Chapter 4

Covariance functions

We have seen that a covariance function is the crucial ingredient in a Gaussian process predictor, as it encodes our assumptions about the function which we wish to learn. From a slightly different viewpoint it is clear that in supervised learning the notion of *similarity* between data points is crucial; it is a basic assumption that points with inputs $\mathbf{x}$ which are close are likely to have similar target values y, and thus training points that are near to a test point should be informative about the prediction at that point. Under the Gaussian process view it is the covariance function that defines nearness or similarity.

similarity

An arbitrary function of input pairs $\mathbf{x}$ and $\mathbf{x}'$ will not, in general, be a valid covariance function.[1] The purpose of this chapter is to give examples of some commonly-used covariance functions and to examine their properties. Section 4.1 defines a number of basic terms relating to covariance functions. Section 4.2 gives examples of stationary, dot-product, and other non-stationary covariance functions, and also gives some ways to make new ones from old. Section 4.3 introduces the important topic of eigenfunction analysis of covariance functions, and states Mercer's theorem which allows us to express the covariance function (under certain conditions) in terms of its eigenfunctions and eigenvalues. The covariance functions given in section 4.2 are valid when the input domain $\mathcal{X}$ is a subset of $\mathbb{R}^D$. In section 4.4 we describe ways to define covariance functions when the input domain is over structured objects such as strings and trees.

valid covariance functions

4.1 Preliminaries

A *stationary* covariance function is a function of $\mathbf{x} - \mathbf{x}'$. Thus it is invariant to translations in the input space.[2] For example the squared exponential co-

stationarity

[1] To be a valid covariance function it must be positive semidefinite, see eq. (4.2).

[2] In stochastic process theory a process which has constant mean and whose covariance function is invariant to translations is called *weakly stationary*. A process is *strictly stationary* if all of its finite dimensional distributions are invariant to translations [Papoulis, 1991, sec. 10.1].

isotropy

variance function given in equation 2.16 is stationary. If further the covariance function is a function only of $|\mathbf{x} - \mathbf{x}'|$ then it is called *isotropic*; it is thus invariant to all rigid motions. For example the squared exponential covariance function given in equation 2.16 is isotropic. As k is now only a function of $r = |\mathbf{x} - \mathbf{x}'|$ these are also known as *radial basis functions* (RBFs).

dot product covariance

If a covariance function depends only on $\mathbf{x}$ and $\mathbf{x}'$ through $\mathbf{x} \cdot \mathbf{x}'$ we call it a *dot product* covariance function. A simple example is the covariance function $k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x} \cdot \mathbf{x}'$ which can be obtained from linear regression by putting $\mathcal{N}(0, 1)$ priors on the coefficients of x_d $(d = 1, \ldots, D)$ and a prior of $\mathcal{N}(0, \sigma_0^2)$ on the bias (or constant function) 1, see eq. (2.15). Another important example is the inhomogeneous polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (\sigma_0^2 + \mathbf{x} \cdot \mathbf{x}')^p$ where p is a positive integer. Dot product covariance functions are invariant to a rotation of the coordinates about the origin, but not translations.

kernel

A general name for a function k of two arguments mapping a pair of inputs $\mathbf{x} \in \mathcal{X}$, $\mathbf{x}' \in \mathcal{X}$ into $\mathbb{R}$ is a *kernel*. This term arises in the theory of integral operators, where the operator T_k is defined as

$$(T_k f)(\mathbf{x}) \;=\; \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')\, d\mu(\mathbf{x}'), \tag{4.1}$$

where μ denotes a measure; see section A.7 for further explanation of this point.[3] A real kernel is said to be *symmetric* if $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$; clearly covariance functions must be symmetric from the definition.

Gram matrix
covariance matrix

Given a set of input points $\{\mathbf{x}_i | i = 1, \ldots, n\}$ we can compute the *Gram matrix* K whose entries are $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. If k is a covariance function we call the matrix K the *covariance matrix*.

positive semidefinite

A real $n \times n$ matrix K which satisfies $Q(\mathbf{v}) = \mathbf{v}^\top K \mathbf{v} \geq 0$ for all vectors $\mathbf{v} \in \mathbb{R}^n$ is called positive semidefinite (PSD). If $Q(\mathbf{v}) = 0$ only when $\mathbf{v} = \mathbf{0}$ the matrix is positive definite. $Q(\mathbf{v})$ is called a *quadratic form*. A symmetric matrix is PSD if and only if all of its eigenvalues are non-negative. A Gram matrix corresponding to a general kernel function need not be PSD, but the Gram matrix corresponding to a covariance function is PSD.

