3

Properties of kernels

As we have seen in Chapter 2, the use of kernel functions provides a powerful and principled way of detecting nonlinear relations using well-understood linear algorithms in an appropriate feature space. The approach decouples the design of the algorithm from the specification of the feature space. This inherent modularity not only increases the flexibility of the approach, it also makes both the learning algorithms and the kernel design more amenable to formal analysis. Regardless of which pattern analysis algorithm is being used, the theoretical properties of a given kernel remain the same. It is the purpose of this chapter to introduce the properties that characterise kernel functions.

We present the fundamental properties of kernels, thus formalising the intuitive concepts introduced in Chapter 2. We provide a characterization of kernel functions, derive their properties, and discuss methods for designing them. We will also discuss the role of prior knowledge in kernel-based learning machines, showing that a universal machine is not possible, and that kernels must be chosen for the problem at hand with a view to capturing our prior belief of the relatedness of different examples. We also give a framework for quantifying the match between a kernel and a learning task.

Given a kernel and a training set, we can form the matrix known as the kernel, or Gram matrix: the matrix containing the evaluation of the kernel function on all pairs of data points. This matrix acts as an information bottleneck, as all the information available to a kernel algorithm, be it about the distribution, the model or the noise, must be extracted from that matrix. It is therefore not surprising that the kernel matrix plays a central role in the development of this chapter.

3.1 Inner products and positive semi-definite matrices

Chapter 2 showed how data can be embedded in a high-dimensional feature space where linear pattern analysis can be performed giving rise to non-linear pattern analysis in the input space. The use of kernels enables this technique to be applied without paying the computational penalty implicit in the number of dimensions, since it is possible to evaluate the inner product between the images of two inputs in a feature space without explicitly computing their coordinates.

These observations imply that we can apply pattern analysis algorithms to the image of the training data in the feature space through indirect evaluation of the inner products. As defined in Chapter 2, a function that returns the inner product between the images of two inputs in some feature space is known as a *kernel function*.

This section reviews the notion and properties of inner products that will play a central role in this book. We will relate them to the positive semi-definiteness of the Gram matrix and general properties of positive semi-definite symmetric functions.

3.1.1 Hilbert spaces

First we recall what is meant by a linear function. Given a vector space X over the reals, a function

$$f : X \longrightarrow \mathbb{R}$$

is linear if $f(\alpha\mathbf{x}) = \alpha f(\mathbf{x})$ and $f(\mathbf{x}+\mathbf{z}) = f(\mathbf{x}) + f(\mathbf{z})$ for all $\mathbf{x}, \mathbf{z} \in X$ and $\alpha \in \mathbb{R}$.

Inner product space A vector space X over the reals $\mathbb{R}$ is an *inner product space* if there exists a real-valued symmetric bilinear (linear in each argument) map $\langle \cdot, \cdot \rangle$, that satisfies

$$\langle \mathbf{x}, \mathbf{x} \rangle \geq 0.$$

The bilinear map is known as the inner, dot or scalar product. Furthermore we will say the inner product is *strict* if

$$\langle \mathbf{x}, \mathbf{x} \rangle = 0 \text{ if and only if } \mathbf{x} = \mathbf{0}.$$

Given a strict inner product space we can define a norm on the space X by

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}.$$

The associated metric or distance between two vectors $\mathbf{x}$ and $\mathbf{z}$ is defined as $d(\mathbf{x},\mathbf{z}) = \|\mathbf{x}-\mathbf{z}\|_2$. For the vector space $\mathbb{R}^n$ the standard inner product is given by

$$\langle \mathbf{x},\mathbf{z}\rangle = \sum_{i=1}^{n} x_i z_i.$$

Furthermore, if the inner product is not strict, those points $\mathbf{x}$ for which $\|\mathbf{x}\| = 0$ form a linear subspace since Proposition 3.5 below shows $\langle \mathbf{x},\mathbf{y}\rangle^2 \leq \|\mathbf{x}\|^2 \|\mathbf{y}\|^2 = 0$, and hence if also $\|\mathbf{z}\| = 0$ we have for all $a, b \in \mathbb{R}$

$$\|a\mathbf{x}+b\mathbf{z}\|^2 = \langle a\mathbf{x}+b\mathbf{z}, a\mathbf{x}+b\mathbf{z}\rangle = a^2 \|\mathbf{x}\|^2 + 2ab\langle \mathbf{x},\mathbf{z}\rangle + b^2 \|\mathbf{z}\|^2 = 0.$$

This means that we can always convert a non-strict inner product to a strict one by taking the quotient space with respect to this subspace.

A vector space with a metric is known as a *metric space*, so that a strict inner product space is also a metric space. A metric space has a derived topology with a sub-basis given by the set of open balls.

An inner product space is sometimes referred to as a Hilbert space, though most researchers require the additional properties of completeness and separability, as well as sometimes requiring that the dimension be infinite. We give a formal definition.

Definition 3.1 A *Hilbert Space* $\mathcal{F}$ is an inner product space with the additional properties that it is *separable* and *complete*. Completeness refers to the property that every Cauchy sequence $\{h_n\}_{n\geq 1}$ of elements of $\mathcal{F}$ converges to a element $h \in \mathcal{F}$, where a Cauchy sequence is one satisfying the property that

$$\sup_{m>n} \|h_n - h_m\| \to 0, \text{ as } n \to \infty.$$

A space $\mathcal{F}$ is separable if for any $\epsilon > 0$ there is a finite set of elements $h_1, \ldots, h_N$ of $\mathcal{F}$ such that for all $h \in \mathcal{F}$

$$\min_i \|h_i - h\| < \epsilon.$$

■

Example 3.2 Let X be the set of all countable sequences of real numbers $\mathbf{x} = (x_1, x_2, \ldots, x_n, \ldots)$, such that the sum

$$\sum_{i=1}^{\infty} x_i^2 < \infty,$$

with the inner product between two sequences $\mathbf{x}$ and $\mathbf{y}$ defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{\infty} x_i y_i.$$

This is the space known as L_2.

The reason for the importance of the properties of completeness and separability is that together they ensure that Hilbert spaces are either isomorphic to $\mathbb{R}^n$ for some finite n or to the space L_2 introduced in Example 3.2. For our purposes we therefore require that the feature space be a complete, separable inner product space, as this will imply that it can be given a coordinate system. Since we will be using the dual representation there will, however, be no need to actually construct the feature vectors.

This fact may seem strange at first since we are learning a linear function represented by a weight vector in this space. But as discussed in Chapter 2 the weight vector is a linear combination of the feature vectors of the training points. Generally, all elements of a Hilbert space are also linear functions in that space via the inner product. For a point $\mathbf{z}$ the corresponding function $f_{\mathbf{z}}$ is given by

$$f_{\mathbf{z}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{z} \rangle.$$

Finding the weight vector is therefore equivalent to identifying an appropriate element of the feature space.

We give two more examples of inner product spaces.

Example 3.3 Let $X = \mathbb{R}^n$, $\mathbf{x} = (x_1, \ldots, x_n)'$, $\mathbf{z} = (z_1, \ldots, z_n)'$. Let λ_i be fixed positive numbers, for $i = 1, \ldots, n$. The following defines a valid inner product on X

$$\langle \mathbf{x}, \mathbf{z} \rangle = \sum_{i=1}^{n} \lambda_i x_i z_i = \mathbf{x}' \boldsymbol{\Lambda} \mathbf{z},$$

where $\boldsymbol{\Lambda}$ is the $n \times n$ diagonal matrix with entries $\boldsymbol{\Lambda}_{ii} = \lambda_i$.

Example 3.4 Let $\mathcal{F} = L_2(X)$ be the vector space of square integrable functions on a compact subset X of $\mathbb{R}^n$ with the obvious definitions of addition and scalar multiplication, that is

$$L_2(X) = \left\{ f : \int_X f(x)^2 \, dx < \infty \right\}.$$

For f, $g \in X$, define the inner product by

$$\langle f, g \rangle = \int_X f(x)g(x)dx.$$

Proposition 3.5 (Cauchy–Schwarz inequality) *In an inner product space*

$$\langle \mathbf{x}, \mathbf{z} \rangle^2 \leq \|\mathbf{x}\|^2 \|\mathbf{z}\|^2 .$$

and the equality sign holds in a strict inner product space if and only if $\mathbf{x}$ *and* $\mathbf{z}$ *are rescalings of the same vector.*

Proof Consider an abitrary $\epsilon > 0$ and the following norm

$$\begin{aligned} 0 &\leq \|(\|\mathbf{z}\| + \epsilon)\mathbf{x} \pm \mathbf{z}(\|\mathbf{x}\| + \epsilon)\|^2 \\ &= \langle (\|\mathbf{z}\| + \epsilon)\mathbf{x} \pm \mathbf{z}(\|\mathbf{x}\| + \epsilon), (\|\mathbf{z}\| + \epsilon)\mathbf{x} \pm \mathbf{z}(\|\mathbf{x}\| + \epsilon)\rangle \\ &= (\|\mathbf{z}\| + \epsilon)^2 \|\mathbf{x}\|^2 + \|\mathbf{z}\|^2 (\|\mathbf{x}\| + \epsilon)^2 \pm 2\langle (\|\mathbf{z}\| + \epsilon)\mathbf{x}, \mathbf{z}(\|\mathbf{x}\| + \epsilon)\rangle \\ &\leq 2(\|\mathbf{z}\| + \epsilon)^2 (\|\mathbf{x}\| + \epsilon)^2 \pm 2(\|\mathbf{z}\| + \epsilon)(\|\mathbf{x}\| + \epsilon)\langle \mathbf{x}, \mathbf{z}\rangle , \end{aligned}$$

implying that

$$\mp \langle \mathbf{x}, \mathbf{z} \rangle \leq (\|\mathbf{x}\| + \epsilon)(\|\mathbf{z}\| + \epsilon).$$

Letting $\epsilon \longrightarrow 0$ gives the first result. In a strict inner product space equality implies

$$\mathbf{x}\|\mathbf{z}\| \pm \mathbf{z}\|\mathbf{x}\| = \mathbf{0},$$

making $\mathbf{x}$ and $\mathbf{z}$ rescalings as required. □

Angles, distances and dimensionality The *angle* θ between two vectors $\mathbf{x}$ and $\mathbf{z}$ of a strict inner product space is defined by

$$\cos\theta = \frac{\langle \mathbf{x}, \mathbf{z} \rangle}{\|\mathbf{x}\| \|\mathbf{z}\|}$$

If $\theta = 0$ the cosine is 1 and $\langle \mathbf{x}, \mathbf{z} \rangle = \|\mathbf{x}\| \|\mathbf{z}\|$, and $\mathbf{x}$ and $\mathbf{z}$ are said to be *parallel.* If $\theta = \frac{\pi}{2}$, the cosine is 0, $\langle \mathbf{x}, \mathbf{z} \rangle = 0$ and the vectors are said to be *orthogonal.*

A set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ of vectors from X is called *orthonormal* if $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta satisfying $\delta_{ij} = 1$ if $i = j$, and 0 otherwise. For an orthonormal set S, and a vector $\mathbf{z} \in X$, the expression

$$\sum_{i=1}^{\ell} \langle \mathbf{x}_i, \mathbf{z} \rangle \mathbf{x}_i$$

is said to be a *Fourier series* for $\mathbf{z}$. If the Fourier series for $\mathbf{z}$ equals $\mathbf{z}$ for all $\mathbf{z}$, then the set S is also a basis. Since a Hilbert space is either equivalent to $\mathbb{R}^n$ or to L_2, it will always be possible to find an orthonormal basis, indeed this basis can be used to define the isomorphism with either $\mathbb{R}^n$ or L_2.

