, x'\rangle\right) = \sum_{n=0}^{\infty} a_n \langle x, x'\rangle^n.$$

Proof. Non-negative weighted sums of kernels are kernels, and products of kernels are kernels, so the following is a kernel **if it converges,**

$$k(x,x') \quad = \quad \sum_{n=0}^{\infty} a_n \left(\langle x, x'\rangle\right)^n.$$

We have by Cauchy-Schwarz that

$$|\langle x, x'\rangle| \leq \|x\|\|x'\| < r,$$

so the Taylor series converges. □

An example of a Taylor series kernel is the exponential.

Example 11 (Exponential kernel)**.** The exponential kernel on $\mathbb{R}^d$ is defined as

$$k(x,x') := \exp\left(\langle x, x'\rangle\right).$$

We may combine all the results above to obtain the following (the proof is an exercise - you will need the product rule, the mapping rule, and the result of Example 11).

Example 12 (Gaussian kernel)**.** The Gaussian kernel on $\mathbb{R}^d$ is defined as

$$k(x,x') := \exp\left(-\gamma^{-2}\|x - x'\|^2\right).$$

3.2 Positive definiteness of an inner product in a Hilbert space

All kernel functions are **positive definite**. In fact, if we have a positive definite function, we know there exists one (or more) feature spaces for which the kernel defines the inner product - we are not obliged to define the feature spaces explicitly. We begin by defining positive definiteness [1, Definition 2], [11, Definition 4.15].

Definition 13 (Positive definite functions)**.** A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if $\forall n \geq 1$, $\forall (a_1, \dots a_n) \in \mathbb{R}^n$, $\forall (x_1, \dots, x_n) \in \mathcal{X}^n$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) \geq 0.$$

The function $k(\cdot, \cdot)$ is *strictly* positive definite if for mutually distinct x_i, the equality holds only when all the a_i are zero.[3]

Every inner product is a positive definite function, and more generally:

Lemma 14. *Let $\mathcal{H}$ be any Hilbert space (not necessarily an RKHS), $\mathcal{X}$ a non-empty set and $\phi : \mathcal{X} \to \mathcal{H}$. Then $k(x, y) := \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ is a positive definite function.*

Proof.

$$\begin{aligned} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \langle a_i \phi(x_i), a_j \phi(x_j) \rangle_{\mathcal{H}} \\ &= \left\langle \sum_{i=1}^{n} a_i \phi(x_i), \sum_{j=1}^{n} a_j \phi(x_j) \right\rangle_{\mathcal{H}} . \\ & \left\| \sum_{i=1}^{n} a_i \phi(x_i) \right\|_{\mathcal{H}}^2 \geq 0 \end{aligned}$$

□

Remarkably, the reverse direction also holds: a positive definite function is guaranteed to be the inner product in a Hilbert space $\mathcal{H}$ between features $\phi(x)$ (which we need not even specify explicitly). The proof is not difficult, but has some technical aspects: see [11, Theorem 4.16 p. 118].

Positive definiteness is the easiest way to prove a sum of kernels is a kernel. Consider two kernels $k_1(x, x')$ and $k_2(x, x')$. Then

[3] Wendland [12, Definition 6.1 p. 65] uses the terminology "positive semi-definite" vs "positive definite".

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \left[k_1(x_i, x_j) + k_2(x_i, x_j)\right]$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j k_1(x_i, x_j) + \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j k_2(x_i, x_j)$$
$$\geq 0$$

4 The reproducing kernel Hilbert space

We have introduced the notation of feature spaces, and kernels on these feature spaces. What's more, we've determined that these kernels are positive definite. In this section, we use these kernels to define *functions* on $\mathcal{X}$. The space of such functions is known as a reproducing kernel Hilbert space.

4.1 Motivating examples

4.1.1 Finite dimensional setting

We start with a simple example using the same finite dimensional feature space we used in the XOR motivating example (Figure 2.1). Consider the feature map

$$\begin{array}{rcl} \phi : \mathbb{R}^2 & \to & \mathbb{R}^3 \\ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & \mapsto & \phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}, \end{array}$$

with the kernel

$$k(x, y) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}^\top \begin{bmatrix} y_1 \\ y_2 \\ y_1 y_2 \end{bmatrix}$$

(i.e., the standard inner product in $\mathbb{R}^3$ between the features). This is a valid kernel in the sense we've considered up till now. We denote by $\mathcal{H}$ this feature space.

Let's now define a function of the features x_1, x_2, and $x_1 x_2$ of x, namely:

$$f(x) = a x_1 + b x_2 + c x_1 x_2.$$

This function is a member of a space of functions mapping from $\mathcal{X} = \mathbb{R}^2$ to $\mathbb{R}$. We can define an equivalent representation for f,

$$f(\cdot) = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{4.1}$$

The notation $f(\cdot)$ refers to the function itself, in the abstract (and in fact, this function might have multiple equivalent representations). We sometimes write f rather than $f(\cdot)$, when there is no ambiguity. The notation $f(x) \in \mathbb{R}$ refers to the function evaluated at a particular point (which is just a real number). With this notation, we can write

$$\begin{aligned} f(x) &= f(\cdot)^\top \phi(x) \\ &:= \langle f(\cdot), \phi(x) \rangle_{\mathcal{H}} \end{aligned}$$

In other words, the evaluation of f at x can be written as an **inner product in feature space** (in this case, just the standard inner product in $\mathbb{R}^3$), and $\mathcal{H}$ is a space of functions mapping $\mathbb{R}^2$ to $\mathbb{R}$. This construction can still be used if there are infinitely many features: from the Cauchy-Schwarz argument in Lemma 9, we may write

$$f(x) = \sum_{\ell=1}^{\infty} f_\ell \phi_\ell(x), \tag{4.2}$$

where the expression is bounded in absolute value as long as $\{f_\ell\}_{\ell=1}^{\infty} \in \ell_2$ (of course, we can't write this function explicitly, since we'd need to enumerate all the f_ℓ).

This line of reasoning leads us to a conclusion that might seem counterintuitive at first: we've seen that $\phi(x)$ is a mapping from $\mathbb{R}^2$ to $\mathbb{R}^3$, but it also defines (the parameters of) a *function* mapping $\mathbb{R}^2$ to $\mathbb{R}$. To see why this is so, we write

$$k(\cdot, y) = \begin{bmatrix} y_1 \\ y_2 \\ y_1 y_2 \end{bmatrix} = \phi(y),$$

using the same convention as in (4.1). This is certainly valid: if you give me a y, I'll give you a vector $k(\cdot, y)$ in $\mathcal{H}$ such that

$$\langle k(\cdot, y), \phi(x) \rangle_{\mathcal{H}} = ax_1 + bx_2 + cx_1x_2,$$

where $a = y_1$, $b = y_2$, and $c = y_1 y_2$ (i.e., for every y, we get a different vector $\begin{bmatrix} a & b & c \end{bmatrix}^\top$). But due to the symmetry of the arguments, we could equally well have written

$$\begin{aligned} \langle k(\cdot, x), \phi(y) \rangle &= uy_1 + vy_2 + wy_1y_2 \\ &= k(x, y). \end{aligned}$$

In other words, we can write $\phi(x) = k(\cdot, x)$ and $\phi(y) = k(\cdot, y)$ without ambiguity. This way of writing the feature mapping is called the **canonical feature map** [11, Lemma 4.19].

This example illustrates the two defining features of an RKHS:

- The feature map of every point is in the feature space:

$$\forall x \in \mathcal{X}, \;\; k(\cdot, x) \in \mathcal{H}$$

,

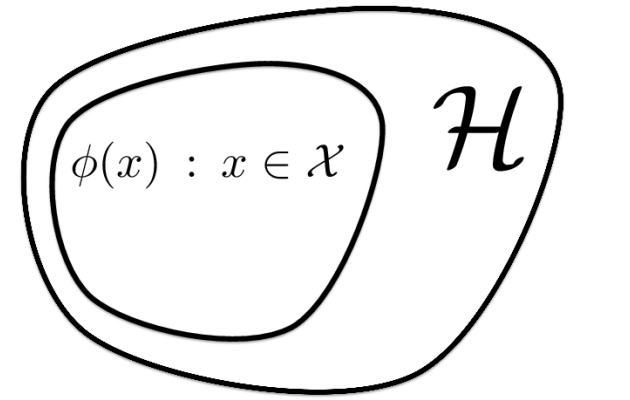

Figure 4.1: Feature space and mapping of input points.