A *kernel* is said to be positive semidefinite if

$$\int k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}')\, d\mu(\mathbf{x})\, d\mu(\mathbf{x}') \geq 0, \tag{4.2}$$

for all $f \in L_2(\mathcal{X}, \mu)$. Equivalently a kernel function which gives rise to PSD Gram matrices for any choice of $n \in \mathbb{N}$ and $\mathcal{D}$ is positive semidefinite. To see this let f be the weighted sum of delta functions at each $\mathbf{x}_i$. Since such functions are limits of functions in $L_2(\mathcal{X}, \mu)$ eq. (4.2) implies that the Gram matrix corresponding to any $\mathcal{D}$ is PSD.

upcrossing rate

For a one-dimensional Gaussian process one way to understand the characteristic length-scale of the process (if this exists) is in terms of the number of upcrossings of a level u. Adler [1981, Theorem 4.1.1] states that the expected

[3] Informally speaking, readers will usually be able to substitute $d\mathbf{x}$ or $p(\mathbf{x})d\mathbf{x}$ for $d\mu(\mathbf{x})$.

number of upcrossings $\mathbb{E}[N_u]$ of the level u on the unit interval by a zero-mean, stationary, almost surely continuous Gaussian process is given by

$$\mathbb{E}[N_u] \;=\; \frac{1}{2\pi}\sqrt{\frac{-k''(0)}{k(0)}}\exp\Big(-\frac{u^2}{2k(0)}\Big). \tag{4.3}$$

If $k''(0)$ does not exist (so that the process is not mean square differentiable) then if such a process has a zero at x_0 then it will almost surely have an infinite number of zeros in the arbitrarily small interval $(x_0, x_0+\delta)$ [Blake and Lindsey, 1973, p. 303].

4.1.1 Mean Square Continuity and Differentiability *

We now describe mean square continuity and differentiability of stochastic processes, following Adler [1981, sec. 2.2]. Let $\mathbf{x}_1, \mathbf{x}_2, \ldots$ be a sequence of points and $\mathbf{x}_*$ be a fixed point in $\mathbb{R}^D$ such that $|\mathbf{x}_k - \mathbf{x}_*| \to 0$ as $k \to \infty$. Then a process $f(\mathbf{x})$ is continuous in mean square at $\mathbf{x}_*$ if $\mathbb{E}[|f(\mathbf{x}_k) - f(\mathbf{x}_*)|^2] \to 0$ as $k \to \infty$. If this holds for all $\mathbf{x}_* \in A$ where A is a subset of $\mathbb{R}^D$ then $f(\mathbf{x})$ is said to be continuous in mean square (MS) over A. A random field is continuous in mean square at $\mathbf{x}_*$ if and only if its covariance function $k(\mathbf{x}, \mathbf{x}')$ is continuous at the point $\mathbf{x} = \mathbf{x}' = \mathbf{x}_*$. For stationary covariance functions this reduces to checking continuity at $k(\mathbf{0})$. Note that MS continuity does not necessarily imply sample function continuity; for a discussion of sample function continuity and differentiability see Adler [1981, ch. 3].

mean square continuity

The mean square derivative of $f(\mathbf{x})$ in the ith direction is defined as

$$\frac{\partial f(\mathbf{x})}{\partial x_i} \;=\; \underset{h\to 0}{\text{l.i.m}}\,\frac{f(\mathbf{x}+h\mathbf{e}_i) - f(\mathbf{x})}{h}, \tag{4.4}$$

when the limit exists, where l.i.m denotes the limit in mean square and $\mathbf{e}_i$ is the unit vector in the ith direction. The covariance function of $\partial f(\mathbf{x})/\partial x_i$ is given by $\partial^2 k(\mathbf{x}, \mathbf{x}')/\partial x_i \partial x_i'$. These definitions can be extended to higher order derivatives. For stationary processes, if the $2k$th-order partial derivative $\partial^{2k} k(\mathbf{x})/\partial^2 x_{i_1} \ldots \partial^2 x_{i_k}$ exists and is finite at $\mathbf{x} = \mathbf{0}$ then the kth order partial derivative $\partial^k f(\mathbf{x})/\partial x_{i_1} \ldots x_{i_k}$ exists for all $\mathbf{x} \in \mathbb{R}^D$ as a mean square limit. Notice that it is the properties of the kernel k around $\mathbf{0}$ that determine the smoothness properties (MS differentiability) of a stationary process.

mean square differentiability

4.2 Examples of Covariance Functions

In this section we consider covariance functions where the input domain $\mathcal{X}$ is a subset of the vector space $\mathbb{R}^D$. More general input spaces are considered in section 4.4. We start in section 4.2.1 with stationary covariance functions, then consider dot-product covariance functions in section 4.2.2 and other varieties of non-stationary covariance functions in section 4.2.3. We give an overview of some commonly used covariance functions in Table 4.1 and in section 4.2.4

we describe general methods for constructing new kernels from old. There exist several other good overviews of covariance functions, see e.g. Abrahamsen [1997].

4.2.1 Stationary Covariance Functions

In this section (and section 4.3) it will be convenient to allow kernels to be a map from $\mathbf{x} \in \mathcal{X}, \mathbf{x}' \in \mathcal{X}$ into $\mathbb{C}$ (rather than $\mathbb{R}$). If a zero-mean process f is complex-valued, then the covariance function is defined as $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[f(\mathbf{x})f^*(\mathbf{x}')]$, where * denotes complex conjugation.