The *rank* of a general $n \times m$ matrix $\mathbf{X}$ is the dimension of the space spanned by its columns also known as the column space. Hence, the rank of $\mathbf{X}$ is the smallest r for which we can express

$$\mathbf{X} = \mathbf{RS},$$

where $\mathbf{R}$ is an $n \times r$ matrix whose linearly independent columns form a basis for the column space of $\mathbf{X}$, while the columns of the $r \times m$ matrix $\mathbf{S}$ express the columns of $\mathbf{X}$ in that basis. Note that we have

$$\mathbf{X}' = \mathbf{S}'\mathbf{R}',$$

and since $\mathbf{S}'$ is $m \times r$, the rank of $\mathbf{X}'$ is less than or equal to the rank of $\mathbf{X}$. By symmetry the two ranks are equal, implying that the dimension of the space spanned by the rows of $\mathbf{X}$ is also equal to its rank.

An $n \times m$ matrix is *full rank* if its rank is equal to $\min(n, m)$.

3.1.2 Gram matrix

Given a set of vectors, $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ the *Gram matrix* is defined as the $\ell \times \ell$ matrix $\mathbf{G}$ whose entries are $\mathbf{G}_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. If we are using a kernel function κ to evaluate the inner products in a feature space with feature map $\boldsymbol{\phi}$, the associated Gram matrix has entries

$$\mathbf{G}_{ij} = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle = \kappa(\mathbf{x}_i, \mathbf{x}_j).$$

In this case the matrix is often referred to as the *kernel matrix.* We will use a standard notation for displaying kernel matrices as:

$\mathbf{K}$	1	2	$\cdots$	ℓ
1	$\kappa(\mathbf{x}_1, \mathbf{x}_1)$	$\kappa(\mathbf{x}_1, \mathbf{x}_2)$	$\cdots$	$\kappa(\mathbf{x}_1, \mathbf{x}_\ell)$
2	$\kappa(\mathbf{x}_2, \mathbf{x}_1)$	$\kappa(\mathbf{x}_2, \mathbf{x}_2)$	$\cdots$	$\kappa(\mathbf{x}_2, \mathbf{x}_\ell)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
ℓ	$\kappa(\mathbf{x}_\ell, \mathbf{x}_1)$	$\kappa(\mathbf{x}_\ell, \mathbf{x}_2)$	$\cdots$	$\kappa(\mathbf{x}_\ell, \mathbf{x}_\ell)$

where the symbol $\mathbf{K}$ in the top left corner indicates that the table represents a kernel matrix – see the Appendix B for a summary of notations.

In Chapter 2, the Gram matrix has already been shown to play an important role in the dual form of some learning algorithms. The matrix is symmetric since $\mathbf{G}_{ij} = \mathbf{G}_{ji}$, that is $\mathbf{G}' = \mathbf{G}$. Furthermore, it contains all the information needed to compute the pairwise distances within the data set as shown above. In the Gram matrix there is of course some information that is lost when compared with the original set of vectors. For example the matrix loses information about the orientation of the original data set with respect to the origin, since the matrix of inner products is invariant to rotations about the origin. More importantly the representation loses information about any alignment between the points and the axes. This again follows from the fact that the Gram matrix is rotationally invariant in the sense that any rotation of the coordinate system will leave the matrix of inner products unchanged.

If we consider the dual form of the ridge regression algorithm described in Chapter 2, we will see that the only information received by the algorithm about the training set comes from the Gram or kernel matrix and the associated output values. This observation will characterise all of the kernel algorithms considered in this book. In other words all the information the pattern analysis algorithms can glean about the training data and chosen feature space is contained in the kernel matrix together with any labelling information.

In this sense we can view the matrix as an *information bottleneck* that must transmit enough information about the data for the algorithm to be able to perform its task. This view also reinforces the view that the kernel matrix is the central data type of all kernel-based algorithms. It is therefore natural to study the properties of these matrices, how they are created, how they can be adapted, and how well they are matched to the task being addressed.

Singular matrices and eigenvalues A matrix $\mathbf{A}$ is singular if there is a non-trivial linear combination of the columns of $\mathbf{A}$ that equals the vector $\mathbf{0}$. If we put the coefficients x_i of this combination into a (non-zero) vector $\mathbf{x}$, we have that

$$\mathbf{A}\mathbf{x} = \mathbf{0} = 0\mathbf{x}.$$

If an $n \times n$ matrix $\mathbf{A}$ is non-singular the columns are linearly independent and hence space a space of dimension n. Hence, we can find vectors $\mathbf{u}_i$ such that

$$\mathbf{A}\mathbf{u}_i = \mathbf{e}_i,$$

where $\mathbf{e}_i$ is the ith unit vector. Forming a matrix $\mathbf{U}$ with ith column equal to $\mathbf{u}_i$ we have

$$\mathbf{AU} = \mathbf{I}$$

the identity matrix. Hence, $\mathbf{U} = \mathbf{A}^{-1}$ is the multiplicative inverse of $\mathbf{A}$.

Given a matrix $\mathbf{A}$, the real number λ and the vector $\mathbf{x}$ are an *eigenvalue* and corresponding *eigenvector* of $\mathbf{A}$ if

$$\mathbf{Ax} = \lambda\mathbf{x}.$$

It follows from the observation above about singular matrices that 0 is an eigenvalue of a matrix if and only if it is singular. Note that for an eigenvalue, eigenvector pair $\mathbf{x}$, λ, the quotient obeys

$$\frac{\mathbf{x}'\mathbf{Ax}}{\mathbf{x}'\mathbf{x}} = \lambda\frac{\mathbf{x}'\mathbf{x}}{\mathbf{x}'\mathbf{x}} = \lambda. \tag{3.1}$$

The quotient of equation (3.1) is known as the *Rayleigh quotient* and will form an important tool in the development of the algorithms of Chapter 6. Consider the optimisation problem

$$\max_{\mathbf{v}} \frac{\mathbf{v}'\mathbf{Av}}{\mathbf{v}'\mathbf{v}} \tag{3.2}$$

and observe that the solution is invariant to rescaling. We can therefore impose the constraint that $\|\mathbf{v}\| = 1$ and solve using a Lagrange multiplier. We obtain for a symmetric matrix $\mathbf{A}$ the optimisation

$$\max_{\mathbf{v}} \left(\mathbf{v}'\mathbf{Av} - \lambda\left(\mathbf{v}'\mathbf{v} - 1\right)\right),$$

which on setting the derivatives with respect to $\mathbf{v}$ equal to zero gives

$$\mathbf{Av} = \lambda\mathbf{v}.$$

We will always assume that an eigenvector is normalised.

Hence, the eigenvector of the largest eigenvalue is the solution of the optimisation (3.2) with the corresponding eigenvalue giving the value of the maximum. Since we are seeking the maximum over a compact set we are guaranteed a solution. A similar approach can also yield the minimum eigenvalue.

The *spectral norm* or 2*-norm* of a matrix $\mathbf{A}$ is defined as

$$\max_{\mathbf{v}} \frac{\|\mathbf{Av}\|}{\|\mathbf{v}\|} = \sqrt{\max_{\mathbf{v}} \frac{\mathbf{v}'\mathbf{A}'\mathbf{Av}}{\mathbf{v}'\mathbf{v}}}. \tag{3.3}$$

Symmetric matrices and eigenvalues We say a square matrix $\mathbf{A}$ is *symmetric* if $\mathbf{A}' = \mathbf{A}$, that is the (i, j) entry equals the (j, i) entry for all i and j. A matrix is *diagonal* if its off-diagonal entries are all 0. A square matrix is *upper* (*lower*) *triangular* if its above (below) diagonal elements are all zero.

For symmetric matrices we have that eigenvectors corresponding to distinct eigenvalues are orthogonal, since if μ, $\mathbf{z}$ is a second eigenvalue, eigenvector pair with $\mu \neq \lambda$, we have that

$$\begin{aligned}\lambda \langle \mathbf{x}, \mathbf{z} \rangle &= \langle \mathbf{A}\mathbf{x}, \mathbf{z} \rangle \\ &= (\mathbf{A}\mathbf{x})' \mathbf{z} \\ &= \mathbf{x}'\mathbf{A}'\mathbf{z} \\ &= \mathbf{x}'\mathbf{A}\mathbf{z} \\ &= \mu \langle \mathbf{x}, \mathbf{z} \rangle ,\end{aligned}$$

implying that $\langle \mathbf{x}, \mathbf{z} \rangle = \mathbf{x}'\mathbf{z} = 0$. This means that if $\mathbf{A}$ is an $n \times n$ symmetric matrix, it can have at most n distinct eigenvalues. Given an eigenvector–eigenvalue pair $\mathbf{x}$, λ of the matrix $\mathbf{A}$, the transformation

$$\mathbf{A} \longmapsto \tilde{\mathbf{A}} = \mathbf{A} - \lambda \mathbf{x}\mathbf{x}',$$

is known as *deflation*. Note that since $\mathbf{x}$ is normalised

$$\tilde{\mathbf{A}}\mathbf{x} = \mathbf{A}\mathbf{x} - \lambda \mathbf{x}\mathbf{x}'\mathbf{x} = \mathbf{0},$$

so that deflation leaves $\mathbf{x}$ an eigenvector but reduces the corresponding eigenvalue to zero. Since eigenvectors corresponding to distinct eigenvalues are orthogonal the remaining eigenvalues of $\mathbf{A}$ remain unchanged. By repeatedly finding the eigenvector corresponding to the largest positive (or smallest negative) eigenvalue and then deflating, we can always find an orthonormal set of n eigenvectors, where eigenvectors corresponding to an eigenvalue of 0 are added by extending the set of eigenvectors obtained by deflation to an orthonormal basis. If we form a matrix $\mathbf{V}$ with the (orthonormal) eigenvectors as columns and a diagonal matrix $\mathbf{\Lambda}$ with $\mathbf{\Lambda}_{ii} = \lambda_i$, $i = 1, \ldots, n$, the corresponding eigenvalues, we have $\mathbf{V}\mathbf{V}' = \mathbf{V}'\mathbf{V} = \mathbf{I}$, the identity matrix and

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}.$$

This is often referred to as the *eigen-decomposition* of $\mathbf{A}$, while the set of eigenvalues $\lambda(\mathbf{A})$ are known as its *spectrum*. We generally assume that the eigenvalues appear in order of decreasing value

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n.$$

Note that a matrix $\mathbf{V}$ with the property $\mathbf{V}\mathbf{V}' = \mathbf{V}'\mathbf{V} = \mathbf{I}$ is known as an *orthonormal or unitary matrix.*

The *principal minors* of a matrix are the submatrices obtained by selecting a subset of the rows and the same subset of columns. The corresponding minor contains the elements that lie on the intersections of the chosen rows and columns.