- **The reproducing property**:

$$\forall x \in \mathcal{X},\, \forall f \in \mathcal{H},\ \ \langle f, k(\cdot, x)\rangle_{\mathcal{H}} = f(x) \tag{4.3}$$

In particular, for any $x, y \in \mathcal{X}$,

$$k(x, y) = \langle k(\cdot, x), k(\cdot, y)\rangle_{\mathcal{H}}.$$

Another, more subtle point to take from this example is that $\mathcal{H}$ is in this case larger than the set of all $\phi(x)$ (see Figure 4.1). For instance, when writing f in (4.1), we could choose $f = [1\ 1\ -1] \in \mathcal{H}$, but this cannot be obtained by the feature map $\phi(x) = [x_1\ x_2\ (x_1 x_2)]$.

4.1.2 Example: RKHS defined by via a Fourier series

Consider a function on the interval $[-\pi, \pi]$ with periodic boundary conditions. This may be expressed as a Fourier series,

$$f(x) = \sum_{l=-\infty}^{\infty} \hat{f}_l \exp(\imath l x),$$

using the orthonormal basis on $[-\pi, \pi]$, noting that

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \exp(\imath \ell x)\overline{\exp(\imath m x)}dx = \begin{cases} 1 & \ell = m, \\ 0 & \ell \neq m, \end{cases}$$

where $\imath = \sqrt{-1}$, and $\bar{a}$ is the complex conjugate of a. We assume $f(x)$ is real, so its Fourier transform is conjugate symmetric,

$$\hat{f}_{-\ell} = \overline{\hat{f}_\ell}.$$

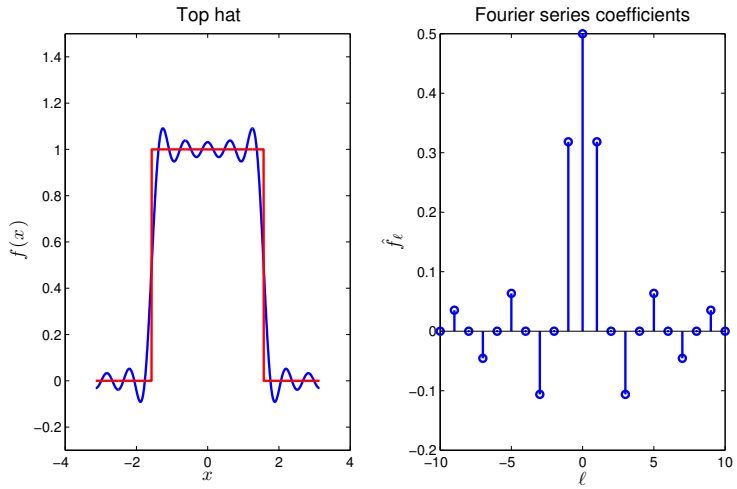

Figure 4.2: "Top hat" function (red) and its approximation via a Fourier series (blue). Only the first 21 terms are used; as more terms are added, the Fourier representation gets closer to the desired function.

As an illustration, consider the "top hat" function,

$$f(x) = \begin{cases} 1 & |x| < T, \\ 0 & T \leq |x| < \pi, \end{cases}$$

with Fourier series

$$\hat{f}_\ell := \frac{\sin(\ell T)}{\ell \pi} \qquad f(x) = \sum_{\ell=0}^{\infty} 2\hat{f}_\ell \cos(\ell x).$$

Due to the symmetry of the Fourier coefficients and the asymmetry of the sine function, the sum can be written over positive ℓ, and only the cosine terms remain. See Figure 4.2.
Assume the kernel takes a single argument, which is the difference in its inputs,

$$k(x, y) = k(x - y),$$

and define the Fourier series representation of k as

$$k(x) = \sum_{l=-\infty}^{\infty} \hat{k}_l \exp(\imath l x), \tag{4.4}$$

where $\hat{k}_{-l} = \hat{k}_l$ and $\overline{\hat{k}_l} = \hat{k}_l$ (a real and symmetric $k$has a real and symmetric Fourier transform). For instance, when the kernel is a Jacobi Theta function ϑ (which looks close to a Gaussian when σ^2 is sufficiently narrower than $[-\pi, \pi]$),

$$k(x) = \frac{1}{2\pi}\vartheta\left(\frac{x}{2\pi}, \frac{\imath\sigma^2}{2\pi}\right), \qquad \hat{k}_\ell \approx \frac{1}{2\pi}\exp\left(\frac{-\sigma^2\ell^2}{2}\right),$$

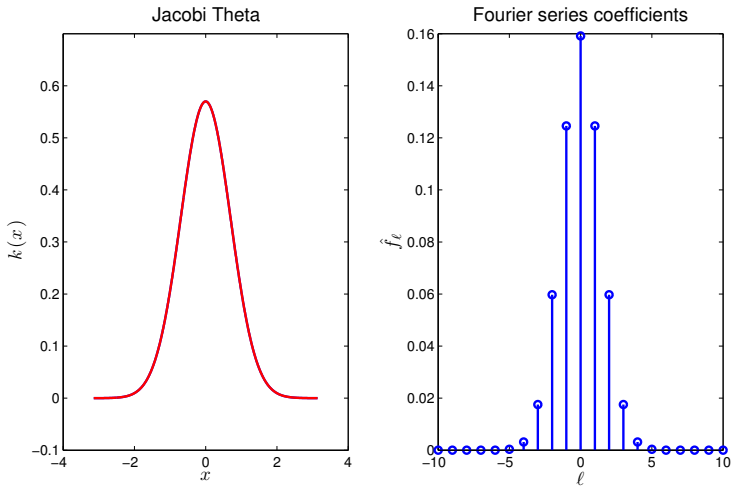

Figure 4.3: Jacobi Theta kernel (red) and its Fourier series representation, which is Gaussian (blue). Again, only the first 21 terms are retained, however the approximation is already very accurate (bearing in mind the Fourier series coefficients decay exponentially).

and the Fourier coefficients are Gaussian (evaluated on a discrete grid). See Figure 4.3.

Recall the standard dot product in L_2, where we take the conjugate of the right-hand term due to the complex valued arguments,

$$\begin{aligned}\langle f, g\rangle_{L_2} &= \left\langle \sum_{\ell=-\infty}^{\infty} \hat{f}_\ell \exp(\imath \ell x), \sum_{m=-\infty}^{\infty} \overline{\hat{g}_m \exp(\imath m x)} \right\rangle_{L_2} \\ &= \sum_{\ell=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{f}_\ell \overline{\hat{g}}_\ell \left\langle \exp(\imath \ell x), \exp(-\imath m x)\right\rangle_{L_2} \\ &= \sum_{\ell=-\infty}^{\infty} \hat{f}_\ell \overline{\hat{g}}_\ell.\end{aligned}$$

We define the dot product in $\mathcal{H}$ to be a roughness penalized dot product, taking the form[4]

$$\langle f, g\rangle_{\mathcal{H}} = \sum_{\ell=-\infty}^{\infty} \frac{\hat{f}_\ell \overline{\hat{g}}_\ell}{\hat{k}_\ell}. \tag{4.5}$$

[4]Note: while this dot product has been defined using the Fourier transform of the kernel, additional technical conditions are required of the kernel for a valid RKHS to be obtained. These conditions are given by Mercer's theorem [11, Theorem 4.49], which when satisfied, imply that the expansion (4.4) converges absolutely and uniformly.

The squared norm of a function f in $\mathcal{H}$ enforces smoothness:

$$\|f\|_{\mathcal{H}}^2 = \langle f, f\rangle_{\mathcal{H}} = \sum_{l=-\infty}^{\infty} \frac{\hat{f}_\ell \overline{\hat{f}_\ell}}{\hat{k}_\ell} = \sum_{l=-\infty}^{\infty} \frac{\left|\hat{f}_\ell\right|^2}{\hat{k}_\ell}. \tag{4.6}$$

If $\hat{k}_\ell$ decays fast, then so must $\hat{f}_\ell$ if we want $\|f\|_{\mathcal{H}}^2 < \infty$. From this norm definition, we see that the RKHS functions are a subset of the functions in L_2, for which finiteness of the norm $\|f\|_{L_2}^2 = \sum_{\ell=-\infty}^{\infty} \left|\hat{f}_\ell\right|^2$ is required (this being less restrictive than 4.6).