A stationary covariance function is a function of $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$. Sometimes in this case we will write k as a function of a single argument, i.e. $k(\boldsymbol{\tau})$.

The covariance function of a *stationary* process can be represented as the Fourier transform of a positive finite measure.

Bochner's theorem

Theorem 4.1 *(Bochner's theorem) A complex-valued function k on $\mathbb{R}^D$ is the covariance function of a weakly stationary mean square continuous complex-valued random process on $\mathbb{R}^D$ if and only if it can be represented as*

$$k(\boldsymbol{\tau}) = \int_{\mathbb{R}^D} e^{2\pi i \mathbf{s}\cdot\boldsymbol{\tau}} \, d\mu(\mathbf{s}) \tag{4.5}$$

where μ is a positive finite measure. □

spectral density
power spectrum

The statement of Bochner's theorem is quoted from Stein [1999, p. 24]; a proof can be found in Gihman and Skorohod [1974, p. 208]. If μ has a density $S(\mathbf{s})$ then S is known as the *spectral density* or *power spectrum* corresponding to k.

The construction given by eq. (4.5) puts non-negative power into each frequency $\mathbf{s}$; this is analogous to the requirement that the prior covariance matrix Σ_p on the weights in equation 2.4 be non-negative definite.

In the case that the spectral density $S(\mathbf{s})$ exists, the covariance function and the spectral density are Fourier duals of each other as shown in eq. (4.6);[4] this is known as the Wiener-Khintchine theorem, see, e.g. Chatfield [1989]

$$k(\boldsymbol{\tau}) = \int S(\mathbf{s}) e^{2\pi i \mathbf{s}\cdot\boldsymbol{\tau}} \, d\mathbf{s}, \qquad S(\mathbf{s}) = \int k(\boldsymbol{\tau}) e^{-2\pi i \mathbf{s}\cdot\boldsymbol{\tau}} \, d\boldsymbol{\tau}. \tag{4.6}$$

Notice that the variance of the process is $k(\mathbf{0}) = \int S(\mathbf{s}) \, d\mathbf{s}$ so the power spectrum must be integrable to define a valid Gaussian process.

To gain some intuition for the definition of the power spectrum given in eq. (4.6) it is important to realize that the complex exponentials $e^{2\pi i \mathbf{s}\cdot\mathbf{x}}$ are eigenfunctions of a stationary kernel with respect to Lebesgue measure (see section 4.3 for further details). Thus $S(\mathbf{s})$ is, loosely speaking, the amount of power allocated on average to the eigenfunction $e^{2\pi i \mathbf{s}\cdot\mathbf{x}}$ with frequency $\mathbf{s}$. $S(\mathbf{s})$ must eventually decay sufficiently fast as $|\mathbf{s}| \to \infty$ so that it is integrable; the

[4] See Appendix A.8 for details of Fourier transforms.

rate of this decay of the power spectrum gives important information about the smoothness of the associated stochastic process. For example it can determine the mean-square differentiability of the process (see section 4.3 for further details).

If the covariance function is isotropic (so that it is a function of r, where $r = |\boldsymbol{\tau}|$) then it can be shown that $S(\mathbf{s})$ is a function of $s \triangleq |\mathbf{s}|$ only [Adler, 1981, Theorem 2.5.2]. In this case the integrals in eq. (4.6) can be simplified by changing to spherical polar coordinates and integrating out the angular variables (see e.g. Bracewell, 1986, ch. 12) to obtain

$$k(r) = \frac{2\pi}{r^{D/2-1}} \int_0^\infty S(s) J_{D/2-1}(2\pi r s) s^{D/2}\, ds, \tag{4.7}$$

$$S(s) = \frac{2\pi}{s^{D/2-1}} \int_0^\infty k(r) J_{D/2-1}(2\pi r s) r^{D/2}\, dr. \tag{4.8}$$

Note that the dependence on the dimensionality D in equation 4.7 means that the same isotropic functional form of the spectral density can give rise to different isotropic covariance functions in different dimensions. Similarly, if we start with a particular isotropic covariance function $k(r)$ the form of spectral density will in general depend on D (see, e.g. the Matérn class spectral density given in eq. (4.15)) and in fact $k(r)$ may not be valid for all D. A necessary condition for the spectral density to exist is that $\int r^{D-1}|k(r)|\, dr < \infty$; see Stein [1999, sec. 2.10] for more details.

We now give some examples of commonly-used isotropic covariance functions. The covariance functions are given in a normalized form where $k(0) = 1$; we can multiply k by a (positive) constant σ_f^2 to get any desired process variance.