If the symmetric matrix $\mathbf{A}$ has k non-zero eigenvalues then we can express the eigen-decomposition as

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}' = \mathbf{V}_k\mathbf{\Lambda}_k\mathbf{V}_k',$$

where $\mathbf{V}_k$ and $\mathbf{\Lambda}_k$ are the matrices containing the k columns of $\mathbf{V}$ and the principal minor of $\mathbf{\Lambda}$ corresponding to non-zero eigenvalues. Hence, $\mathbf{A}$ has rank at most k. Given any vector $\mathbf{v}$ in the span of the columns of $\mathbf{V}_k$ we have

$$\mathbf{v} = \mathbf{V}_k\mathbf{u} = \mathbf{A}\mathbf{V}_k\mathbf{\Lambda}_k^{-1}\mathbf{u},$$

where $\mathbf{\Lambda}_k^{-1}$ is the diagonal matrix with inverse entries, so that the columns of $\mathbf{A}$ span the same k-dimensional space, implying the rank of a symmetric matrix $\mathbf{A}$ is equal to the number of non-zero eigenvalues.

For a matrix with all eigenvalues non-zero we can write

$$\mathbf{A}^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}',$$

as

$$\mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}'\mathbf{V}\mathbf{\Lambda}\mathbf{V}' = \mathbf{I},$$

showing again that only full rank matrices are invertible.

For symmetric matrices the spectral norm can now be simply evaluated since the eigen-decomposition of $\mathbf{A}'\mathbf{A} = \mathbf{A}^2$ is given by

$$\mathbf{A}^2 = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'\mathbf{V}\mathbf{\Lambda}\mathbf{V}' = \mathbf{V}\mathbf{\Lambda}^2\mathbf{V}',$$

so that the spectrum of $\mathbf{A}^2$ is $\left\{\lambda^2 : \lambda \in \lambda\left(\mathbf{A}\right)\right\}$. Hence, by (3.3) we have

$$\|\mathbf{A}\| = \max_{\lambda\in\lambda(\mathbf{A})} |\lambda| \, .$$

The Courant–Fisher Theorem gives a further characterisation of eigenvalues extending the characterisation of the largest eigenvalue given by the Raleigh quotient. It considers maximising or minimising the quotient in a subspace T of specified dimension, and then choosing the subspace either to minimise the maximum or maximise the minimum. The largest eigenvalue case corresponds to taking the dimension of T to be that of the whole space and hence maximising the quotient in the whole space.

Theorem 3.6 (Courant–Fisher) *If $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric, then for $k = 1, \ldots, n$, the kth eigenvalue $\lambda_k(A)$ of the matrix A satisfies*

$$\lambda_k(\mathbf{A}) = \max_{\dim(T)=k} \min_{0\neq \mathbf{v}\in T} \frac{\mathbf{v}'\mathbf{A}\mathbf{v}}{\mathbf{v}'\mathbf{v}} = \min_{\dim(T)=n-k+1} \max_{0\neq \mathbf{v}\in T} \frac{\mathbf{v}'\mathbf{A}\mathbf{v}}{\mathbf{v}'\mathbf{v}},$$

with the extrema achieved by the corresponding eigenvector.

Positive semi-definite matrices A symmetric matrix is *positive semi-definite*, if its eigenvalues are all non-negative. By Theorem 3.6 this holds if and only if

$$\mathbf{v}'\mathbf{A}\mathbf{v} \geq 0$$

for all vectors $\mathbf{v}$, since the minimal eigenvalue satisfies

$$\lambda_m(\mathbf{A}) = \min_{0\neq \mathbf{v}\in\mathbb{R}^n} \frac{\mathbf{v}'\mathbf{A}\mathbf{v}}{\mathbf{v}'\mathbf{v}}.$$

Similarly a matrix is *positive definite*, if its eigenvalues are positive or equivalently

$$\mathbf{v}'\mathbf{A}\mathbf{v} > 0, \text{ for } \mathbf{v} \neq \mathbf{0}.$$

We now give two results concerning positive semi-definite matrices.

Proposition 3.7 *Gram and kernel matrices are positive semi-definite.*

Proof Considering the general case of a kernel matrix let

$$\mathbf{G}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle, \text{ for } i, j = 1, \ldots, \ell.$$

For any vector $\mathbf{v}$ we have

$$\begin{aligned} \mathbf{v}'\mathbf{G}\mathbf{v} &= \sum_{i,j=1}^{\ell} v_i v_j \mathbf{G}_{ij} = \sum_{i,j=1}^{\ell} v_i v_j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle \\ &= \left\langle \sum_{i=1}^{\ell} v_i \boldsymbol{\phi}(\mathbf{x}_i), \sum_{j=1}^{\ell} v_j \boldsymbol{\phi}(\mathbf{x}_j) \right\rangle \\ &= \left\| \sum_{i=1}^{\ell} v_i \boldsymbol{\phi}(\mathbf{x}_i) \right\|^2 \geq 0, \end{aligned}$$

as required. □

Proposition 3.8 *A matrix* $\mathbf{A}$ *is positive semi-definite if and only if* $\mathbf{A} = \mathbf{B}'\mathbf{B}$ *for some real matrix* $\mathbf{B}$.

Proof Suppose $\mathbf{A} = \mathbf{B}'\mathbf{B}$, then for any vector $\mathbf{v}$ we have

$$\mathbf{v}'\mathbf{A}\mathbf{v}{=}\mathbf{v}'\mathbf{B}'\mathbf{B}\mathbf{v} = \|\mathbf{B}\mathbf{v}\|^2 \geq 0,$$

implying $\mathbf{A}$ is positive semi-definite.

Now suppose $\mathbf{A}$ is positive semi-definite. Let $\mathbf{AV} = \mathbf{V\Lambda}$ be the eigen-decomposition of $\mathbf{A}$ and set $\mathbf{B} = \sqrt{\mathbf{\Lambda}}\mathbf{V}'$, where $\sqrt{\mathbf{\Lambda}}$ is the diagonal matrix with entries $\left(\sqrt{\mathbf{\Lambda}}\right)_{ii} = \sqrt{\lambda_i}$. The matrix exists since the eigenvalues are non-negative. Then

$$\mathbf{B}'\mathbf{B} = \mathbf{V}\sqrt{\mathbf{\Lambda}}\sqrt{\mathbf{\Lambda}}\mathbf{V}' = \mathbf{V\Lambda V}' = \mathbf{AVV}' = \mathbf{A},$$

as required. □

The choice of the matrix $\mathbf{B}$ in the proposition is not unique. For example the *Cholesky decomposition* of a positive semi-definite matrix $\mathbf{A}$ provides an alternative factorisation

$$\mathbf{A} = \mathbf{R}'\mathbf{R},$$

where the matrix $\mathbf{R}$ is upper-triangular with a non-negative diagonal. The Cholesky decomposition is the unique factorisation that has this property; see Chapter 5 for more details.

The next proposition gives another useful characterisation of positive (semi-) definiteness.

Proposition 3.9 *A matrix* $\mathbf{A}$ *is positive (semi-)definite if and only if all of its principal minors are positive (semi-)definite.*

Proof Consider a $k \times k$ minor $\mathbf{M}$ of $\mathbf{A}$. Clearly by inserting 0s in the positions of the rows that were not chosen for the minor $\mathbf{M}$ we can extend any vector $\mathbf{u} \in \mathbb{R}^k$ to a vector $\mathbf{v} \in \mathbb{R}^n$. Observe that for $\mathbf{A}$ positive semi-definite

$$\mathbf{u}'\mathbf{M}\mathbf{u} = \mathbf{v}'\mathbf{A}\mathbf{v} \geq 0,$$

with strict inequality if $\mathbf{A}$ is positive definite and $\mathbf{u} \neq \mathbf{0}$. Hence, if $\mathbf{A}$ is positive (semi-)definite so is $\mathbf{M}$. The reverse implication follows, since $\mathbf{A}$ is a principal minor of itself. □

Note that each diagonal entry is a principal minor and so must be non-negative for a positive semi-definite matrix.

Determinant and trace The *determinant* $\det(\mathbf{A})$ of a square matrix $\mathbf{A}$ is the product of its eigenvalues. Hence, for a positive definite matrix the determinant will be strictly positive, while for singular matrices it will be zero.

If we consider the matrix as a linear transformation

$$\mathbf{x} \longmapsto \mathbf{A}\mathbf{x} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'\mathbf{x},$$

$\mathbf{V}'\mathbf{x}$ computes the projection of $\mathbf{x}$ onto the eigenvectors that form the columns of $\mathbf{V}$, multiplication by $\boldsymbol{\Lambda}$ rescales the projections, while the product with $\mathbf{V}$ recomputes the resulting vector. Hence the image of the unit sphere is an ellipse with its principal axes equal to the eigenvectors and with its lengths equal to the eigenvalues. The ratio of the volume of the image of the unit sphere to its pre-image is therefore equal to the absolute value of the determinant (the determinant is negative if the sphere has undergone a reflection). The same holds for any translation of a cube of any size aligned with the principal axes. Since we can approximate any shape arbitrarily closely with a collection of such cubes, it follows that the ratio of the volume of the image of any object to that of its pre-image is equal to the determinant. If we follow $\mathbf{A}$ with a second transformation $\mathbf{B}$ and consider the volume ratios, we conclude that $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$.

The *trace* $\operatorname{tr}(\mathbf{A})$ of a $n \times n$ square matrix $\mathbf{A}$ is the sum of its diagonal entries

$$\operatorname{tr}(\mathbf{A}) = \sum_{i=1}^{n} \mathbf{A}_{ii}.$$

Since we have

$$\operatorname{tr}(\mathbf{AB}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij}\mathbf{B}_{ji} = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{B}_{ij}\mathbf{A}_{ji} = \operatorname{tr}(\mathbf{BA}),$$

the trace remains invariant under transformations of the form $\mathbf{A} \longrightarrow \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ for unitary $\mathbf{V}$ since

$$\operatorname{tr}(\mathbf{V}^{-1}\mathbf{A}\mathbf{V}) = \operatorname{tr}((\mathbf{A}\mathbf{V})\mathbf{V}^{-1}) = \operatorname{tr}(\mathbf{A}).$$

It follows by taking $\mathbf{V}$ from the eigen-decomposition of $\mathbf{A}$ that the trace of a matrix is equal to the sum of its eigenvalues.

3.2 Characterisation of kernels

Recall that a kernel function computes the inner product of the images under an embedding $\boldsymbol{\phi}$ of two data points

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle .$$

We have seen how forming a matrix of the pairwise evaluations of a kernel function on a set of inputs gives a positive semi-definite matrix. We also saw in Chapter 2 how a kernel function implicitly defines a feature space that in many cases we do not need to construct explicitly. This second observation suggests that we may also want to create kernels without explicitly constructing the feature space. Perhaps the structure of the data and our knowledge of the particular application suggest a way of comparing two inputs. The function that makes this comparison is a candidate for a kernel function.

A general characterisation So far we have only one way of verifying that the function is a kernel, that is to construct a feature space for which the function corresponds to first performing the feature mapping and then computing the inner product between the two images. For example we used this technique to show the polynomial function is a kernel and to show that the exponential of the cardinality of a set intersection is a kernel.

We will now introduce an alternative method of demonstrating that a candidate function is a kernel. This will provide one of the theoretical tools needed to create new kernels, and combine old kernels to form new ones.

One of the key observations is the relation with positive semi-definite matrices. As we saw above the kernel matrix formed by evaluating a kernel on all pairs of any set of inputs is positive semi-definite. This forms the basis of the following definition.

Definition 3.10 [Finitely positive semi-definite functions] A function

$$\kappa : X \times X \longrightarrow \mathbb{R}$$

satisfies the finitely positive semi-definite property if it is a symmetric function for which the matrices formed by restriction to any finite subset of the space X are positive semi-definite. ■

Note that this definition does not require the set X to be a vector space. We will now demonstrate that the finitely positive semi-definite property characterises kernels. We will do this by explicitly constructing the feature space assuming only this property. We first state the result in the form of a theorem.