We next check whether the reproducing property holds for a function $f(x) \in \mathcal{H}$. Define a function

$$g(x) := k(x-z) = \sum_{\ell=-\infty}^{\infty} \exp(\imath\ell x) \underbrace{\hat{k}_\ell \exp(-\imath\ell z)}_{\hat{g}_\ell}$$

Then for a function[5] $f(\cdot) \in \mathcal{H}$,

$$\begin{aligned} \langle f(\cdot), k(\cdot, z)\rangle_{\mathcal{H}} &= \langle f(\cdot), g(\cdot)\rangle_{\mathcal{H}} \\ &= \sum_{\ell=-\infty}^{\infty} \frac{\hat{f}_\ell \left(\overline{\hat{k}_\ell \exp(-\imath\ell z)}\right)}{\hat{k}_\ell} \\ &= \sum_{\ell=-\infty}^{\infty} \hat{f}_\ell \exp(\imath\ell z) \;=\; f(z). \end{aligned} \tag{4.7}$$

Finally, as a special case of the above, we verify the reproducing property for the kernel itself. Recall kernel definition,

$$k(x-y) = \sum_{\ell=-\infty}^{\infty} \hat{k}_\ell \exp(\imath\ell(x-y)) = \sum_{\ell=-\infty}^{\infty} \hat{k}_\ell \exp(\imath\ell x)\exp(-\imath\ell y)$$

Define two functions of a variable x as kernels centered at y and z, respectively,

$$\begin{aligned} f(x) := k(x-y) &= \sum_{\ell=-\infty}^{\infty} \hat{k}_\ell \exp(\imath\ell(x-y)) \\ &= \sum_{\ell=-\infty}^{\infty} \exp(\imath\ell x) \underbrace{\hat{k}_\ell \exp(-\imath\ell y)}_{\hat{f}_\ell} \\ g(x) := k(x-z) &= \sum_{\ell=-\infty}^{\infty} \exp(\imath\ell x) \underbrace{\hat{k}_\ell \exp(-\imath\ell z)}_{\hat{g}_\ell} \end{aligned}$$

[5]Exercise: what happens if we change the order, and write $\langle f(\cdot), k(\cdot, x)\rangle_{\mathcal{H}}$? Hint: $f(x) = \overline{f(x)}$ since the function is real-valued.

Applying the dot product definition in $\mathcal{H}$, we obtain

$$\begin{aligned}\langle k(\cdot,y),k(\cdot,z)\rangle_{\mathcal{H}} &= \langle f,g\rangle_{\mathcal{H}}\\ &= \sum_{\ell=-\infty}^{\infty}\frac{\hat{f}_\ell\overline{\hat{g}_\ell}}{\hat{k}_\ell}\\ &= \sum_{\ell=-\infty}^{\infty}\frac{\left(\hat{k}_\ell\exp(-\imath\ell y)\right)\left(\overline{\hat{k}_\ell\exp(-\imath\ell z)}\right)}{\hat{k}_\ell}\\ &= \sum_{\ell=-\infty}^{\infty}\hat{k}_\ell\exp(\imath\ell(z-y)) \;=\; k(z-y).\end{aligned}$$

You might be wondering how the dot product in (4.5) relates to our original definition of an RKHS function in (4.2): the latter equation, updated to reflect that the features are complex-valued (and changing the sum index to run from $-\infty$ to ∞) is

$$f(x)=\sum_{\ell=-\infty}^{\infty}f_\ell\overline{\phi_\ell(x)},$$

which is an expansion in terms of coefficients f_ℓ and features $\phi_\ell(x)$. Writing the dot product from (4.7) earlier,

$$\begin{aligned}\langle f(\cdot),k(\cdot,z)\rangle_{\mathcal{H}} &= \sum_{\ell=-\infty}^{\infty}\frac{\hat{f}_\ell\left(\overline{\hat{k}_\ell\exp(-\imath\ell z)}\right)}{\hat{k}_\ell}\\ &= \sum_{\ell=-\infty}^{\infty}\frac{\hat{f}_\ell}{\sqrt{\hat{k}_\ell}}\left(\sqrt{\hat{k}_\ell}\left(\overline{\exp(-\imath\ell z)}\right)\right),\end{aligned}$$

it's clear that

$$f_\ell=\frac{\hat{f}_\ell}{\sqrt{\hat{k}_\ell}},\qquad\qquad \phi_\ell(x)=\sqrt{\hat{k}_\ell}\left(\exp(-\imath\ell z)\right),$$

and for this feature definition, the reproducing property holds,

$$\langle\phi_\ell(x),\phi_\ell(x')\rangle_{\mathcal{H}}=\sum_{\ell=-\infty}^{\infty}\left(\sqrt{\hat{k}_\ell}\left(\exp(-\imath\ell x)\right)\right)\left(\sqrt{\hat{k}_\ell}\left(\overline{\exp(-\imath\ell x')}\right)\right)=k(x-x').$$

4.1.3 Example: RKHS defined using the exponentiated quadratic kernel on $\mathbb{R}$

Let's now consider the more general setting of kernels on $\mathbb{R}$, where we can no longer use the Fourier series expansion (the arguments in this section also apply to the multivariate case $\mathbb{R}^d$). Our discussion follows [2, Sections 3.1 - 3.3]. We

start by defining the eigenexpansion of $k(x, x')$ with respect to a non-negative finite measure μ on $\mathcal{X} := \mathbb{R}$,

$$\lambda_i e_i(x) = \int k(x, x') e_i(x') d\mu(x'), \qquad \int_{L_2(\mu)} e_i(x) e_j(x) d\mu(x) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases} \tag{4.8}$$

For the purposes of this example, we'll use the Gaussian density μ, meaning

$$d\mu(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-x^2\right) dx \tag{4.9}$$

We can write

$$k(x, x') = \sum_{\ell=1}^{\infty} \lambda_\ell e_\ell(x) e_\ell(x'), \tag{4.10}$$

which converges in $L_2(\mu)$.[6] If we choose an exponentiated quadratic kernel,

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right),$$

the eigenexpansion is

$$\begin{aligned} \lambda_k &\propto b^k \qquad b < 1 \\ e_k(x) &\propto \exp(-(c - a)x^2) H_k(x\sqrt{2c}), \end{aligned}$$

where a, b, c are functions of σ, and H_k is kth order Hermite polynomial [6, Section 4.3]. Three eigenfunctions are plotted in Figure 4.4.

We are given two functions f, g in $L_2(\mu)$, expanded in the orthonormal system $\{e_\ell\}_{\ell=1}^{\infty}$,

$$f(x) = \sum_{\ell=1}^{\infty} \hat{f}_\ell e_\ell(x) \qquad g(x) = \sum_{\ell=1}^{\infty} \hat{g}_\ell e_\ell(x), \tag{4.11}$$

The standard dot product in $L_2(\mu)$ between f, g is

$$\begin{aligned} \langle f, g \rangle_{L_2(\mu)} &= \left\langle \sum_{\ell=1}^{\infty} \hat{f}_\ell e_\ell(x), \sum_{\ell=1}^{\infty} \hat{g}_\ell e_\ell(x) \right\rangle_{L_2(\mu)} \\ &= \sum_{\ell=1}^{\infty} \hat{f}_\ell \hat{g}_\ell. \end{aligned}$$

As with the Fourier case, we will define the dot product in $\mathcal{H}$ to have a roughness penalty, yielding

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{\ell=1}^{\infty} \frac{\hat{f}_\ell \hat{g}_\ell}{\lambda_\ell} \qquad \|f\|_{\mathcal{H}}^2 = \sum_{\ell=1}^{\infty} \frac{\hat{f}_\ell^2}{\lambda_\ell}, \tag{4.12}$$

[6] As with the Fourier example in Section (4.1.2), there are certain technical conditions needed when defining an RKHS kernel, to ensure that the sum in (4.10) converges in a stronger sense than $L_2(\mu)$. This requires a generalization of Mercer's theorem to non-compact domains.