Squared Exponential Covariance Function

The *squared exponential* (SE) covariance function has already been introduced in chapter 2, eq. (2.16) and has the form

squared exponential

$$k_{\mathrm{SE}}(r) = \exp\Big(-\frac{r^2}{2\ell^2}\Big), \tag{4.9}$$

with parameter ℓ defining the *characteristic length-scale*. Using eq. (4.3) we see that the mean number of level-zero upcrossings for a SE process in 1-d is $(2\pi\ell)^{-1}$, which confirms the rôle of ℓ as a length-scale. This covariance function is infinitely differentiable, which means that the GP with this covariance function has mean square derivatives of all orders, and is thus very smooth. The spectral density of the SE covariance function is $S(s) = (2\pi\ell^2)^{D/2} \exp(-2\pi^2\ell^2 s^2)$. Stein [1999] argues that such strong smoothness assumptions are unrealistic for modelling many physical processes, and recommends the Matérn class (see below). However, the squared exponential is probably the most widely-used kernel within the kernel machines field.

characteristic length-scale

infinitely divisible

The SE kernel is *infinitely divisible* in that $(k(r))^t$ is a valid kernel for all $t > 0$; the effect of raising k to the power of t is simply to rescale ℓ.

infinite network construction for SE covariance function

We now digress briefly, to show that the squared exponential covariance function can also be obtained by expanding the input $\mathbf{x}$ into a feature space defined by Gaussian-shaped basis functions centered densely in $\mathbf{x}$-space. For simplicity of exposition we consider scalar inputs with basis functions

$$\phi_c(x) \;=\; \exp\big(-\frac{(x-c)^2}{2\ell^2}\big), \tag{4.10}$$

where c denotes the centre of the basis function. From sections 2.1 and 2.2 we recall that with a Gaussian prior on the weights $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_p^2 I)$, this gives rise to a GP with covariance function

$$k(x_p, x_q) \;=\; \sigma_p^2 \sum_{c=1}^{N} \phi_c(x_p)\phi_c(x_q). \tag{4.11}$$

Now, allowing an infinite number of basis functions centered everywhere on an interval (and scaling down the variance of the prior on the weights with the number of basis functions) we obtain the limit

$$\lim_{N\to\infty} \frac{\sigma_p^2}{N} \sum_{c=1}^{N} \phi_c(x_p)\phi_c(x_q) \;=\; \sigma_p^2 \int_{c_{\min}}^{c_{\max}} \phi_c(x_p)\phi_c(x_q)dc. \tag{4.12}$$

Plugging in the Gaussian-shaped basis functions eq. (4.10) and letting the integration limits go to infinity we obtain

$$\begin{aligned} k(x_p, x_q) \;&= \sigma_p^2 \int_{-\infty}^{\infty} \exp\big(-\frac{(x_p-c)^2}{2\ell^2}\big) \exp\big(-\frac{(x_q-c)^2}{2\ell^2}\big)dc \\ &= \sqrt{\pi}\ell\sigma_p^2 \exp\big(-\frac{(x_p-x_q)^2}{2(\sqrt{2}\ell)^2}\big), \end{aligned} \tag{4.13}$$

which we recognize as a squared exponential covariance function with a $\sqrt{2}$ times longer length-scale. The derivation is adapted from MacKay [1998]. It is straightforward to generalize this construction to multivariate $\mathbf{x}$. See also eq. (4.30) for a similar construction where the centres of the basis functions are sampled from a Gaussian distribution; the constructions are equivalent when the variance of this Gaussian tends to infinity.

The Matérn Class of Covariance Functions

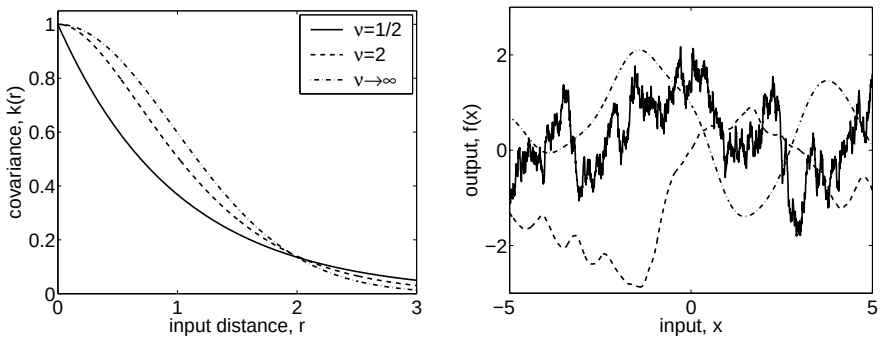
Matérn class

The *Matérn class* of covariance functions is given by

$$k_{\text{Matern}}(r) \;=\; \frac{2^{1-\nu}}{\Gamma(\nu)}\Big(\frac{\sqrt{2\nu}r}{\ell}\Big)^{\nu} K_\nu\Big(\frac{\sqrt{2\nu}r}{\ell}\Big), \tag{4.14}$$

with positive parameters ν and ℓ, where K_ν is a modified Bessel function [Abramowitz and Stegun, 1965, sec. 9.6]. This covariance function has a spectral density

$$S(s) \;=\; \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\ell^{2\nu}} \Big(\frac{2\nu}{\ell^2} + 4\pi^2 s^2\Big)^{-(\nu+D/2)} \tag{4.15}$$



(a) (b)