Theorem 3.11 (Characterisation of kernels) *A function*

$$\kappa : X \times X \longrightarrow \mathbb{R},$$

which is either continuous or has a finite domain, can be decomposed

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle$$

into a feature map ϕ into a Hilbert space F applied to both its arguments followed by the evaluation of the inner product in F if and only if it satisfies the finitely positive semi-definite property.

Proof The 'only if' implication is simply the result of Proposition 3.7. We will now show the reverse implication. We therefore assume that κ satisfies the finitely positive semi-definite property and proceed to construct a feature mapping $\boldsymbol{\phi}$ into a Hilbert space for which κ is the kernel.

There is one slightly unusual aspect of the construction in that the elements of the feature space will in fact be functions. They are, however, points in a vector space and will fulfil all the required properties. Recall our observation in Section 3.1.1 that learning a weight vector is equivalent to identifying an element of the feature space, in our case one of the functions. It is perhaps natural therefore that the feature space is actually the set of functions that we will be using in the learning problem

$$\mathcal{F} = \left\{ \sum_{i=1}^{\ell} \alpha_i \kappa(\mathbf{x}_i, \cdot) : \ \ell \in \mathbb{N}, \ \mathbf{x}_i \in X, \ \alpha_i \in \mathbb{R}, \ i = 1, \dots, \ell \right\}.$$

We have chosen to use a caligraphic $\mathcal{F}$ reserved for function spaces rather than the normal F of a feature space to emphasise that the elements are functions. We should, however, emphasise that this feature space is a set of points that are in fact functions. Note that we have used a $\cdot$ to indicate the position of the argument of the function. Clearly, the space is closed under multiplication by a scalar and addition of functions, where addition is defined by

$$f, g \in \mathcal{F} \Longrightarrow (f + g)(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}).$$

Hence, $\mathcal{F}$ is a vector space. We now introduce an inner product on $\mathcal{F}$ as follows. Let $f, g \in \mathcal{F}$ be given by

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) \quad \text{and} \quad g(\mathbf{x}) = \sum_{i=1}^{n} \beta_i \kappa(\mathbf{z}_i, \mathbf{x})$$

then we define

$$\langle f, g\rangle = \sum_{i=1}^{\ell}\sum_{j=1}^{n} \alpha_i \beta_j \kappa(\mathbf{x}_i, \mathbf{z}_j) = \sum_{i=1}^{\ell} \alpha_i g(\mathbf{x}_i) = \sum_{j=1}^{n} \beta_j f(\mathbf{z}_j), \tag{3.4}$$

where the second and third equalities follow from the definitions of f and g. It is clear from these equalities that $\langle f, g\rangle$ is real-valued, symmetric and bilinear and hence satisfies the properties of an inner product, provided

$$\langle f, f\rangle \geq 0 \text{ for all } f \in \mathcal{F}.$$

But this follows from the assumption that all kernel matrices are positive semi-definite, since

$$\langle f, f\rangle = \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} \geq 0,$$

where $\boldsymbol{\alpha}$ is the vector with entries α_i, $i = 1, \ldots, \ell$, and $\mathbf{K}$ is the kernel matrix constructed on $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\ell$.

There is a further property that follows directly from the equations (3.4) if we take $g = \kappa(\mathbf{x}, \cdot)$

$$\langle f, \kappa(\mathbf{x}, \cdot)\rangle = \sum_{i=1}^{\ell} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) = f(\mathbf{x}). \tag{3.5}$$

This fact is known as the *reproducing property* of the kernel. It remains to show the two additional properties of completeness and separability. Separability will follow if the input space is countable or the kernel is continuous, but we omit the technical details of the proof of this fact. For completeness consider a fixed input $\mathbf{x}$ and a Cauchy sequence $(f_n)_{n=1}^{\infty}$. We have

$$(f_n(\mathbf{x}) - f_m(\mathbf{x}))^2 = \langle f_n - f_m, \kappa(\mathbf{x}, \cdot)\rangle^2 \leq \|f_n - f_m\|^2 \kappa(\mathbf{x}, \mathbf{x})$$

by the Cauchy–Schwarz inequality. Hence, $f_n(\mathbf{x})$ is a bounded Cauchy sequence of real numbers and hence has a limit. If we define the function

$$g(\mathbf{x}) = \lim_{n\to\infty} f_n(\mathbf{x}),$$

and include all such limit functions in $\mathcal{F}$ we obtain the Hilbert space F_κ associated with the kernel κ.

We have constructed the feature space, but must specify the image of an input $\mathbf{x}$ under the mapping $\boldsymbol{\phi}$

$$\boldsymbol{\phi} : \mathbf{x} \in X \longmapsto \boldsymbol{\phi}(\mathbf{x}) = \kappa(\mathbf{x}, \cdot) \in F_\kappa.$$

We can now evaluate the inner product between an element of F_κ and the image of an input $\mathbf{x}$ using equation (3.5)

$$\langle f, \boldsymbol{\phi}(\mathbf{x}) \rangle = \langle f, \kappa(\mathbf{x}, \cdot) \rangle = f(\mathbf{x}).$$

This is precisely what we require, namely that the function f can indeed be represented as the linear function defined by an inner product (with itself) in the feature space F_κ. Furthermore the inner product is strict since if $\|f\| = 0$, then for all $\mathbf{x}$ we have that

$$f(\mathbf{x}) = \langle f, \boldsymbol{\phi}(\mathbf{x}) \rangle \leq \|f\| \, \|\boldsymbol{\phi}(\mathbf{x})\| = 0.$$

□

Given a function κ that satisfies the finitely positive semi-definite property we will refer to the corresponding space F_κ as its *Reproducing Kernel Hilbert Space (RKHS)*. Similarly, we will use the notation $\langle \cdot, \cdot \rangle_{F_\kappa}$ for the corresponding inner product when we wish to emphasise its genesis.

Remark 3.12 [Reproducing property] We have shown how any kernel can be used to construct a Hilbert space in which the reproducing property holds. It is fairly straightforward to see that if a symmetric function $\kappa(\cdot, \cdot)$ satisfies the reproducing property in a Hilbert space F of functions

$$\langle \kappa(\mathbf{x}, \cdot), f(\cdot) \rangle_{\mathcal{F}} = f(\mathbf{x}), \text{ for } f \in \mathcal{F},$$

then κ satisfies the finitely positive semi-definite property, since

$$\begin{aligned}
\sum_{i,j=1}^{\ell} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \langle \kappa(\mathbf{x}_i, \cdot), \kappa(\mathbf{x}_j, \cdot) \rangle_{\mathcal{F}} \\
&= \left\langle \sum_{i=1}^{\ell} \alpha_i \kappa(\mathbf{x}_i, \cdot), \sum_{j=1}^{\ell} \alpha_j \kappa(\mathbf{x}_j, \cdot) \right\rangle_{\mathcal{F}} \\
&= \left\| \sum_{i=1}^{\ell} \alpha_i \kappa(\mathbf{x}_i, \cdot) \right\|_{\mathcal{F}}^2 \geq 0.
\end{aligned}$$

■

Mercer kernel We are now able to show Mercer's theorem as a consequence of the previous analysis. Mercer's theorem is usually used to construct a feature space for a valid kernel. Since we have already achieved this with the RKHS construction, we do not actually require Mercer's theorem itself. We include it for completeness and because it defines the feature space in terms of an explicit feature vector rather than using the function space of our RKHS construction. Recall the definition of the function space $L_2(X)$ from Example 3.4.

Theorem 3.13 (Mercer) *Let X be a compact subset of $\mathbb{R}^n$. Suppose κ is a continuous symmetric function such that the integral operator $T_\kappa : L_2(X) \to L_2(X)$*

$$(T_\kappa f)(\cdot) = \int_X \kappa(\cdot, \mathbf{x}) f(\mathbf{x}) d\mathbf{x},$$

is positive, that is

$$\int_{X \times X} \kappa(\mathbf{x}, \mathbf{z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geq 0,$$

for all $f \in L_2(X)$. Then we can expand $\kappa(\mathbf{x}, \mathbf{z})$ in a uniformly convergent series (on $X \times X$) in terms of functions $\boldsymbol{\phi}_j$, satisfying $\langle \boldsymbol{\phi}_j, \boldsymbol{\phi}_i \rangle = \delta_{ij}$

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty} \boldsymbol{\phi}_j(\mathbf{x}) \boldsymbol{\phi}_j(\mathbf{z}).$$

Furthermore, the series $\sum_{i=1}^{\infty} \|\boldsymbol{\phi}_i\|^2_{L_2(X)}$ is convergent.

Proof The theorem will follow provided the positivity of the integral operator implies our condition that all finite submatrices are positive semi-definite. Suppose that there is a finite submatrix on the points $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ that is not positive semi-definite. Let the vector $\boldsymbol{\alpha}$ be such that

$$\sum_{i,j=1}^{\ell} \kappa(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j = \epsilon < 0,$$

and let

$$f_\sigma(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i \frac{1}{(2\pi\sigma)^{d/2}} \exp\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right) \in L_2(X),$$

where d is the dimension of the space X. We have that

$$\lim_{\sigma \to 0} \int_{X \times X} \kappa(\mathbf{x}, \mathbf{z}) f_\sigma(\mathbf{x}) f_\sigma(\mathbf{z}) d\mathbf{x} d\mathbf{z} = \epsilon.$$

But then for some $\sigma > 0$ the integral will be less than 0 contradicting the positivity of the integral operator.

Now consider an orthonormal basis $\boldsymbol{\phi}_i(\cdot)$, $i = 1, \ldots$ of F_κ the RKHS of the kernel κ. Then we have the Fourier series for $\kappa(\mathbf{x}, \cdot)$

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{\infty} \langle \kappa(\mathbf{x}, \cdot), \boldsymbol{\phi}_i(\cdot) \rangle \boldsymbol{\phi}_i(\mathbf{z}) = \sum_{i=1}^{\infty} \boldsymbol{\phi}_i(\mathbf{x}) \boldsymbol{\phi}_i(\mathbf{z}),$$

as required.

Finally, to show that the series $\sum_{i=1}^{\infty} \|\boldsymbol{\phi}_i\|^2_{L_2(X)}$ is convergent, using the compactness of X we obtain

$$\begin{aligned} \infty &> \int_X \kappa(\mathbf{x}, \mathbf{x}) d\mathbf{x} = \lim_{n\to\infty} \int_X \sum_{i=1}^{n} \boldsymbol{\phi}_i(\mathbf{x}) \boldsymbol{\phi}_i(\mathbf{x}) d\mathbf{x} \\ &= \lim_{n\to\infty} \sum_{i=1}^{n} \int_X \boldsymbol{\phi}_i(\mathbf{x}) \boldsymbol{\phi}_i(\mathbf{x}) d\mathbf{x} = \lim_{n\to\infty} \sum_{i=1}^{n} \|\boldsymbol{\phi}_i\|^2_{L_2(X)} \end{aligned}$$

□

Example 3.14 Consider the kernel function $\kappa(\mathbf{x}, \mathbf{z}) = \kappa(\mathbf{x} - \mathbf{z})$. Such a kernel is said to be *translation invariant*, since the inner product of two inputs is unchanged if both are translated by the same vector. Consider the one-dimensional case in which κ is defined on the interval $[0, 2\pi]$ in such a way that $\kappa(u)$ can be extended to a continuous, symmetric, periodic function on $\mathbb{R}$. Such a function can be expanded in a uniformly convergent Fourier series

$$\kappa(u) = \sum_{n=0}^{\infty} a_n \cos(nu).$$

In this case we can expand $\kappa(x - z)$ as follows

$$\kappa(x - z) = a_0 + \sum_{n=1}^{\infty} a_n \sin(nx) \sin(nz) + \sum_{n=1}^{\infty} a_n \cos(nx) \cos(nz).$$

Provided the a_n are all positive this shows $\kappa(x, z)$ is the inner product in the feature space defined by the orthogonal features

$$\{\boldsymbol{\phi}_i(x)\}_{i=0}^{\infty} = (1, \sin(x), \cos(x), \sin(2x), \cos(2x), \ldots, \sin(nx), \cos(nx), \ldots),$$

since the functions, 1, $\cos(nu)$ and $\sin(nu)$ form a set of orthogonal functions on the interval $[0, 2\pi]$. Hence, normalising them will provide a set of Mercer features. Note that the embedding is defined independently of the parameters a_n, which subsequently control the geometry of the feature space.