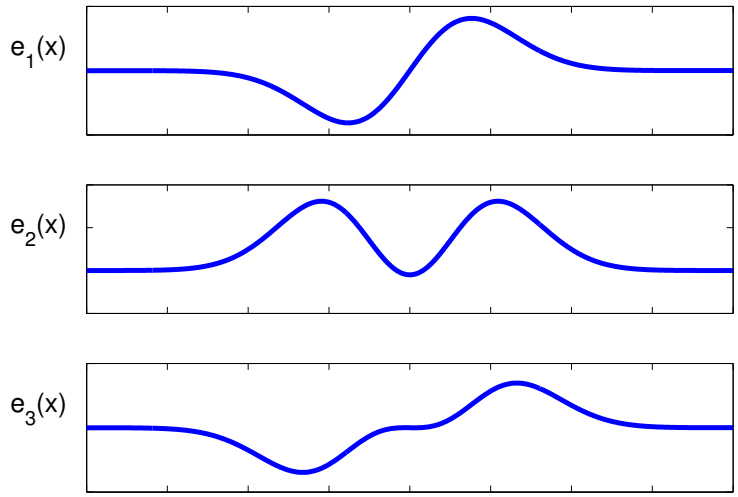

Figure 4.4: First three eigenfunctions for the exponentiated quadratic kernel with respect to a Gaussian measure μ.

where you should compare with (4.5) and (4.6). The RKHS functions are a subset of the functions in $L_2(\mu)$, with norm $\|f\|_{L_2}^2 = \sum_{\ell=1}^{\infty} \hat{f}_\ell^2 < \infty$ (less restrictive than 4.12).

Also just like the Fourier case, we can explicitly construct the feature map that gives our original expression of the RHKS function in (4.2), namely

$$f(x) = \sum_{\ell=1}^{\infty} f_\ell \phi_\ell(x).$$

We write the kernel centered at x as

$$g(x) := k(x - z) = \sum_{\ell=1}^{\infty} e_\ell(x) \underbrace{\lambda_\ell e_\ell(z)}_{\hat{g}_\ell}$$

Beginning with (4.11), we get

$$\begin{aligned} f(x) = \langle f, g \rangle_{\mathcal{H}} &= \sum_{\ell=1}^{\infty} \frac{\hat{f}_\ell \hat{g}_\ell}{\lambda_\ell} \\ &= \sum_{\ell=1}^{\infty} \frac{\hat{f}_\ell \left(\lambda_\ell e_\ell(z)\right)}{\lambda_\ell} \\ &= \sum_{\ell=1}^{\infty} \frac{\hat{f}_\ell}{\sqrt{\lambda_\ell}} \left(\sqrt{\lambda_\ell} e_\ell(z)\right), \end{aligned}$$

hence

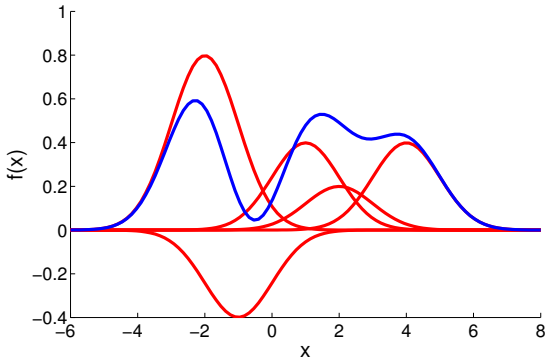

Figure 4.5: An RKHS function. The kernel here is an exponentiated quadratic. The blue function is obtained by taking the sum of red kernels, which are centred at x_i and scaled by α_i.

$$f_\ell = \frac{\hat{f}_\ell}{\sqrt{\lambda_\ell}} \qquad \phi_\ell(x) = \sqrt{\lambda_\ell} e_\ell(x), \tag{4.13}$$

and the reproducing property holds,[7]

$$\sum_{\ell=1}^{\infty} \phi_\ell(x)\phi_\ell(x') = k(x, x').$$

4.1.4 Tractable form of functions in an infinite dimensional RKHS, and explicit feature space construction

When a feature space is infinite dimensional, functions are generally expressed as linear combinations of kernels at particular points, such that the features need never be explicitly written down. The key is to satisfy the reproducing property in eq. (4.3) (and in Definition (15) below). Let's see, as an example, an RKHS function for an exponentiated quadratic kernel,

$$f(x) := \sum_{i=1}^{m} \alpha_i k(x_i, x). \tag{4.14}$$

We show an example function in Figure 4.5.

[7] Note also that the features are square summable, since

$$\begin{aligned} \|\phi(x)\|_{\mathcal{H}}^2 = \|\phi(x)\|_{\ell_2}^2 &= \sum_{\ell=1}^{\infty} \lambda_\ell e_\ell(x) e_\ell(x) \\ &= k(x, x) < \infty. \end{aligned}$$

The eigendecomposition in (4.10) and the feature definition in (4.13) yield

$$k(x,x') = \sum_{\ell=1}^{\infty} \underbrace{\left[\sqrt{\lambda_\ell} e_\ell(x)\right]}_{\phi_\ell(x)} \underbrace{\left[\sqrt{\lambda_\ell} e_\ell(x')\right]}_{\phi_\ell(x')}.$$

and (4.14) can be rewritten

$$f(x) = \sum_{\ell=1}^{\infty} f_\ell \phi_\ell(x) = \sum_{\ell=1}^{\infty} f_\ell \underbrace{\left[\sqrt{\lambda_\ell} e_\ell(x)\right]}_{\phi_\ell(x)}, \tag{4.15}$$

where

$$f_\ell = \sum_{i=1}^{m} \alpha_i \sqrt{\lambda_\ell} \phi_\ell(x_i).$$

The coefficients $\{f_\ell\}_{\ell=1}^{\infty}$ are square summable since

$$\|f\|_{\ell_2} = \left\| \sum_{i=1}^{m} \alpha_i \phi(x_i) \right\| \leq \sum_{i=1}^{m} |\alpha_i| \, \|\phi(x_i)\| < \infty.$$

The key point is that we need never explicitly compute the eigenfunctions e_ℓ or the eigenexpansion (4.10) to specify functions in the RKHS: we simply write our functions in terms of the kernels, as in (4.14).

4.2 Formal definitions

In this section, we cover the **reproducing property**, which is what makes a Hilbert space a reproducing kernel Hilbert space (RKHS). We next show that every reproducing kernel Hilbert space has a unique positive definite kernel, and vice-versa: this is the Moore-Aronszajn theorem.

Our discussion of the reproducing property follows [1, Ch. 1] and [11, Ch. 4]. We use the notation $f(\cdot)$ to indicate we consider the function itself, and not just the function evaluated at a particular point. For the kernel $k(x_i, \cdot)$, one argument is fixed at x_i, and the other is free (recall the kernel is symmetric).

Definition 15 (Reproducing kernel Hilbert space (first definition))**.** [1, p. 7] Let $\mathcal{H}$ be a Hilbert space of $\mathbb{R}$-valued functions defined on a non-empty set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called *a reproducing kernel*[8] of $\mathcal{H}$, and $\mathcal{H}$ is a reproducing kernel Hilbert space, if k satisfies

- $\forall x \in \mathcal{X}, \;\; k(\cdot, x) \in \mathcal{H}$,
- $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \;\; \langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$ (the reproducing property).

[8] We've deliberately used the same notation for the kernel as we did for positive definite kernels earlier. We will see in the next section that we are referring in both cases to the same object.

In particular, for any $x, y \in \mathcal{X}$,

$$k(x,y) = \langle k(\cdot,x), k(\cdot,y)\rangle_{\mathcal{H}}. \tag{4.16}$$

Recall that a kernel is an inner product between feature maps: then $\phi(x) = k(\cdot,x)$ is a valid feature map (so every reproducing kernel is indeed a kernel in the sense of Definition (3)).

The reproducing property has an interesting consequence for functions in $\mathcal{H}$. We define δ_x to be the operator of evaluation at x, i.e.

$$\delta_x f = f(x) \quad \forall f \in \mathcal{H},\ x \in \mathcal{X}.$$

We then get the following *equivalent* definition for a reproducing kernel Hilbert space.

Definition 16 (Reproducing kernel Hilbert space (second definition))**.** [11, Definition 4.18] $\mathcal{H}$ is an RKHS if for all $x \in \mathcal{X}$, the evaluation operator δ_x is bounded: there exists a corresponding $\lambda_x \geq 0$ such that $\forall f \in \mathcal{H}$,

$$|f(x)| = |\delta_x f| \leq \lambda_x \|f\|_{\mathcal{H}}$$

This definition means that when two functions are identical in the RHKS norm, they agree at every point:

$$|f(x) - g(x)| = |\delta_x (f-g)| \leq \lambda_x \|f-g\|_{\mathcal{H}} \quad \forall f, g \in \mathcal{H}.$$

This is a particularly useful property[9] if we're using RKHS functions to make predictions at a given point x, by optimizing over $f \in \mathcal{H}$. That these definitions are equivalent is shown in the following theorem.