Figure 4.1: Panel (a): covariance functions, and (b): random functions drawn from Gaussian processes with Matérn covariance functions, eq. (4.14), for different values of ν, with $\ell = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.

in D dimensions. Note that the scaling is chosen so that for $\nu \to \infty$ we obtain the SE covariance function $e^{-r^2/2\ell^2}$, see eq. (A.25). Stein [1999] named this the Matérn class after the work of Matérn [1960]. For the Matérn class the process $f(\mathbf{x})$ is k-times MS differentiable if and only if $\nu > k$. The Matérn covariance functions become especially simple when ν is half-integer: $\nu = p + 1/2$, where p is a non-negative integer. In this case the covariance function is a product of an exponential and a polynomial of order p, the general expression can be derived from [Abramowitz and Stegun, 1965, eq. 10.2.15], giving

$$k_{\nu=p+1/2}(r) \;=\; \exp\Big(-\frac{\sqrt{2\nu}r}{\ell}\Big)\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\Big(\frac{\sqrt{8\nu}r}{\ell}\Big)^{p-i}. \qquad (4.16)$$

It is possible that the most interesting cases for machine learning are $\nu = 3/2$ and $\nu = 5/2$, for which

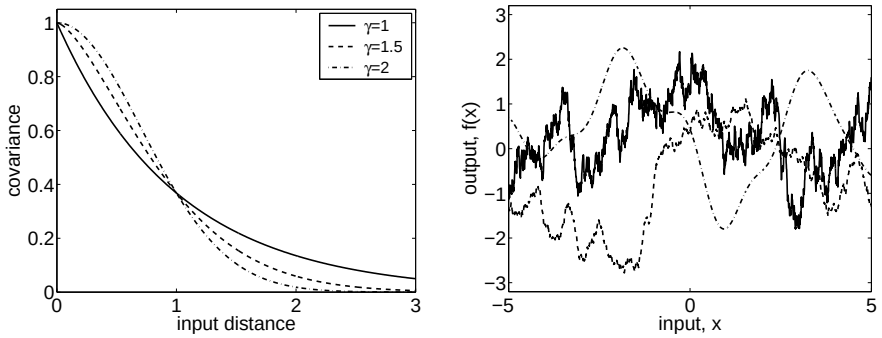
$$\begin{aligned} k_{\nu=3/2}(r) &= \Big(1+\frac{\sqrt{3}r}{\ell}\Big)\exp\Big(-\frac{\sqrt{3}r}{\ell}\Big), \\ k_{\nu=5/2}(r) &= \Big(1+\frac{\sqrt{5}r}{\ell}+\frac{5r^2}{3\ell^2}\Big)\exp\Big(-\frac{\sqrt{5}r}{\ell}\Big), \end{aligned} \qquad (4.17)$$

since for $\nu = 1/2$ the process becomes very rough (see below), and for $\nu \geq 7/2$, in the absence of explicit prior knowledge about the existence of higher order derivatives, it is probably very hard from finite noisy training examples to distinguish between values of $\nu \geq 7/2$ (or even to distinguish between finite values of ν and $\nu \to \infty$, the smooth squared exponential, in this case). For example a value of $\nu = 5/2$ was used in [Cornford et al., 2002].

Ornstein-Uhlenbeck Process and Exponential Covariance Function

The special case obtained by setting $\nu = 1/2$ in the Matérn class gives the exponential covariance function $k(r) = \exp(-r/\ell)$. The corresponding process

exponential



(a) (b)

Figure 4.2: Panel (a) covariance functions, and (b) random functions drawn from Gaussian processes with the γ-exponential covariance function eq. (4.18), for different values of γ, with $\ell = 1$. The sample functions are only differentiable when $\gamma = 2$ (the SE case). The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.

Ornstein-Uhlenbeck process

is MS continuous but not MS differentiable. In $D = 1$ this is the covariance function of the Ornstein-Uhlenbeck (OU) process. The OU process [Uhlenbeck and Ornstein, 1930] was introduced as a mathematical model of the velocity of a particle undergoing Brownian motion. More generally in $D = 1$ setting $\nu + 1/2 = p$ for integer p gives rise to a particular form of a continuous-time AR(p) Gaussian process; for further details see section B.2.1. The form of the Matérn covariance function and samples drawn from it for $\nu = 1/2$, $\nu = 2$ and $\nu \to \infty$ are illustrated in Figure 4.1.