Example 3.14 provides some useful insight into the role that the choice of kernel can play. The parameters a_n in the expansion of $\kappa(u)$ are its Fourier coefficients. If, for some n, we have $a_n = 0$, the corresponding features are removed from the feature space. Similarly, small values of a_n mean that the feature is given low weighting and so will have less influence on the choice of hyperplane. Hence, the choice of kernel can be seen as choosing a filter with a particular spectral characteristic, the effect of which is to control the influence of the different frequencies in determining the optimal separation.

Covariance kernels The fact that Mercer's theorem enables us to express a kernel as a sum over a set of functions of the product of their values on the two inputs

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty} \boldsymbol{\phi}_j(\mathbf{x})\boldsymbol{\phi}_j(\mathbf{z}).$$

This suggests a different view of kernels as a covariance function determined by a probability distribution over a function class. In general, given a distribution $q(f)$ over a function class $\mathcal{F}$, the covariance function is given by

$$\kappa_q(\mathbf{x}, \mathbf{z}) = \int_{\mathcal{F}} f(\mathbf{x}) f(\mathbf{z}) q(f) df.$$

We will refer to such a kernel as a *covariance kernel*. We can see that this is a kernel by considering the mapping

$$\boldsymbol{\phi} : \mathbf{x} \longmapsto (f(\mathbf{x}))_{f \in \mathcal{F}}$$

into the space of functions on $\mathcal{F}$ with inner product given by

$$\langle a(\cdot), b(\cdot) \rangle = \int_{\mathcal{F}} a(f) \, b(f) \, q(f) \, df.$$

This definition is quite natural if we consider that the ideal kernel for learning a function f is given by

$$\kappa_f(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) f(\mathbf{z}), \tag{3.6}$$

since the space $\mathcal{F} = \mathcal{F}_{\kappa_f}$ in this case contains functions of the form

$$\sum_{i=1}^{\ell} \alpha_i \kappa_f(\mathbf{x}_i, \cdot) = \sum_{i=1}^{\ell} \alpha_i f(\mathbf{x}_i) f(\cdot) = C f(\cdot).$$

So for the kernel κ_f, the corresponding $\mathcal{F}$ is one-dimensional, containing only multiples of f. We can therefore view κ_q as taking a combination of these simple kernels for all possible f weighted according to the prior distribution q. Any kernel derived in this way is a valid kernel, since it is easily verified that it satisfies the finitely positive semi-definite property

$$\begin{aligned}
\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j\kappa_q(\mathbf{x}_i,\mathbf{x}_j) &= \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j\int_{\mathcal{F}}f(\mathbf{x}_i)f(\mathbf{x}_j)q(f)df \\
&= \int_{\mathcal{F}}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j f(\mathbf{x}_i)f(\mathbf{x}_j)q(f)df \\
&= \int_{\mathcal{F}}\left(\sum_{i=1}^{\ell}\alpha_i f(\mathbf{x}_i)\right)^2 q(f)df \geq 0.
\end{aligned}$$

Furthermore, if the underlying class $\mathcal{F}$ of functions are $\{-1,+1\}$-valued, the kernel κ_q will be normalised since

$$\kappa_q(\mathbf{x},\mathbf{x}) = \int_{\mathcal{F}}f(\mathbf{x})f(\mathbf{x})q(f)df = \int_{\mathcal{F}}q(f)df = 1.$$

We will now show that every kernel can be obtained as a covariance kernel in which the distribution has a particular form. Given a valid kernel κ, consider the Gaussian prior q that generates functions f according to

$$f(\mathbf{x}) = \sum_{i=1}^{\infty}u_i\boldsymbol{\phi}_i(\mathbf{x}),$$

where $\boldsymbol{\phi}_i$ are the orthonormal functions of Theorem 3.13 for the kernel κ, and u_i are generated according to the Gaussian distribution $\mathcal{N}(0,1)$ with mean 0 and standard deviation 1. Notice that this function will be in $L_2(X)$ with probability 1, since using the orthonormality of the $\boldsymbol{\phi}_i$ we can bound its expected norm by

$$\begin{aligned}
\mathbb{E}\left[\|f\|^2_{L_2(X)}\right] &= \mathbb{E}\left[\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}u_iu_j\left\langle\boldsymbol{\phi}_i,\boldsymbol{\phi}_j\right\rangle_{L_2(X)}\right] \\
&= \sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\mathbb{E}\left[u_iu_j\right]\left\langle\boldsymbol{\phi}_i,\boldsymbol{\phi}_j\right\rangle_{L_2(X)} \\
&= \sum_{i=1}^{\infty}\mathbb{E}[u_i^2]\|\boldsymbol{\phi}_i\|^2_{L_2(X)} = \sum_{i=1}^{\infty}\|\boldsymbol{\phi}_i\|^2_{L_2(X)} < \infty,
\end{aligned}$$

where the final inequality follows from Theorem 3.13. Since the norm is a positive function it follows that the measure of functions not in $L_2(X)$ is 0, as otherwise the expectation would not be finite. But curiously the function will almost certainly not be in $\mathcal{F}_\kappa$ for infinite-dimensional feature spaces. We therefore take the distribution q to be defined over the space $L_2(X)$.

The covariance function κ_q is now equal to

$$\begin{aligned}
\kappa_q(\mathbf{x},\mathbf{z}) &= \int_{L_2(X)} f(\mathbf{x})f(\mathbf{z})q(f)df \\
&= \lim_{n\to\infty} \sum_{i,j=1}^{n} \phi_i(\mathbf{x})\phi_j(\mathbf{z}) \int_{\mathbb{R}^n} u_i u_j \prod_{k=1}^{n} \left(\frac{1}{\sqrt{2\pi}} \exp(-u_k^2/2) du_k \right) \\
&= \lim_{n\to\infty} \sum_{i,j=1}^{n} \phi_i(\mathbf{x})\phi_j(\mathbf{z})\delta_{ij} = \sum_{i=1}^{\infty} \phi_i(\mathbf{x})\phi_i(\mathbf{z}) \\
&= \kappa(\mathbf{x},\mathbf{z}).
\end{aligned}$$

3.3 The kernel matrix

Given a training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ and kernel function $\kappa(\cdot,\cdot)$, we introduced earlier the kernel or Gram matrix $\mathbf{K} = (\mathbf{K}_{ij})_{i,j=1}^{\ell}$ with entries

$$\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j), \text{ for } i,j = 1,\ldots,\ell.$$

The last subsection was devoted to showing that the function κ is a valid kernel provided its kernel matrices are positive semi-definite for all training sets S, the so-called finitely positive semi-definite property. This fact enables us to manipulate kernels without necessarily considering the corresponding feature space. Provided we maintain the finitely positive semi-definite property we are guaranteed that we have a valid kernel, that is, that there exists a feature space for which it is the corresponding kernel function. Reasoning about the similarity measure implied by the kernel function may be more natural than performing an explicit construction of its feature space.

The intrinsic modularity of kernel machines also means that any kernel function can be used provided it produces symmetric, positive semi-definite kernel matrices, and any kernel algorithm can be applied, as long as it can accept as input such a matrix together with any necessary labelling information. In other words, the kernel matrix acts as an interface between the data input and learning modules.

Kernel matrix as information bottleneck In view of our characterisation of kernels in terms of the finitely positive semi-definite property, it becomes clear why the kernel matrix is perhaps the core ingredient in the theory of kernel methods. It contains all the information available in order to perform the learning step, with the sole exception of the output labels in the case of supervised learning. It is worth bearing in mind that it is only through the kernel matrix that the learning algorithm obtains information about the choice of feature space or model, and indeed the training data itself.

The finitely positive semi-definite property can also be used to justify intermediate processing steps designed to improve the representation of the data, and hence the overall performance of the system through manipulating the kernel matrix before it is passed to the learning machine. One simple example is the addition of a constant to the diagonal of the matrix. This has the effect of introducing a soft margin in classification or equivalently regularisation in regression, something that we have already seen in the ridge regression example. We will, however, describe more complex manipulations of the kernel matrix that correspond to more subtle tunings of the feature space.

In view of the fact that it is only through the kernel matrix that the learning algorithm receives information about the feature space and input data, it is perhaps not surprising that some properties of this matrix can be used to assess the generalization performance of a learning system. The properties vary according to the type of learning task and the subtlety of the analysis, but once again the kernel matrix plays a central role both in the derivation of generalisation bounds and in their evaluation in practical applications.

The kernel matrix is not only the central concept in the design and analysis of kernel machines, it can also be regarded as the central data structure in their implementation. As we have seen, the kernel matrix acts as an interface between the data input module and the learning algorithms. Furthermore, many model adaptation and selection methods are implemented by manipulating the kernel matrix as it is passed between these two modules. Its properties affect every part of the learning system from the computation, through the generalisation analysis, to the implementation details.

Remark 3.15 [Implementation issues] One small word of caution is perhaps worth mentioning on the implementation side. Memory constraints mean that it may not be possible to store the full kernel matrix in memory for very large datasets. In such cases it may be necessary to recompute the kernel function as entries are needed. This may have implications for both the choice of algorithm and the details of the implementation. ■

Another important aspect of our characterisation of valid kernels in terms of the finitely positive semi-definite property is that the same condition holds for kernels defined over any kind of inputs. We did not require that the inputs should be real vectors, so that the characterisation applies whatever the type of the data, be it strings, discrete structures, images, time series, and so on. Provided the kernel matrices corresponding to any finite training set are positive semi-definite the kernel computes the inner product after projecting pairs of inputs into some feature space. Figure 3.1 illustrates this point with an embedding showing objects being mapped to feature vectors by the mapping $\boldsymbol{\phi}$.

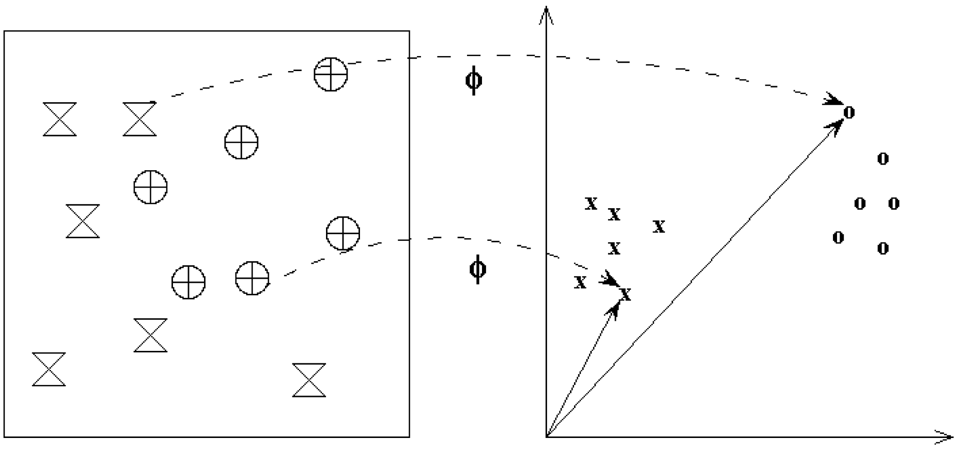

Fig. 3.1. The use of kernels enables the application of the algorithms to non-vectorial data.