Theorem 17 (Reproducing kernel equivalent to bounded δ_x)**.** *[1, Theorem 1] $\mathcal{H}$ is a reproducing kernel Hilbert space (i.e., its evaluation operators δ_x are bounded linear operators), if and only if $\mathcal{H}$ has a reproducing kernel.*

Proof. We only prove here that if $\mathcal{H}$ has a reproducing kernel, then δ_x is a bounded linear operator. The proof in the other direction is more complicated [11, Theorem 4.20], and will be covered in the advanced part of the course (briefly, it uses the Riesz representer theorem).

Given that a Hilbert space $\mathcal{H}$ has a reproducing kernel k with the reproducing property $\langle f, k(\cdot,x)\rangle_{\mathcal{H}} = f(x)$, then

$$\begin{aligned} |\delta_x[f]| &= |f(x)| \\ &= |\langle f, k(\cdot,x)\rangle_{\mathcal{H}}| \\ &\leq \|k(\cdot,x)\|_{\mathcal{H}} \|f\|_{\mathcal{H}} \\ &= \langle k(\cdot,x), k(\cdot,x)\rangle_{\mathcal{H}}^{1/2} \|f\|_{\mathcal{H}} \\ &= k(x,x)^{1/2} \|f\|_{\mathcal{H}} \end{aligned}$$

where the third line uses the Cauchy-Schwarz inequality. Consequently, $\delta_x : \mathcal{F} \to \mathbb{R}$ is a bounded linear operator where $\lambda_x = k(x,x)^{1/2}$. □

[9] This property certainly does not hold for all Hilbert spaces: for instance, it fails to hold on the set of square integrable functions $L_2(\mathcal{X})$.

Finally, the following theorem is very fundamental [1, Theorem 3 p. 19], [11, Theorem 4.21], and will be proved in the advanced part of the course:

Theorem 18 (Moore-Aronszajn)**.** *[1, Theorem 3] Every positive definite kernel k is associated with a unique RKHS $\mathcal{H}$.*

Note that the feature map is *not* unique (as we saw earlier): only the kernel is. Functions in the RKHS can be written as linear combinations of feature maps,

$$f(\cdot) := \sum_{i=1}^{m} \alpha_i k(x_i, \cdot),$$

as in Figure 4.5, as well as the limits of Cauchy sequences (where we can allow $m \to \infty$).

5 Application 1: Distance between means in feature space

Suppose we have two distributions p, q and we sample $(x_i)_{i=1}^m$ from p and $(y_i)_{i=1}^n$ from q. What is the distance between their means *in feature space*? This exercise illustrates that using the reproducing property, you can compute this distance without ever having to evaluate the feature map.

Answer:

$$\begin{aligned}
\left\| \frac{1}{m}\sum_{i=1}^{m}\phi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(y_j) \right\|_{\mathcal{H}}^2 &= \left\langle \frac{1}{m}\sum_{i=1}^{m}\phi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(y_j), \frac{1}{m}\sum_{i=1}^{m}\phi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(y_j) \right\rangle_{\mathcal{H}} \\
&= \frac{1}{m^2}\left\langle \sum_{i=1}^{m}\phi(x_i), \sum_{i=1}^{m}\phi(x_i) \right\rangle + \dots \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} k(x_i, x_j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k(y_i, y_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i, y_j).
\end{aligned}$$

What might this distance be useful for? In the case $\phi(x) = x$, we can use this statistic to distinguish distributions with different means. If we use the feature mapping $\phi(x) = [x\ x^2]$ we can distinguish both means and variances. More complex feature spaces permit us to distinguish increasingly complex features of the distributions. As we'll see in much more detail later in the course, there are kernels that can distinguish *any* two distributions [3, 10].

6 Application 2: Kernel PCA

This is one of the most famous kernel algorithms: see [7, 8].

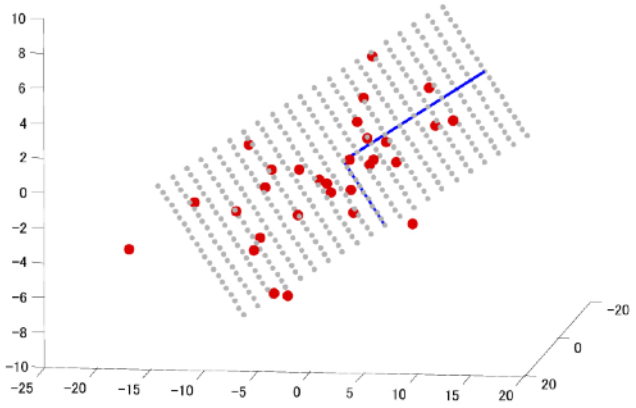

Figure 6.1: PCA in $\mathbb{R}^3$, for data in a two-dimensional subspace. The blue lines represent the first two principal directions, and the grey dots represent the 2-D plane in $\mathbb{R}^3$ on which the data lie (figure by Kenji Fukumizu).

6.1 Description of the algorithm

Goal of classical PCA: to find a d-dimensional subspace of a higher dimensional space (D-dimensional, $\mathbb{R}^D$) containing the directions of maximum variance. See Figure 6.1.

$$\begin{aligned} u_1 &= \arg\max_{\|u\|\leq 1} \frac{1}{n}\sum_{i=1}^{n}\left(u^\top\left(x_i - \frac{1}{n}\sum_{i=1}^{n} x_i\right)\right)^2 \\ &= \arg\max_{\|u\|\leq 1} u^\top C u \end{aligned}$$

where

$$\begin{aligned} C &= \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \frac{1}{n}\sum_{i=1}^{n} x_i\right)\left(x_i - \frac{1}{n}\sum_{i=1}^{n} x_i\right)^\top \\ &= \frac{1}{n} X H X^\top, \end{aligned}$$

where $X = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}$, $H = I - n^{-1}\mathbf{1}_{n\times n}$, and $\mathbf{1}_{n\times n}$ is an $n \times n$ matrix of ones (note that $H = HH$, i.e. the matrix H is idempotent). We've looked at the first principal component, but all of the principal components u_i are the eigenvectors of the covariance matrix C (thus, each is orthogonal to all the previous ones). We have the eigenvalue equation

$$n\lambda_i u_i = C u_i.$$

We now do this in feature space:

$$\begin{aligned} f_1 &= \arg\max_{\|f\|_{\mathcal{H}}\leq 1} \frac{1}{n}\sum_{i=1}^{n}\left(\left\langle f, \phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right\rangle_{\mathcal{H}}\right)^2 \\ &= \arg\max_{\|f\|_{\mathcal{H}}\leq 1} \mathrm{var}(f). \end{aligned}$$

First, observe that we can write

$$\begin{aligned} f_\ell &= \sum_{i=1}^{n}\alpha_{\ell i}\left(\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right), \\ &= \sum_{i=1}^{n}\alpha_{\ell i}\tilde{\phi}(x_i), \end{aligned}$$

since any component orthogonal to the span of $\tilde{\phi}(x_i) := \phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$ vanishes when we take the inner product. The f are now elements of the feature space: if we were to use a Gaussian kernel, we could plot the *function* f by choosing the canonical feature map $\phi(x) = k(x, \cdot)$.

We can also define an infinite dimensional analog of the covariance:

$$\begin{aligned} C &= \frac{1}{n}\sum_{i=1}^{n}\left(\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right)\otimes\left(\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right), \\ &= \frac{1}{n}\sum_{i=1}^{n}\tilde{\phi}(x_i)\otimes\tilde{\phi}(x_i) \end{aligned}$$

where we use the definition

$$(a\otimes b)c := \langle b, c\rangle_{\mathcal{H}}\, a \tag{6.1}$$

this is analogous to the case of finite dimensional vectors, $(ab^\top)c = a(b^\top c)$. Writing this, we get

$$f_\ell\lambda_\ell = Cf_\ell. \tag{6.2}$$

Let's look at the right hand side: to apply (6.1), we use

$$\begin{aligned} &\left\langle \tilde{\phi}(x_i), \sum_{i=1}^{n}\alpha_{\ell i}\tilde{\phi}(x_i)\right\rangle_{\mathcal{H}} \\ =&\sum_{j=1}^{n}\alpha_{\ell j}\tilde{k}(x_i, x_j), \end{aligned}$$

where $\tilde{k}(x_i, x_j)$ is the (i,j)th entry of the matrix $\tilde{K} := HKH$ (this is an easy exercise!). Thus,

$$Cf_\ell = \frac{1}{n}\sum_{i=1}^{n}\beta_{\ell i}\tilde{\phi}(x_i), \qquad \beta_{\ell i} = \sum_{j=1}^{n}\alpha_{\ell j}\tilde{k}(x_i, x_j).$$