The γ-exponential Covariance Function

γ-exponential

The γ-exponential family of covariance functions, which includes both the exponential and squared exponential, is given by

$$k(r) \;=\; \exp\big(-(r/\ell)^{\gamma}\big) \quad \text{for} \quad 0 < \gamma \leq 2. \tag{4.18}$$

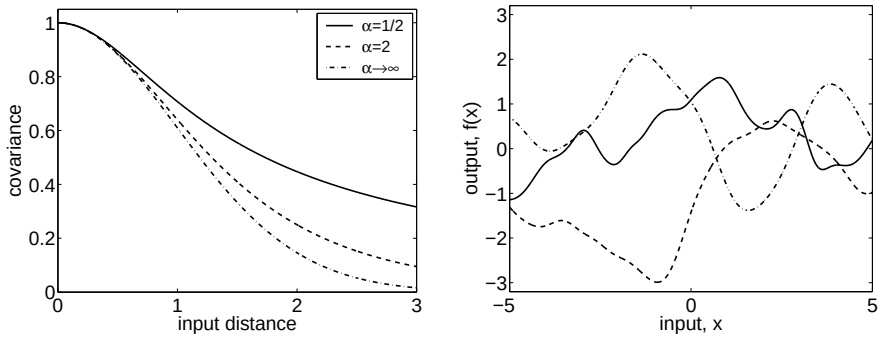
Although this function has a similar number of parameters to the Matérn class, it is (as Stein [1999] notes) in a sense less flexible. This is because the corresponding process is not MS differentiable except when $\gamma = 2$ (when it is infinitely MS differentiable). The covariance function and random samples from the process are shown in Figure 4.2. A proof of the positive definiteness of this covariance function can be found in Schoenberg [1938].

Rational Quadratic Covariance Function

rational quadratic

The *rational quadratic* (RQ) covariance function

$$k_{\mathrm{RQ}}(r) \;=\; \Big(1 + \frac{r^2}{2\alpha\ell^2}\Big)^{-\alpha} \tag{4.19}$$



(a) (b)

Figure 4.3: Panel (a) covariance functions, and (b) random functions drawn from Gaussian processes with rational quadratic covariance functions, eq. (4.20), for different values of α with $\ell = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.

with $\alpha,\ \ell > 0$ can be seen as a *scale mixture* (an infinite sum) of squared exponential (SE) covariance functions with different characteristic length-scales (sums of covariance functions are also a valid covariance, see section 4.2.4). Parameterizing now in terms of inverse squared length scales, $\tau = \ell^{-2}$, and putting a gamma distribution on $p(\tau|\alpha,\beta) \propto \tau^{\alpha-1}\exp(-\alpha\tau/\beta)$,[5] we can add up the contributions through the following integral

scale mixture

$$\begin{aligned} k_{\mathrm{RQ}}(r) &= \int p(\tau|\alpha,\beta) k_{\mathrm{SE}}(r|\tau)\, d\tau \\ &\propto \int \tau^{\alpha-1}\exp\Big(-\frac{\alpha\tau}{\beta}\Big)\exp\Big(-\frac{\tau r^2}{2}\Big)\, d\tau \ \propto\ \Big(1+\frac{r^2}{2\alpha\ell^2}\Big)^{-\alpha}, \end{aligned} \tag{4.20}$$

where we have set $\beta^{-1} = \ell^2$. The rational quadratic is also discussed by Matérn [1960, p. 17] using a slightly different parameterization; in our notation the limit of the RQ covariance for $\alpha \to \infty$ (see eq. (A.25)) is the SE covariance function with characteristic length-scale ℓ, eq. (4.9). Figure 4.3 illustrates the behaviour for different values of α; note that the process is infinitely MS differentiable for every α in contrast to the Matérn covariance function in Figure 4.1.

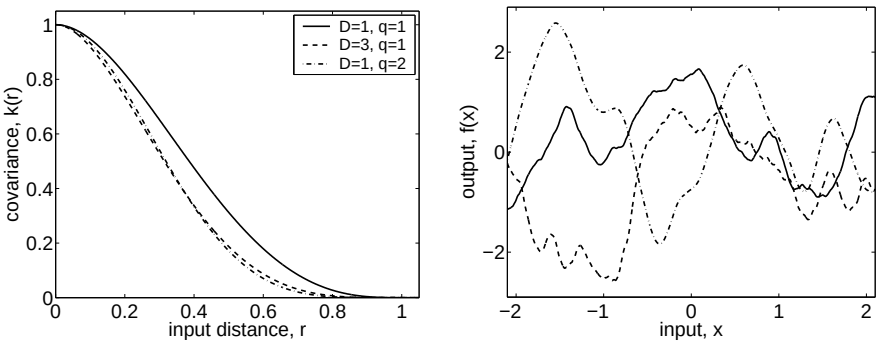
The previous example is a special case of kernels which can be written as superpositions of SE kernels with a distribution $p(\ell)$ of length-scales ℓ, $k(r) = \int \exp(-r^2/2\ell^2)p(\ell)\, d\ell$. This is in fact the most general representation for an isotropic kernel which defines a valid covariance function in any dimension D, see [Stein, 1999, sec. 2.10].

Piecewise Polynomial Covariance Functions with Compact Support

A family of piecewise polynomial functions with compact support provide another interesting class of covariance functions. Compact support means that

piecewise polynomial covariance functions with compact support

[5] Note that there are several common ways to parameterize the Gamma distribution—our choice is convenient here: α is the "shape" and β is the mean.