Remark 3.16 [Kernels and prior knowledge] The kernel contains all of the information available to the learning machine about the relative positions of the inputs in the feature space. Naturally, if structure is to be discovered in the data set, the data must exhibit that structure through the kernel matrix. If the kernel is too general and does not give enough importance to specific types of similarity. In the language of our discussion of priors this corresponds to giving weight to too many different classifications. The kernel therefore views with the same weight any pair of inputs as similar or dissimilar, and so the off-diagonal entries of the kernel matrix become very small, while the diagonal entries are close to 1. The kernel can therefore only represent the concept of identity. This leads to overfitting since we can easily classify a training set correctly, but the kernel has no way of generalising to new data. At the other extreme, if a kernel matrix is completely uniform, then every input is similar to every other input. This corresponds to every input being mapped to the same feature vector and leads to underfitting of the data since the only functions that can be represented easily are those which map all points to the same class. Geometrically the first situation corresponds to inputs being mapped to orthogonal points in the feature space, while in the second situation all points are merged into the same image. In both cases there are no non-trivial natural classes in the data, and hence no real structure that can be exploited for generalisation. ■

Remark 3.17 [Kernels as oracles] It is possible to regard a kernel as defining a similarity measure between two data points. It can therefore be considered as an oracle, guessing the similarity of two inputs. If one uses normalised kernels, this can be thought of as the a priori probability of the inputs being in the same class minus the a priori probability of their being in different classes. In the case of a covariance kernel over a class of classification functions this is precisely the meaning of the kernel function under the prior distribution $q(f)$, since

$$\kappa_q(\mathbf{x},\mathbf{z}) = \int_{\mathcal{F}} f(\mathbf{x})f(\mathbf{z})q(f)df = P_q\left(f(\mathbf{x}) = f(\mathbf{z})\right) - P_q\left(f(\mathbf{x}) \neq f(\mathbf{z})\right).$$

■

Remark 3.18 [Priors over eigenfunctions] Notice that the kernel matrix can be decomposed as follows

$$\mathbf{K} = \sum_{i=1}^{\ell} \lambda_i \mathbf{v}_i \mathbf{v}_i',$$

where $\mathbf{v}_i$ are eigenvectors and λ_i are the corresponding eigenvalues. This decomposition is reminiscent of the form of a covariance kernel if we view each eigenvector $\mathbf{v}_i$ as a function over the set of examples and treat the eigenvalues as a (unnormalised) distribution over these functions. We can think of the eigenvectors as defining a feature space, though this is restricted to the training set in the form given above. Extending this to the eigenfunctions of the underlying integral operator

$$f(\cdot) \longmapsto \int_X \kappa(\mathbf{x},\cdot) f(\mathbf{x})\, d\mathbf{x}$$

gives another construction for the feature space of Mercer's theorem. We can therefore think of a kernel as defining a prior over the eigenfunctions of the kernel operator. This connection will be developed further when we come to consider principle components analysis. In general, defining a good kernel involves incorporating the functions that are likely to arise in the particular application and excluding others. ■

Remark 3.19 [Hessian matrix] For supervised learning with a target vector of $\{+1, -1\}$ values $\mathbf{y}$, we will often consider the matrix $\mathbf{H}_{ij} = y_i y_j \mathbf{K}_{ij}$. This matrix is known as the *Hessian* for reasons to be clarified later. It can be defined as the Schur product (entrywise multiplication) of the matrix $\mathbf{y}\mathbf{y}'$ and $\mathbf{K}$. If $\lambda, \mathbf{v}$ is an eigenvalue-eigenvector pair of $\mathbf{K}$ then $\lambda, \mathbf{u}$ is an eigenvalue-eigenvector pair of $\mathbf{H}$, where $u_i = v_i y_i$, for all i. ■

Selecting a kernel We have already seen in the covariance kernels how the choice of kernel amounts to encoding our prior expectation about the possible functions we may be expected to learn. Ideally we select the kernel based on our prior knowledge of the problem domain and restrict the learning to the task of selecting the particular pattern function in the feature space defined by the chosen kernel. Unfortunately, it is not always possible to make the right choice of kernel a priori. We are rather forced to consider a family of kernels defined in a way that again reflects our prior expectations, but which leaves open the choice of the particular kernel that will be used. The learning system must now solve two tasks, that of choosing a kernel from the family, and either subsequently or concurrently of selecting a pattern function in the feature space of the chosen kernel.

Many different approaches can be adopted for solving this two-part learning problem. The simplest examples of kernel families require only limited amount of additional information that can be estimated from the training data, frequently without using the label information in the case of a supervised learning task.

More elaborate methods that make use of the labelling information need a measure of 'goodness' to drive the kernel selection stage of the learning. This can be provided by introducing a notion of similarity between kernels and choosing the kernel that is closest to the ideal kernel described in equation (3.6) given by $\kappa(\mathbf{x}, \mathbf{z}) = y(\mathbf{x})y(\mathbf{z})$. A measure of matching between kernels or, in the case of the ideal kernel, between a kernel and a target should satisfy some basic properties: it should be symmetric, should be maximised when its arguments are equal, and should be minimised when applied to two independent kernels.

Furthermore, in practice the comparison with the ideal kernel will only be feasible when restricted to the kernel matrix on the training set rather than between complete functions, since the ideal kernel can only be computed on the training data. It should therefore be possible to justify that reliable estimates of the true similarity can be obtained using only the training set.

Cone of kernel matrices Positive semi-definite matrices form a *cone* in the vector space of $\ell \times \ell$ matrices, where by cone we mean a set closed under addition and under multiplication by non-negative scalars. This is important if we wish to optimise over such matrices, since it implies that they will be convex, an important property in ensuring the existence of efficient methods. The study of optimization over such sets is known as semi-definite programming (SDP). In view of the central role of the kernel matrix in the above discussion, it is perhaps not surprising that this recently developed field has started to play a role in kernel optimization algorithms.

We now introduce a measure of similarity between two kernels. First consider the *Frobenius inner product* between pairs of matrices with identical dimensions

$$\langle \mathbf{M}, \mathbf{N} \rangle = \mathbf{M} \cdot \mathbf{N} = \sum_{i,j=1}^{\ell} \mathbf{M}_{ij} \mathbf{N}_{ij} = \operatorname{tr}(\mathbf{M}'\mathbf{N}).$$

The corresponding matrix norm is known as the *Frobenius norm*. Furthermore if we consider $\operatorname{tr}(\mathbf{M}'\mathbf{N})$ as a function of $\mathbf{M}$, its gradient is of course $\mathbf{N}$.

Based on this inner product a simple measure of similarity between two kernel matrices $\mathbf{K}_1$and $\mathbf{K}_2$ is the following:

Definition 3.20 The *alignment* $A(\mathbf{K}_1, \mathbf{K}_2)$ between two kernel matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ is given by

$$A(\mathbf{K}_1, \mathbf{K}_2) = \frac{\langle \mathbf{K}_1, \mathbf{K}_2 \rangle}{\sqrt{\langle \mathbf{K}_1, \mathbf{K}_1 \rangle \langle \mathbf{K}_2, \mathbf{K}_2 \rangle}}$$

The alignment between a kernel $\mathbf{K}$ and a target $\mathbf{y}$ is simply $A(\mathbf{K}, \mathbf{y}\mathbf{y}')$, as $\mathbf{y}\mathbf{y}'$ is the ideal kernel for that target. For $\mathbf{y} \in \{-1, +1\}^{\ell}$ this becomes

$$A(\mathbf{K}, \mathbf{y}\mathbf{y}') = \frac{\mathbf{y}'\mathbf{K}\mathbf{y}}{\ell \|\mathbf{K}\|}.$$

■

Since the alignment can be viewed as the cosine of the angle between the matrices viewed as ℓ^2-dimensional vectors, it satisfies $-1 \leq A(\mathbf{K}_1, \mathbf{K}_2) \leq 1$.

The definition of alignment has not made use of the fact that the matrices we are considering are positive semi-definite. For such matrices the lower bound on alignment is in fact 0 as can be seen from the following proposition.

Proposition 3.21 *Let* $\mathbf{M}$ *be symmetric. Then* $\mathbf{M}$ *is positive semi-definite if and only if* $\langle \mathbf{M}, \mathbf{N} \rangle \geq 0$ *for every positive semi-definite* $\mathbf{N}$.

Proof Let $\lambda_1, \lambda_2, \ldots, \lambda_\ell$ be the eigenvalues of $\mathbf{M}$ with corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\ell$. It follows that

$$\langle \mathbf{M}, \mathbf{N} \rangle = \left\langle \sum_{i=1}^{\ell} \lambda_i \mathbf{v}_i \mathbf{v}_i', \mathbf{N} \right\rangle = \sum_{i=1}^{\ell} \lambda_i \langle \mathbf{v}_i \mathbf{v}_i', \mathbf{N} \rangle = \sum_{i=1}^{\ell} \lambda_i \mathbf{v}_i' \mathbf{N} \mathbf{v}_i.$$

Note that $\mathbf{v}_i' \mathbf{N} \mathbf{v}_i \geq 0$ if $\mathbf{N}$ is positive semi-definite and we can choose $\mathbf{N}$ so that only one of these is non-zero. Furthermore, $\mathbf{M}$ is positive semi-definite if and only if $\lambda_i \geq 0$ for all i, and so $\langle \mathbf{M}, \mathbf{N} \rangle \geq 0$ for all positive semi-definite $\mathbf{N}$ if and only if $\mathbf{M}$ is positive semi-definite. □

The alignment can also be considered as a Pearson correlation coefficient between the random variables $\mathbf{K}_1(\mathbf{x}, \mathbf{z})$ and $\mathbf{K}_2(\mathbf{x}, \mathbf{z})$ generated with a uniform distribution over the pairs $(\mathbf{x}_i, \mathbf{z}_j)$. It is also easily related to the distance between the normalised kernel matrices in the Frobenius norm

$$\left\| \frac{\mathbf{K}_1}{\|\mathbf{K}_1\|} - \frac{\mathbf{K}_2}{\|\mathbf{K}_2\|} \right\| = 2 - A(\mathbf{K}_1, \mathbf{K}_2)$$

3.4 Kernel construction

The characterization of kernel functions and kernel matrices given in the previous sections is not only useful for deciding whether a given candidate is a valid kernel. One of its main consequences is that it can be used to justify a series of rules for manipulating and combining simple kernels to obtain more complex and useful ones. In other words, such operations on one or more kernels can be shown to preserve the finitely positive semi-definiteness 'kernel' property. We will say that the class of kernel functions is *closed* under such operations. These will include operations on kernel functions and operations directly on the kernel matrix. As long as we can guarantee that the result of an operation will always be a positive semi-definite symmetric matrix, we will still be embedding the data in a feature space, albeit a feature space transformed by the chosen operation. We first consider the case of operations on the kernel function.