We can now project both sides of (6.2) onto each of the centred mappings $\tilde{\phi}(x_q)$: this gives a set of equations which must all be satisifed to get an equivalent eigenproblem to (6.2). This gives

$$\left\langle \tilde{\phi}(x_q), \mathrm{LHS} \right\rangle_{\mathcal{H}} = \lambda_\ell \left\langle \tilde{\phi}(x_q), f_\ell \right\rangle = \lambda_\ell \sum_{i=1}^{n} \alpha_{\ell i} \tilde{k}(x_q, x_i) \qquad \forall q \in \{1 \dots n\}$$

$$\left\langle \tilde{\phi}(x_q), \mathrm{RHS} \right\rangle_{\mathcal{H}} = \left\langle \tilde{\phi}(x_q), C f_\ell \right\rangle_{\mathcal{H}} = \frac{1}{n} \sum_{i=1}^{n} \tilde{k}(x_q, x_i) \left(\sum_{j=1}^{n} \alpha_{\ell j} \tilde{k}(x_i, x_j) \right) \qquad \forall q \in \{1 \dots n\}$$

Writing this as a matrix equation,

$$n\lambda_\ell \widetilde{K} \alpha_\ell = \widetilde{K}^2 \alpha_\ell,$$

or equivalently

$$n\lambda_\ell \alpha_\ell = \widetilde{K} \alpha_\ell. \tag{6.3}$$

Thus the α_ℓ are the eigenvectors of $\widetilde{K}$: it is not necessary to ever use the feature map $\phi(x_i)$ explicitly!

How do we ensure the eigenfunctions f have unit norm in feature space?

$$\begin{aligned}
&\|f\|_{\mathcal{H}}^2 \\
&= \left\langle \sum_{i=1}^{n} \alpha_i \tilde{\phi}(x_i), \sum_{i=1}^{n} \alpha_i \tilde{\phi}(x_i) \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i \left\langle \tilde{\phi}(x_i), \tilde{\phi}(x_j) \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_i \tilde{k}(x_i, x_j) \\
&= \alpha^\top \widetilde{K} \alpha = n\lambda \alpha^\top \alpha = n\lambda \|\alpha\|^2.
\end{aligned}$$

Thus to re-normalise α such that $\|f\| = 1$, it suffices to replace

$$\alpha \leftarrow \alpha / \sqrt{n\lambda}$$

(assuming the original solutions to (6.3) have $\|\alpha\| = 1$).

How do you project a new point x^* onto the principal component f? Assuming f is properly normalised, the projection is

$$\begin{aligned}
P_f \phi(x^*) &= \langle \phi(x^*), f \rangle_{\mathcal{H}} f \\
&= \sum_{i=1}^{n} \alpha_i \left(\sum_{j=1}^{n} \alpha_j \left\langle \phi(x^*), \tilde{\phi}(x_i) \right\rangle_{\mathcal{H}} \right) \tilde{\phi}(x_i) \\
&= \sum_{i=1}^{n} \alpha_i \left(\sum_{j=1}^{n} \alpha_j \left(k(x^*, x_j) - \frac{1}{n} \sum_{\ell=1}^{n} k(x^*, x_\ell) \right) \right) \tilde{\phi}(x_i).
\end{aligned}$$

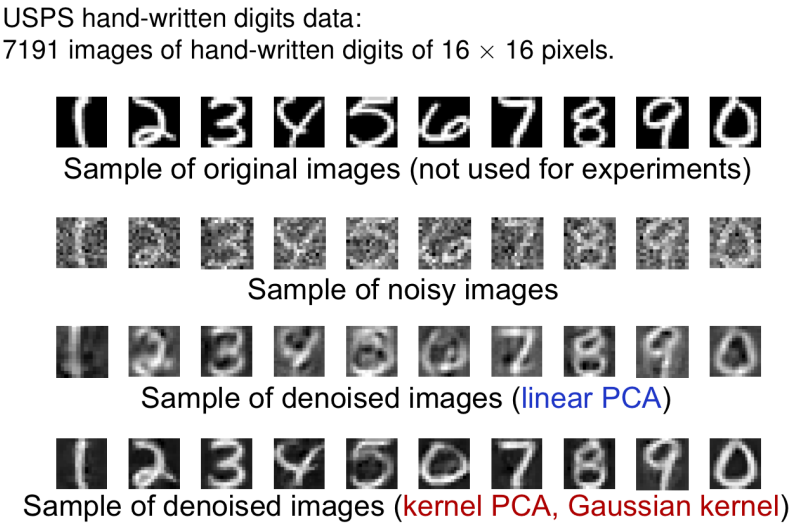

Figure 6.2: Hand-written digit denoising example (from Kenji Fukumizu's slides).

6.2 Example: image denoising

We consider the problem of denoising hand-written digits. Denote by

$$P_d\phi(x^*) = \quad P_{f_1}\phi(x^*) \quad + \ldots + \quad P_{f_d}\phi(x^*)$$

the projection of $\phi(x^*)$ onto one of the first d eigenvectors from kernel PCA (recall these are orthogonal). We define the nearest point $y \in \mathcal{X}$ to this feature space projection as the solution to the problem

$$y^* = \arg\min_{y \in \mathcal{X}} \|\phi(y) - P_d\phi(x^*)\|^2_{\mathcal{H}} .$$

In many cases, it will not be possible to reduce the squared error to zero, as there will be no single y^* corresponding to an exact solution. As in linear PCA, we can use the projection onto a subspace for denoising. By doing this in feature space, we can take into account the fact that data may not be distributed as a simple Gaussian, but can lie in a submanifold in input space, which nonlinear PCA can discover. See Figure 6.2.

7 Application 3: Ridge regression

In this section, we describe ridge regression. This is the algorithm used for the regression plots at the start of the document (Figure 2.3): it is very simple to implement and usually works quite well (except when the data have outliers, since it uses a squared loss).

7.1 A loss-based interpretation

7.1.1 Finite dimensional case

This discussion may be found in a number of sources. We draw from [9, Section 2.2]. We are given n training points in $\mathbb{R}^D$, which we arrange in a matrix $X = \left[\begin{array}{ccc} x_1 & \dots & x_n \end{array}\right] \in \mathbb{R}^{D\times n}$. To each of these points, there corresponds an output y_i, which we arrange in a column vector $y := \left[\begin{array}{ccc} y_1 & \dots & y_n \end{array}\right]^\top$. Define some $\lambda > 0$. Our goal is:

$$\begin{aligned} a^* &= \arg\min_{a\in\mathbb{R}^D} \left(\sum_{i=1}^n (y_i - x_i^\top a)^2 + \lambda \|a\|^2 \right) \\ &= \arg\min_{a\in\mathbb{R}^D} \left(\left\|y - X^\top a\right\|^2 + \lambda \|a\|^2 \right), \end{aligned}$$

where the second term $\lambda\|a\|^2$ is chosen to avoid problems in high dimensional spaces (see below). Expanding out the above term, we get

$$\begin{aligned} \left\|y - X^\top a\right\|^2 + \lambda\|a\|^2 &= y^\top y - 2y^\top X a + a^\top X X^\top a + \lambda a^\top a \\ &= y^\top y - 2y^\top X^\top a + a^\top \left(XX^\top + \lambda I\right) a = (*) \end{aligned}$$

Define $b = \left(XX^\top + \lambda I\right)^{1/2} a$, where the square root is well defined since the matrix is positive definite (it may be that $XX^\top$ is not invertible, for instance, when $D > n$, so adding λI ensures we can substitute $a = \left(XX^\top + \lambda I\right)^{-1/2} b$). Then

$$\begin{aligned} (*) &= y^\top y - 2y^\top X^\top \left(XX^\top + \lambda I\right)^{-1/2} b + b^\top b \\ &= y^\top y + \left\| \left(XX^\top + \lambda I\right)^{-1/2} Xy - b \right\|^2 - \left\| y^\top X^\top \left(XX^\top + \lambda I\right)^{-1/2} \right\|^2, \end{aligned}$$

where we complete the square. This is minimized when

$$\begin{aligned} b^* &= \left(XX^\top + \lambda I\right)^{-1/2} Xy \quad \text{or} \\ a^* &= \left(XX^\top + \lambda I\right)^{-1} Xy, \end{aligned}$$

which is the classic regularized least squares solution.[10]