(a) (b)

Figure 4.4: Panel (a): covariance functions, and (b): random functions drawn from Gaussian processes with piecewise polynomial covariance functions with compact support from eq. (4.21), with specified parameters.

the covariance between points become exactly zero when their distance exceeds a certain threshold. This means that the covariance matrix will become sparse by construction, leading to the possibility of computational advantages.[6] The challenge in designing these functions is how to guarantee positive definiteness. Multiple algorithms for deriving such covariance functions are discussed by Wendland [2005, ch. 9]. These functions are usually not positive definite for all input dimensions, but their validity is restricted up to some maximum dimension D. Below we give examples of covariance functions $k_{\mathrm{pp}D,q}(r)$ which are positive definite in $\mathbb{R}^D$

positive definiteness

restricted dimension

$$
\begin{aligned}
k_{\mathrm{pp}D,0}(r) &= (1-r)_+^j, && \text{where } j = \lfloor \tfrac{D}{2} \rfloor + q + 1, \\
k_{\mathrm{pp}D,1}(r) &= (1-r)_+^{j+1}\big((j+1)r+1\big), \\
k_{\mathrm{pp}D,2}(r) &= (1-r)_+^{j+2}\big((j^2+4j+3)r^2+(3j+6)r+3\big)/3, \\
k_{\mathrm{pp}D,3}(r) &= (1-r)_+^{j+3}\big((j^3+9j^2+23j+15)r^3+ \\
&\qquad (6j^2+36j+45)r^2+(15j+45)r+15\big)/15.
\end{aligned}
\tag{4.21}
$$

The properties of three of these covariance functions are illustrated in Figure 4.4. These covariance functions are $2q$-times continuously differentiable, and thus the corresponding processes are q-times mean-square differentiable, see section 4.1.1. It is interesting to ask to what extent one could use the compactly-supported covariance functions described above in place of the other covariance functions mentioned in this section, while obtaining inferences that are similar. One advantage of the compact support is that it gives rise to sparsity of the Gram matrix which could be exploited, for example, when using iterative solutions to GPR problem, see section 8.3.6.

[6] If the product of the inverse covariance matrix with a vector (needed e.g. for prediction) is computed using a conjugate gradient algorithm, then products of the covariance matrix with vectors are the basic computational unit, and these can obviously be carried out much faster if the matrix is sparse.

Further Properties of Stationary Covariance Functions

The covariance functions given above decay monotonically with r and are always positive. However, this is not a necessary condition for a covariance function. For example Yaglom [1987] shows that $k(r) = c(\alpha r)^{-\nu} J_\nu(\alpha r)$ is a valid covariance function for $\nu \geq (D-2)/2$ and $\alpha > 0$; this function has the form of a damped oscillation.

Anisotropic versions of these isotropic covariance functions can be created by setting $r^2(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^\top M (\mathbf{x} - \mathbf{x}')$ for some positive semidefinite M. If M is diagonal this implements the use of different length-scales on different dimensions—for further discussion of automatic relevance determination see section 5.1. General M's have been considered by Matérn [1960, p. 19], Poggio and Girosi [1990] and also in Vivarelli and Williams [1999]; in the latter work a low-rank M was used to implement a linear dimensionality reduction step from the input space to lower-dimensional feature space. More generally, one could assume the form

anisotropy

$$M = \Lambda\Lambda^\top + \Psi \tag{4.22}$$

where Λ is a $D \times k$ matrix whose columns define k directions of high relevance, and Ψ is a diagonal matrix (with positive entries), capturing the (usual) axis-aligned relevances, see also Figure 5.1 on page 107. Thus M has a factor analysis form. For appropriate choices of k this may represent a good trade-off between flexibility and required number of parameters.

factor analysis distance

Stationary kernels can also be defined on a periodic domain, and can be readily constructed from stationary kernels on $\mathbb{R}$. Given a stationary kernel $k(x)$, the kernel $k_{\mathbb{T}}(x) = \sum_{m \in \mathbb{Z}} k(x + ml)$ is periodic with period l, as shown in section B.2.2 and Schölkopf and Smola [2002, eq. 4.42].

periodization

4.2.2 Dot Product Covariance Functions

As we have already mentioned above the kernel $k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x} \cdot \mathbf{x}'$ can be obtained from linear regression. If $\sigma_0^2 = 0$ we call this the homogeneous linear kernel, otherwise it is inhomogeneous. Of course this can be generalized to $k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x}^\top \Sigma_p \mathbf{x}'$ by using a general covariance matrix Σ_p on the components of $\mathbf{x}$, as described in eq. (2.4).[7] It is also the case that $k(\mathbf{x}, \mathbf{x}') = (\sigma_0^2 + \mathbf{x}^\top \Sigma_p \mathbf{x}')^p$ is a valid covariance function for positive integer p, because of the general result that a positive-integer power of a given covariance function is also a valid covariance function, as described in section 4.2.4. However, it is also interesting to show an explicit feature space construction for the polynomial covariance function. We consider the homogeneous polynomial case as the inhomogeneous case can simply be obtained by considering $\mathbf{x}$ to be extended

[7] Indeed the bias term could also be included in the general expression.