3.4.1 Operations on kernel functions

The following proposition can be viewed as showing that kernels satisfy a number of closure properties, allowing us to create more complicated kernels from simple building blocks.

Proposition 3.22 (Closure properties) *Let κ_1 and κ_2 be kernels over $X \times X$, $X \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^+$, $f(\cdot)$ a real-valued function on X, ϕ: $X \longrightarrow \mathbb{R}^N$ with κ_3 a kernel over $\mathbb{R}^N \times \mathbb{R}^N$, and $\mathbf{B}$ a symmetric positive semi-definite $n \times n$ matrix. Then the following functions are kernels:*

(i) $\kappa(\mathbf{x},\mathbf{z}) = \kappa_1(\mathbf{x},\mathbf{z}) + \kappa_2(\mathbf{x},\mathbf{z})$,
(ii) $\kappa(\mathbf{x},\mathbf{z}) = a\kappa_1(\mathbf{x},\mathbf{z})$,
(iii) $\kappa(\mathbf{x},\mathbf{z}) = \kappa_1(\mathbf{x},\mathbf{z})\kappa_2(\mathbf{x},\mathbf{z})$,
(iv) $\kappa(\mathbf{x},\mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$,
(v) $\kappa(\mathbf{x},\mathbf{z}) = \kappa_3(\phi(\mathbf{x}),\phi(\mathbf{z}))$,
(vi) $\kappa(\mathbf{x},\mathbf{z}) = \mathbf{x}'\mathbf{B}\mathbf{z}$.

Proof Let S a finite set of points $\{\mathbf{x}_1,\dots,\mathbf{x}_\ell\}$, and let $\mathbf{K}_1$ and $\mathbf{K}_2$, be the corresponding kernel matrices obtained by restricting κ_1 and κ_2 to these points. Consider any vector $\boldsymbol{\alpha} \in \mathbb{R}^\ell$. Recall that a matrix $\mathbf{K}$ is positive semi-definite if and only if $\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} \geq 0$, for all $\boldsymbol{\alpha}$.

(i) We have

$$\boldsymbol{\alpha}'\left(\mathbf{K}_1 + \mathbf{K}_2\right)\boldsymbol{\alpha} = \boldsymbol{\alpha}'\mathbf{K}_1\boldsymbol{\alpha} + \boldsymbol{\alpha}'\mathbf{K}_2\boldsymbol{\alpha} \geq 0,$$

and so $\mathbf{K}_1+\mathbf{K}_2$ is positive semi-definite and $\kappa_1 + \kappa_2$ a kernel function.

(ii) Similarly $\boldsymbol{\alpha}'a\mathbf{K}_1\boldsymbol{\alpha} = a\boldsymbol{\alpha}'\mathbf{K}_1\boldsymbol{\alpha} \geq 0$, verifying that $a\kappa_1$ is a kernel.

(iii) Let

$$\mathbf{K} = \mathbf{K}_1 \bigotimes \mathbf{K}_2$$

be the tensor product of the matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ obtained by replacing each entry of $\mathbf{K}_1$ by $\mathbf{K}_2$ multiplied by that entry. The tensor product of two positive semi-definite matrices is itself positive semi-definite since the eigenvalues of the product are all pairs of products of the eigenvalues of the two components. The matrix corresponding to the function $\kappa_1\kappa_2$ is known as the *Schur product* $\mathbf{H}$ of $\mathbf{K}_1$ and $\mathbf{K}_2$ with entries the products of the corresponding entries in the two components. The matrix $\mathbf{H}$ is a principal submatrix of $\mathbf{K}$ defined by a set of columns and the same set of rows. Hence for any $\boldsymbol{\alpha} \in \mathbb{R}^\ell$, there is a corresponding $\boldsymbol{\alpha}_1 \in \mathbb{R}^{\ell^2}$, such that

$$\boldsymbol{\alpha}'\mathbf{H}\boldsymbol{\alpha} = \boldsymbol{\alpha}_1'\mathbf{K}\boldsymbol{\alpha}_1 \geq 0,$$

and so $\mathbf{H}$ is positive semi-definite as required.

(iv) Consider the 1-dimensional feature map

$$\phi : \mathbf{x} \longmapsto f(\mathbf{x}) \in \mathbb{R};$$

then $\kappa(\mathbf{x}, \mathbf{z})$ is the corresponding kernel.

(v) Since κ_3 is a kernel, the matrix obtained by restricting κ_3 to the points $\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_\ell)$ is positive semi-definite as required.

(vi) Consider the diagonalisation of $\mathbf{B} = \mathbf{V}'\boldsymbol{\Lambda}\mathbf{V}$ by an orthogonal matrix $\mathbf{V}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix containing the non-negative eigenvalues. Let $\sqrt{\boldsymbol{\Lambda}}$ be the diagonal matrix with the square roots of the eigenvalues and set $\mathbf{A} = \sqrt{\boldsymbol{\Lambda}}\mathbf{V}$. We therefore have

$$\kappa(\mathbf{x}, \mathbf{z}) = \mathbf{x}'\mathbf{B}\mathbf{z} = \mathbf{x}'\mathbf{V}'\boldsymbol{\Lambda}\mathbf{V}\mathbf{z} = \mathbf{x}'\mathbf{A}'\mathbf{A}\mathbf{z} = \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{z} \rangle,$$

the inner product using the linear feature mapping $\mathbf{A}$.

$\square$

Remark 3.23 [Schur product] The combination of kernels given in part (iii) is often referred to as the *Schur product.* We can decompose any kernel into the Schur product of its normalisation and the 1-dimensional kernel of part (iv) with $f(\mathbf{x}) = \sqrt{\kappa(\mathbf{x}, \mathbf{x})}$. $\blacksquare$

The original motivation for introducing kernels was to search for nonlinear patterns by using linear functions in a feature space created using a nonlinear feature map. The last example of the proposition might therefore seem an irrelevance since it corresponds to a linear feature map. Despite this, such mappings can be useful in practice as they can rescale the geometry of the space, and hence change the relative weightings assigned to different linear functions. In Chapter 10 we will describe the use of such feature maps in applications to document analysis.

Proposition 3.24 *Let* $\kappa_1(\mathbf{x}, \mathbf{z})$ *be a kernel over* $X \times X$*, where* $\mathbf{x}$, $\mathbf{z} \in X$*, and* $p(x)$ *is a polynomial with positive coefficients. Then the following functions are also kernels:*

(i) $\kappa(\mathbf{x}, \mathbf{z}) = p(\kappa_1(\mathbf{x}, \mathbf{z}))$,
(ii) $\kappa(\mathbf{x}, \mathbf{z}) = \exp(\kappa_1(\mathbf{x}, \mathbf{z}))$,
(iii) $\kappa(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2 / (2\sigma^2))$.

Proof We consider the three parts in turn:

(i) For a polynomial the result follows from parts (i), (ii), (iii) of Proposition 3.22 with part (iv) covering the constant term if we take $f(\cdot)$ to be a constant.

(ii) The exponential function can be arbitrarily closely approximated by polynomials with positive coefficients and hence is a limit of kernels. Since the finitely positive semi-definiteness property is closed under taking pointwise limits, the result follows.

(iii) By part (ii) we have that $\exp(\langle \mathbf{x}, \mathbf{z} \rangle / \sigma^2)$ is a kernel for $\sigma \in \mathbb{R}^+$. We now normalise this kernel (see Section 2.3.2) to obtain the kernel

$$\begin{aligned} \frac{\exp(\langle \mathbf{x}, \mathbf{z} \rangle / \sigma^2)}{\sqrt{\exp(\|\mathbf{x}\|^2 / \sigma^2) \exp(\|\mathbf{z}\|^2 / \sigma^2)}} &= \exp\left(\frac{\langle \mathbf{x}, \mathbf{z} \rangle}{\sigma^2} - \frac{\langle \mathbf{x}, \mathbf{x} \rangle}{2\sigma^2} - \frac{\langle \mathbf{z}, \mathbf{z} \rangle}{2\sigma^2} \right) \\ &= \exp\left(- \frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2} \right). \end{aligned}$$

□

Remark 3.25 [Gaussian kernel] The final kernel of Proposition 3.24 is known as the *Gaussian kernel.* These functions form the hidden units of a radial basis function network, and hence using this kernel will mean the hypotheses are radial basis function networks. It is therefore also referred to as the *RBF kernel.* We will discuss this kernel further in Chapter 9. ■

Embeddings corresponding to kernel constructions Proposition 3.22 shows that we can create new kernels from existing kernels using a number of simple operations. Our approach has demonstrated that new functions are kernels by showing that they are finitely positive semi-definite. This is sufficient to verify that the function is a kernel and hence demonstrates that there exists a feature space map for which the function computes the corresponding inner product. Often this information provides sufficient insight for the user to sculpt an appropriate kernel for a particular application. It is, however, sometimes helpful to understand the effect of the kernel combination on the structure of the corresponding feature space.

The proof of part (iv) used a feature space construction, while part (ii) corresponds to a simple re-scaling of the feature vector by $\sqrt{a}$. For the addition of two kernels in part (i) the feature vector is the concatenation of the corresponding vectors

$$\boldsymbol{\phi}(\mathbf{x}) = [\boldsymbol{\phi}_1(\mathbf{x}), \boldsymbol{\phi}_2(\mathbf{x})],$$

since

$$\begin{aligned}\kappa(\mathbf{x},\mathbf{z}) &= \langle \boldsymbol{\phi}(\mathbf{x}),\boldsymbol{\phi}(\mathbf{z})\rangle = \langle [\boldsymbol{\phi}_1(\mathbf{x}),\boldsymbol{\phi}_2(\mathbf{x})],[\boldsymbol{\phi}_1(\mathbf{z}),\boldsymbol{\phi}_2(\mathbf{z})]\rangle && (3.7)\\ &= \langle \boldsymbol{\phi}_1(\mathbf{x}),\boldsymbol{\phi}_1(\mathbf{z})\rangle + \langle \boldsymbol{\phi}_2(\mathbf{x}),\boldsymbol{\phi}_2(\mathbf{z})\rangle && (3.8)\\ &= \kappa_1(\mathbf{x},\mathbf{z})+\kappa_2(\mathbf{x},\mathbf{z}).\end{aligned}$$

For the Hadamard construction of part (iii) the corresponding features are the products of all pairs of features one from the first feature space and one from the second. Thus, the (i,j)th feature is given by

$$\boldsymbol{\phi}(\mathbf{x})_{ij} = \boldsymbol{\phi}_1(\mathbf{x})_i\boldsymbol{\phi}_2(\mathbf{x})_j \text{ for } i=1,\dots,N_1 \text{ and } j=1,\dots,N_2,$$

where N_i is the dimension of the feature space corresponding to $\boldsymbol{\phi}_i$, $i=1,2$. The inner product is now given by

$$\begin{aligned}\kappa(\mathbf{x},\mathbf{z}) &= \langle \boldsymbol{\phi}(\mathbf{x}),\boldsymbol{\phi}(\mathbf{z})\rangle = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\boldsymbol{\phi}(\mathbf{x})_{ij}\boldsymbol{\phi}(\mathbf{z})_{ij}\\ &= \sum_{i=1}^{N_1}\boldsymbol{\phi}_1(\mathbf{x})_i\boldsymbol{\phi}_1(\mathbf{z})_i\sum_{j=1}^{N_2}\boldsymbol{\phi}_2(\mathbf{x})_j\boldsymbol{\phi}_2(\mathbf{z})_j && (3.9)\\ &= \kappa_1(\mathbf{x},\mathbf{z})\kappa_2(\mathbf{x},\mathbf{z}). && (3.10)\end{aligned}$$