[10] This proof differs from the usual derivation, which we give here for ease of reference (this is not the approach we use, since we are later going to extend our reasoning to feature spaces: derivatives in feature space also exist when the space is infinite dimensional, however for the purposes of ridge regression they can be avoided). We use [5, eqs. (61) and (73)]

$$\frac{\partial a^\top U a}{\partial a} = (U + U^\top)a, \qquad \frac{\partial v^\top a}{\partial a} = \frac{\partial a^\top v}{\partial a} = v,$$

Taking the derivative of the expanded expression $(*)$ and setting to zero,

$$\begin{aligned} \frac{\partial}{\partial a}\left(\left\|y - X^\top a\right\|^2 + \lambda\|a\|^2 \right) &= -2Xy + 2\left(XX^\top + \lambda I\right) a = 0, \\ a &= \left(XX^\top + \lambda I\right)^{-1} Xy. \end{aligned}$$

7.1.2 Finite dimensional case: more informative expression

We may rewrite this expression in a way that is more informative (and more easily kernelized). Assume without loss of generality that $D > n$ (this will be useful when we move to feature spaces, where D can be very large or even infinite). We can perform an SVD on X, i.e.

$$X = USV^\top,$$

where

$$U = \left[\begin{array}{ccc} u_1 & \dots & u_D \end{array} \right] \quad S = \left[\begin{array}{cc} \tilde{S} & 0 \\ 0 & 0 \end{array} \right] \quad V = \left[\begin{array}{cc} \tilde{V} & 0 \end{array} \right].$$

Here U is $D \times D$ and $U^\top U = UU^\top = I_D$ (the subscript denotes the size of the unit matrix), S is $D \times D$, where the top left diagonal $\tilde{S}$ has n non-zero entries, and V is $n \times D$, where only the first n columns are non-zero, and $\tilde{V}^\top \tilde{V} = \tilde{V}\tilde{V}^\top = I_n$.[11] Then

$$\begin{aligned} a^* &= \left(XX^\top + \lambda I_D\right)^{-1} Xy \\ &= \left(US^2U^\top + \lambda I_D\right)^{-1} USV^\top y \\ &= U\left(S^2 + \lambda I_D\right)^{-1} U^\top USV^\top y \\ &= U\left(S^2 + \lambda I_D\right)^{-1} SV^\top y \\ &= US\left(S^2 + \lambda I_D\right)^{-1} V^\top y \\ &= U\underbrace{SV^\top V}_{\text{(a)}}\left(S^2 + \lambda I_D\right)^{-1} V^\top y \\ &\underset{\text{(b)}}{=} X(X^\top X + \lambda I_n)^{-1} y \end{aligned} \tag{7.1}$$

Step (a) is allowed since both S and $V^\top V$ are non-zero in the same sized top-left block, and $V^\top V$ is just the unit matrix in that block. Step (b) occurs as follows

$$\begin{aligned} V\left(S^2 + \lambda I_D\right)^{-1} V^\top &= \left[\begin{array}{cc} \tilde{V} & 0 \end{array} \right] \left[\begin{array}{cc} \left(\tilde{S}^2 + \lambda I_n\right)^{-1} & 0 \\ 0 & (\lambda I_{D-n})^{-1} \end{array} \right] \left[\begin{array}{c} \tilde{V}^\top \\ 0 \end{array} \right] \\ &= \tilde{V}\left(\tilde{S}^2 + \lambda I_n\right)^{-1} \tilde{V}^\top \\ &= \left(X^\top X + \lambda I_n\right)^{-1}. \end{aligned}$$

What's interesting about this result is that $a^* = \sum_{i=1}^n \alpha_i^* x_i$, i.e. a is a weighted sum of columns of X. Again, one can obtain this result straightfor-

[11] Another more economical way to write the SVD would be

$$X = U\left[\begin{array}{c} \tilde{S} \\ 0 \end{array} \right] \tilde{V}^\top,$$

but as we'll see, we will need the larger form.

wardly by applying established linear algebra results: the proof here is informative, however, since we are explicitly demonstrating the steps we take, and hence we can be assured the same steps will still work even if D is infinite.[12]

7.1.3 Feature space case

We now consider the case where we use features $\phi(x_i)$ in the place of x_i:

$$a^* \quad = \quad \arg\min_{a\in\mathcal{H}} \left(\sum_{i=1}^{n} \left(y_i - \langle a, \phi(x_i)\rangle_{\mathcal{H}}\right)^2 + \lambda \|a\|_{\mathcal{H}}^2 \right).$$

We could consider a number of options: e.g. the polynomial features or sinusoidal features

$$\phi_p(x) = \begin{bmatrix} x \\ x^2 \\ \vdots \\ x^\ell \end{bmatrix} \qquad \phi_s(x) = \begin{bmatrix} \sin x \\ \cos x \\ \sin 2x \\ \vdots \\ \cos \ell x \end{bmatrix}$$

In these cases, a is a vector of length ℓ giving weight to each of these features so as to find the mapping between x and y. We can also consider feature vectors of *infinite* length, as we discussed before.

It is straightforward to obtain the feature space solution of the ridge regression equation in the light of the previous section: with some cumbersome notation, write[13]

$$X = \begin{bmatrix} \phi(x_1) & \dots & \phi(x_n) \end{bmatrix}.$$

All of the steps that led us to (7.1) then follow, where in particular

$$XX^\top = \sum_{i=1}^{n} \phi(x_i) \otimes \phi(x_i)$$

(using the notation (6.1) we introduced from kernel PCA), and

$$(X^\top X)_{ij} = \langle \phi(x_i), \phi(x_j)\rangle_{\mathcal{H}} = k(x_i, x_j).$$

[12] We could apply one of the many variants of the Woodbury identity [5, eqs. (147)]. If P and R are positive definite, then

$$\left(P^{-1} + B^\top R^{-1} B\right) B^\top R^{-1} = PB^\top \left(BPB^\top + R\right)^{-1}. \tag{7.2}$$

Setting $P = \lambda^{-1} I$, $B = X^\top$, and $R = I$, we get

$$\begin{aligned} a = \left(XX^\top + \lambda I\right)^{-1} Xy &= \lambda^{-1} X \left(\lambda^{-1} X^\top X + I\right) y \\ &= X \left(X^\top X + \lambda I\right) y. \end{aligned}$$

[13] For infinite dimensional feature spaces, the operator X still has a singular value decomposition - this will be covered later in the course.

Making these replacements, we get

$$\begin{aligned} a^* &= X(K+\lambda I_n)^{-1}y \\ &= \sum_{i=1}^{n} \alpha_i^* \phi(x_i) \qquad \alpha^* = (K+\lambda I_n)^{-1}y. \end{aligned}$$

Note that the proof becomes much easier if we *begin* with the knowledge that a is a linear combination of feature space mappings of points,[14]

$$a = \sum_{i=1}^{n} \alpha_i \phi(x_i).$$

Then

$$\begin{aligned} \sum_{i=1}^{n} \left(y_i - \langle a, \phi(x_i)\rangle_{\mathcal{H}}\right)^2 + \lambda \|a\|_{\mathcal{H}}^2 &= \|y - K\alpha\|^2 + \lambda \alpha^\top K \alpha \\ &= y^\top y - 2y^\top K\alpha + \alpha^\top \left(K^2 + \lambda K\right)\alpha \end{aligned}$$

Differentiating wrt α and setting this to zero, we get

$$\alpha^* = (K+\lambda I_n)^{-1}y$$

as before.

7.2 Link of RKHS norm with smoothness of regression function

What does $\|f\|_{\mathcal{H}}^2$ have to do with smoothing? We illustrate this with two examples, taken from earlier in the notes. Recall from Section 4.1.3 that functions in the Gaussian RKHS take the form

$$f(x) = \sum_{\ell=1}^{\infty} \underbrace{f_\ell \sqrt{\lambda_\ell}}_{\hat{f}_\ell} e_\ell(x), \qquad \|f\|_{\mathcal{H}}^2 = \sum_{\ell=1}^{\infty} \frac{\hat{f}_\ell^2}{\lambda_\ell},$$

where the eigenfunctions $e_\ell(x)$ were illustrated in Figure (4.4), and satisfy the orthonormality condition (4.8) for the measure (4.9). The constraint $\|f\|_{\mathcal{H}}^2 < \infty$ means that the $\hat{f}_\ell^2$ must decay faster than λ_ℓ with increasing ℓ. In other words, basis functions $e_\ell(x)$ with larger ℓ are given less weight: these are the non-smooth functions.

The same effect can be seen if we use the feature space in Section 4.1.2. Recall that functions on the periodic domain $[-\pi, \pi]$ have the representation

$$f(x) = \sum_{\ell=-\infty}^{\infty} \hat{f}_\ell \exp(\imath \ell x).$$

[14] This is a specific instance of the representer theorem, which we will encounter later.

Again,

$$\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} = \sum_{l=-\infty}^{\infty} \frac{\left|\hat{f}_\ell\right|^2}{\hat{k}_\ell}.$$

This means $\left|\hat{f}_\ell\right|^2$ must decay faster than $\hat{k}_\ell$ for the norm to be finite.[15] This serves to suppress the terms $\exp(\imath\ell x)$ for large ℓ, which are the non-smooth terms.

7.3 Model selection

In kernel ridge regression, we have control over two things: the kernel we use, and the weight λ. The kernel used controls the smoothness of the class of functions we consider. The weight λ controls the tradeoff between function smoothness and fitting error. We now look at these properties more closely, doing kernel ridge regression with a Gaussian kernel,

$$k(x, y) = \exp\left(\frac{-\|x - y\|^2}{\sigma}\right).$$

From Figure 7.1, we see that too large a λ prioritises smoothness over getting a small prediction error on the points, resulting in a very smooth function which barely follows the shape of the underlying data - in other words, we are **underfitting**. Too small a λ gives too much priority to fitting small fluctuations in the data due to noise, at the expense of smoothness: in this case, we are **overfitting**. Finally, an apparently good choice is $\lambda = 0.1$, where the regression curve fits the underlying trend without being overly influenced by noise.

Figure 7.2 shows how the kernel width σ affects the fit of ridge regression. Too large a σ results in underfitting: the regression function is too smooth. Too small a σ results in overfitting. There is some overlap in the effect on prediction quality of σ and λ.

How do we choose λ and σ, and how do we evaluate the resulting performance of our learning algorithm? One commonly used approach is to combine m-fold cross-validation and a held-out test set. See Algorithm 1.

7.4 A Bayesian interpretation

The Bayesian interpretation of ridge regression can be found in [6, Chapter 2]. Advantage: also get uncertainty estimate in the prediction.

8 Acknowledgements

Thanks to Gergo Bohner, Peter Dayan, Agnieszka Grabska-Barwinska, Wittawat Jitkrittum, Peter Latham, Arian Maleki, Kirsty McNaughton, Sam Pat-

[15] The rate of decay of $\hat{k}_l$ will depend on the properties of the kernel. Some relevant results may be found at `http://en.wikipedia.org/wiki/Convergence_of_Fourier_series`

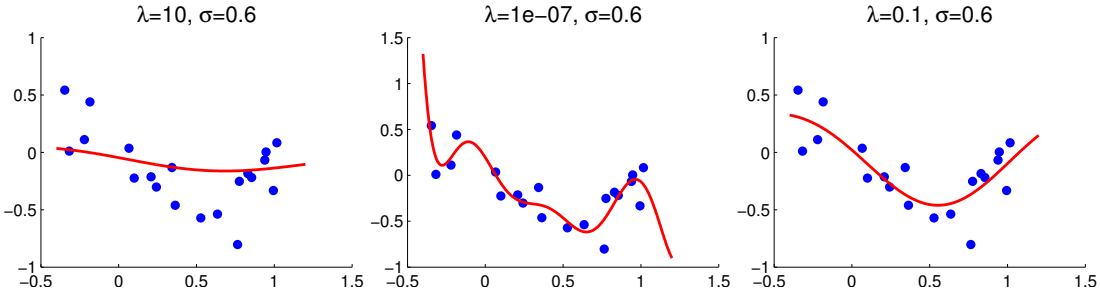

Figure 7.1: Effect of choice of λ on the fit of ridge regression.

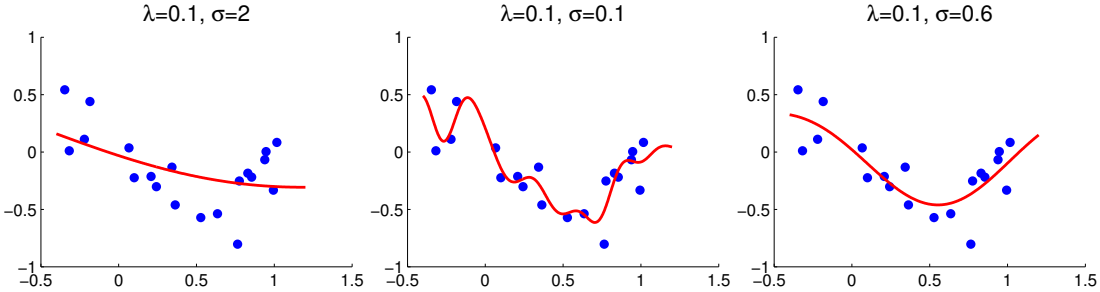

Figure 7.2: Effect of choice of σ on the fit of ridge regression.

Algorithm 1 m-fold cross validation and held-out test set.

1. Start with a dataset $Z := X, Y$, where X is a matrix with n columns, corresponding to the n training points, and Y is a vector having n rows. We split this into a training set of size n_{tr} and a test set of size $n_{\text{te}} = 1 - n_{\text{tr}}$.
2. Break the trainining set into m equally sized chunks, each of size $n_{\text{val}} = n_{\text{tr}}/m$. Call these $X_{\text{val},i}, Y_{\text{val},i}$ for $i \in \{1, \ldots, m\}$
3. For each λ, σ pair
   (a) For each $X_{\text{val},i}, Y_{\text{val},i}$
      i. Train the ridge regression on the remaining trainining set data $X_{\text{tr}} \setminus X_{\text{val},i}$ and $Y_{\text{tr}} \setminus Y_{\text{val},i}$,
      ii. Evaluate its error on the validation data $X_{\text{val},i}, Y_{\text{val},i}$
   (b) Average the errors on the validation sets to get the average validation error for λ, σ.
4. Choose λ^*, σ^* with the lowest average validation error
5. Measure the performance on the test set $X_{\text{te}}, Y_{\text{te}}$.

Table 1: Fourier series relations in 1-D.

Description of rule	Input space	Frequency space
Shift	$f(x - x_0)$	$\tilde{f}_l \exp(-\imath l (2\pi/T) x_0)$
Input real	$f^*(x) = f(x)$	$\tilde{f}_l = -\tilde{f}_l^*$
Input even, real	$f^*(x) = f(x)$, $f(-x) = f(x)$	$\tilde{f}_l = -\tilde{f}_l^*$
Scaling	$f(ax)$	T changes accordingly
Differentiation	$\frac{d}{dx} f(x)$	$\imath l (2\pi/T) \tilde{f}_l$
Parseval's theorem	$\int_{-T/2}^{T/2} f(x) g^*(x) dx$	$\sum_{k=-\infty}^{\infty} \tilde{f}_l \tilde{g}_l^*$

terson, and Dino Sejdinovic for providing feedback on the notes, and correcting errors.

A The Fourier series on $\left[-\frac{T}{2}, \frac{T}{2}\right]$ with periodic boundary conditions

We consider the case in which $f(x)$ is periodic with period T, so that we need only specify $f(x) : \left[-\frac{T}{2}, \frac{T}{2}\right] \to \mathbb{R}$. In this case, we obtain the Fourier series expansion

$$\tilde{f}_l = \frac{1}{T} \int_{-T/2}^{T/2} f(x) \exp\left(-\imath l x \frac{2\pi}{T}\right) dx = \frac{1}{T} \tilde{f}\left(l \frac{2\pi}{T}\right), \tag{A.1}$$

such that

$$f(x) = \sum_{l=-\infty}^{\infty} \tilde{f}_l \exp\left(\imath l x \frac{2\pi}{T}\right). \tag{A.2}$$

Thus the $\tilde{f}_l$ represent the Fourier transform at frequencies $\omega = l\frac{2\pi}{T}$, scaled by T^{-1}. We document a number of useful Fourier series relations in Table 1.

References

[1] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004.

[2] Felipe Cucker and Steve Smale. Best choices for regularization parameters in learning theory: On the bias–variance problem. *Foundations of Computational Mathematics*, 2(4):413–428, October 2002.

[3] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample problem. In *Advances in Neural Information Processing Systems 15*, pages 513–520, Cambridge, MA, 2007. MIT Press.

[4] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, February 2002.

[5] K. B. Petersen and M. S. Pedersen. The matrix cookbook, 2008. Version 20081110.

[6] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[7] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10:1299–1319, 1998.

[8] Bernhard Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[9] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

[10] B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.

[11] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Information Science and Statistics. Springer, 2008.

[12] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge, UK, 2005.