The definition of the feature space in this case appears to depend on the choice of coordinate system since it makes use of the specific embedding function. The fact that the new kernel can be expressed simply in terms of the base kernels shows that in fact it is invariant to this choice. For the case of an exponent of a single kernel

$$\kappa(\mathbf{x},\mathbf{z}) = \kappa_1(\mathbf{x},\mathbf{z})^s,$$

we obtain by induction that the corresponding feature space is indexed by all monomials of degree s

$$\boldsymbol{\phi}_{\mathbf{i}}(\mathbf{x}) = \boldsymbol{\phi}_1(\mathbf{x})_1^{i_1}\boldsymbol{\phi}_1(\mathbf{x})_2^{i_2}\dots\boldsymbol{\phi}_1(\mathbf{x})_N^{i_N}, \tag{3.11}$$

where $\mathbf{i} = (i_1,\dots,i_N) \in \mathbb{N}^N$ satisfies

$$\sum_{j=1}^{N} i_j = s.$$

Remark 3.26 [Feature weightings] It is important to observe that the monomial features do not all receive an equal weighting in this embedding. This is due to the fact that in this case there are repetitions in the expansion given in equation (3.11), that is, products of individual features which lead to the same function $\phi_{\mathbf{i}}$. For example, in the 2-dimensional degree-2 case, the inner product can be written as

$$\begin{aligned}\kappa(\mathbf{x},\mathbf{z}) &= 2x_1x_2z_1z_2 + x_1^2z_1^2 + x_2^2z_2^2 \\ &= \left\langle \left(\sqrt{2}x_1x_2, x_1^2, x_2^2\right), \left(\sqrt{2}z_1z_2, z_1^2, z_2^2\right) \right\rangle,\end{aligned}$$

where the repetition of the cross terms leads to a weighting factor of $\sqrt{2}$. ■

Remark 3.27 [Features of the Gaussian kernel] Note that from the proofs of parts (ii) and (iii) of Proposition 3.24 the Gaussian kernel is a polynomial kernel of infinite degree. Hence, its features are all possible monomials of input features with no restriction placed on the degrees. The Taylor expansion of the exponential function

$$\exp(x) = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$$

shows that the weighting of individual monomials falls off as $i!$ with increasing degree. ■

3.4.2 Operations on kernel matrices

We can also transform the feature space by performing operations on the kernel matrix, provided that they leave it positive semi-definite and symmetric. This type of transformation raises the question of how to compute the kernel on new test points.

In some cases we may have already constructed the kernel matrix on both the training and test points so that the transformed kernel matrix contains all of the information that we will require. In other cases the transformation of the kernel matrix corresponds to a computable transformation in the feature space, hence enabling the computation of the kernel on test points.

In addition to these computational problems there is also the danger that by adapting the kernel based on the particular kernel matrix, we may have adjusted it in a way that is too dependent on the training set and does not perform well on new data.

For the present we will ignore these concerns and mention a number of different transformations that will prove useful in different contexts, where possible explaining the corresponding effect in the feature space. Detailed presentations of these methods will be given in Chapters 5 and 6.

Simple transformations There are a number of very simple transformations that have practical significance. For example adding a constant to all of the entries in the matrix corresponds to adding an extra constant feature, as follows from parts (i) and (iv) of Proposition 3.22. This effectively augments the class of functions with an adaptable offset, though this has a slightly different effect than introducing such an offset into the algorithm itself as is done with for example support vector machines.

Another simple operation is the addition of a constant to the diagonal. This corresponds to adding a new different feature for each input, hence enhancing the independence of all the inputs. This forces algorithms to create functions that depend on more of the training points. In the case of hard margin support vector machines this results in the so-called 2-norm soft margin algorithm, to be described in Chapter 7..

A further transformation that we have already encountered in Section 2.3.2 is that of normalising the data in the feature space. This transformation can be implemented for a complete kernel matrix with a short sequence of operations, to be described in Chapter 5.

Centering data Centering data in the feature space is a more complex transformation, but one that can again be performed by operations on the kernel matrix. The aim is to move the origin of the feature space to the centre of mass of the training examples. Furthermore, the choice of the centre of mass can be characterised as the origin for which the sum of the norms of the points is minimal. Since the sum of the norms is the trace of the kernel matrix this is also equal to the sum of its eigenvalues. It follows that this choice of origin minimises the sum of the eigenvalues of the corresponding kernel matrix. We describe how to perform this centering transformation on a kernel matrix in Chapter 5.

Subspace projection In high-dimensional feature spaces there is no a priori reason why the eigenvalues of the kernel matrix should decay. If each input vector is orthogonal to the remainder, the eigenvalues will be equal to the norms of the inputs. If the points are constrained in a low-dimensional subspace, the number of non-zero eigenvalues is equal to the subspace dimension. Since the sum of the eigenvalues will still be equal to the sum of the squared norms, the individual eigenvalues will be correspondingly larger.

Although it is unlikely that data will lie exactly in a low-dimensional subspace, it is not unusual that the data can be accurately approximated by projecting into a carefully chosen low-dimensional subspace. This means that the sum of the squares of the distances between the points and their approximations is small. We will see in Chapter 6 that in this case the first eigenvectors of the covariance matrix will be a basis of the subspace, while the sum of the remaining eigenvalues will be equal to the sum of the squared residuals. Since the eigenvalues of the covariance and kernel matrices are the same, this means that the kernel matrix can be well approximated by a low-rank matrix.

It may be that the subspace corresponds to the underlying structure of the data, and the residuals are the result of measurement or estimation noise. In this case, subspace projections give a better model of the data for which the corresponding kernel matrix is given by the low-rank approximation. Hence, forming a low-rank approximation of the kernel matrix can be an effective method of de-noising the data. In Chapter 10 we will also refer to this method of finding a more accurate model of the data as *semantic focussing.*

In Chapters 5 and 6 we will present in more detail methods for creating low-rank approximations, including projection into the subspace spanned by the first eigenvectors, as well as using the subspace obtained by performing a partial Gram–Schmidt orthonormalisation of the data points in the feature space, or equivalently taking a partial Cholesky decomposition of the kernel matrix. In both cases the projections and inner products of new test points can be evaluated using just the original kernel.

Whitening If a low-dimensional approximation fails to capture the data accurately enough, we may still find an eigen-decomposition useful in order to alter the scaling of the feature space by adjusting the size of the eigenvalues. One such technique, known as *whitening*, sets all of the eigenvalues to 1, hence creating a feature space in which the data distribution is spherically symmetric. Alternatively, values may be chosen to optimise some measure of fit of the kernel, such as the alignment.

Sculpting the feature space All these operations amount to moving the points in the feature space, by sculpting their inner product matrix. In some cases those modifications can be done in response to prior information as, for example, in the cases of adding a constant to the whole matrix, adding a constant to the diagonal and normalising the data. The second type of modification makes use of parameters estimated from the matrix itself as in the examples of centering the data, subspace projection and whitening. The final example of adjusting the eigenvalues to create a kernel that fits the data will usually make use of the corresponding labels or outputs.

We can view these operations as a first phase of learning in which the most appropriate feature space is selected for the data. As with many traditional learning algorithms, kernel methods improve their performance when data are preprocessed and the right features are selected. In the case of kernels it is also possible to view this process as selecting the right topology for the input space, that is, a topology which either correctly encodes our prior knowledge concerning the similarity between data points or learns the most appropriate topology from the training set.

Viewing kernels as defining a topology suggests that we should make use of prior knowledge about invariances in the input space. For example, translations and rotations of hand written characters leave their label unchanged in a character recognition task, indicating that these transformed images, though distant in the original metric, should become close in the topology defined by the kernel.

Part III of the book will look at a number of methods for creating kernels for different data types, introducing prior knowledge into kernels, fitting a generative model to the data and creating a derived kernel, and so on. The aim of the current chapter has been to provide the framework on which these later chapters can build.

3.5 Summary

- Kernels compute the inner product of projections of two data points into a feature space.
- Kernel functions are characterised by the property that all finite kernel matrices are positive semi-definite.
- Mercer's theorem is an equivalent formulation of the finitely positive semi-definite property for vector spaces.
- The finitely positive semi-definite property suggests that kernel matrices form the core data structure for kernel methods technology.
- Complex kernels can be created by simple operations that combine simpler kernels.
- By manipulating kernel matrices one can tune the corresponding embedding of the data in the kernel-defined feature space.

3.6 Further reading and advanced topics

Jorgen P. Gram (1850–1916) was a Danish actuary, remembered for (re)discovering the famous orthonormalisation procedure that bears his name, and for studying the properties of the matrix $\mathbf{A}'\mathbf{A}$. The Gram matrix is a central concept in this book, and its many properties are well-known in linear algebra. In general, for properties of positive (semi-)definite matrices and general linear algebra, we recommend the excellent book of Carl Meyer [96], and for a discussion of the properties of the cone of PSD matrices, the collection [163].

The use of Mercer's theorem for interpreting kernels as inner products in a feature space was introduced into machine learning in 1964 by the work of Aizermann, Bravermann and Rozoener on the method of potential functions [1], but its possibilities did not begin to be fully understood until it was used in the article by Boser, Guyon and Vapnik that introduced the support vector method [16] (see also discussion in Section 2.7).

The mathematical theory of kernels is rather old: Mercer's theorem dates back to 1909 [95], and the study of reproducing kernel Hilbert spaces was developed by Aronszajn in the 1940s. This theory was used in approximation and regularisation theory, see for example the book of Wahba and her 1999 survey [152], [153]. The seed idea for polynomial kernels was contained in [104]. Reproducing kernels were extensively used in machine learning and neural networks by Poggio and Girosi from the early 1990s. [48]. Related results can be found in [97]. More references about the rich regularization literature can be found in section 4.6.

Chapter 1 of Wahba's book [152] gives a number of theoretical results on kernel functions and can be used an a reference. Closure properties are discussed in [52] and in [97]. Anova kernels were introduced by Burges and Vapnik [21]. The theory of positive definite functions was also developed in the context of covariance and correlation functions, so that classical work in statistics is closely related [153], [154].

The discussion about Reproducing Kernel Hilbert Spaces in this chapter draws on the paper of Haussler [52]. Our characterization of kernel functions, by means of the finitely positive semi-definite property, is based on a theorem of Saitoh [111]. This approach paves the way to the use of general kernels on general types of data, as suggested by [116] and developed by Watkins [155], [154] and Haussler [52]. These works have greatly extended the use of kernels, showing that they can in fact be defined on general objects, which do not need to be Euclidean spaces, allowing their use in a swathe of new real-world applications, on input spaces as diverse as biological sequences, text, and images.

The notion of kernel alignment was proposed by [33] in order to capture the idea of similarity of two kernel functions, and hence of the embedding they induce, and the information they extract from the data. A number of formal properties of such quantity are now known, many of which are discussed in the technical report , but two are most relevant here: its interpretation as the inner product in the cone of positive semi-definite matrices, and consequently its interpretation as a kernel between kernels, that is a higher order kernel function. Further papers on this theme include [70], [71]. This latest interpretation of alignment was further analysed in [102].

For constantly updated pointers to online literature and free software see the book's companion website: www.kernel-methods.net