by concatenating a constant. We write

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^p = \Big(\sum_{d=1}^{D} x_d x'_d\Big)^p = \Big(\sum_{d_1=1}^{D} x_{d_1} x'_{d_1}\Big) \cdots \Big(\sum_{d_p=1}^{D} x_{d_p} x'_{d_p}\Big)$$
$$= \sum_{d_1=1}^{D} \cdots \sum_{d_p=1}^{D} (x_{d_1} \cdots x_{d_p})(x'_{d_1} \cdots x'_{d_p}) \triangleq \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}'). \quad (4.23)$$

Notice that this sum apparently contains D^p terms but in fact it is less than this as the order of the indices in the monomial $x_{d_1} \cdots x_{d_p}$ is unimportant, e.g. for $p = 2$, $x_1 x_2$ and $x_2 x_1$ are the same monomial. We can remove the redundancy by defining a vector $\mathbf{m}$ whose entry m_d specifies the number of times index d appears in the monomial, under the constraint that $\sum_{i=1}^{D} m_i = p$. Thus $\phi_{\mathbf{m}}(\mathbf{x})$, the feature corresponding to vector $\mathbf{m}$ is proportional to the monomial $x_1^{m_1} \ldots x_D^{m_D}$. The degeneracy of $\phi_{\mathbf{m}}(\mathbf{x})$ is $\frac{p!}{m_1! \ldots m_D!}$ (where as usual we define $0! = 1$), giving the feature map

$$\phi_{\mathbf{m}}(\mathbf{x}) = \sqrt{\frac{p!}{m_1! \cdots m_D!}} x_1^{m_1} \cdots x_D^{m_D}. \quad (4.24)$$

For example, for $p = 2$ in $D = 2$, we have $\boldsymbol{\phi}(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2)^\top$. Dot-product kernels are sometimes used in a normalized form given by eq. (4.35).

For regression problems the polynomial kernel is a rather strange choice as the prior variance grows rapidly with $|\mathbf{x}|$ for $|\mathbf{x}| > 1$. However, such kernels have proved effective in high-dimensional classification problems (e.g. take $\mathbf{x}$ to be a vectorized binary image) where the input data are binary or greyscale normalized to $[-1, 1]$ on each dimension [Schölkopf and Smola, 2002, sec. 7.8].

4.2.3 Other Non-stationary Covariance Functions

Above we have seen examples of non-stationary dot product kernels. However, there are also other interesting kernels which are not of this form. In this section we first describe the covariance function belonging to a particular type of neural network; this construction is due to Neal [1996].

Consider a network which takes an input $\mathbf{x}$, has one hidden layer with N_H units and then linearly combines the outputs of the hidden units with a bias b to obtain $f(\mathbf{x})$. The mapping can be written

$$f(\mathbf{x}) = b + \sum_{j=1}^{N_H} v_j h(\mathbf{x}; \mathbf{u}_j), \quad (4.25)$$

where the v_js are the hidden-to-output weights and $h(\mathbf{x}; \mathbf{u})$ is the hidden unit transfer function (which we shall assume is bounded) which depends on the input-to-hidden weights $\mathbf{u}$. For example, we could choose $h(\mathbf{x}; \mathbf{u}) = \tanh(\mathbf{x} \cdot \mathbf{u})$. This architecture is important because it has been shown by Hornik [1993] that networks with one hidden layer are universal approximators as the number of

hidden units tends to infinity, for a wide class of transfer functions (but excluding polynomials). Let b and the v's have independent zero-mean distributions of variance σ_b^2 and σ_v^2, respectively, and let the weights $\mathbf{u}_j$ for each hidden unit be independently and identically distributed. Denoting all weights by $\mathbf{w}$, we obtain (following Neal [1996])

$$\mathbb{E}_{\mathbf{w}}[f(\mathbf{x})] = 0 \tag{4.26}$$

$$\mathbb{E}_{\mathbf{w}}[f(\mathbf{x})f(\mathbf{x}')] = \sigma_b^2 + \sum_j \sigma_v^2 \mathbb{E}_{\mathbf{u}}[h(\mathbf{x};\mathbf{u}_j)h(\mathbf{x}';\mathbf{u}_j)] \tag{4.27}$$

$$= \sigma_b^2 + N_H \sigma_v^2 \mathbb{E}_{\mathbf{u}}[h(\mathbf{x};\mathbf{u})h(\mathbf{x}';\mathbf{u})], \tag{4.28}$$

where eq. (4.28) follows because all of the hidden units are identically distributed. The final term in equation 4.28 becomes $\omega^2 \mathbb{E}_{\mathbf{u}}[h(\mathbf{x};\mathbf{u})h(\mathbf{x}';\mathbf{u})]$ by letting σ_v^2 scale as ω^2/N_H.

The sum in eq. (4.27) is over N_H identically and independently distributed random variables. As the transfer function is bounded, all moments of the distribution will be bounded and hence the central limit theorem can be applied, showing that the stochastic process will converge to a Gaussian process in the limit as $N_H \to \infty$.

neural network covariance function

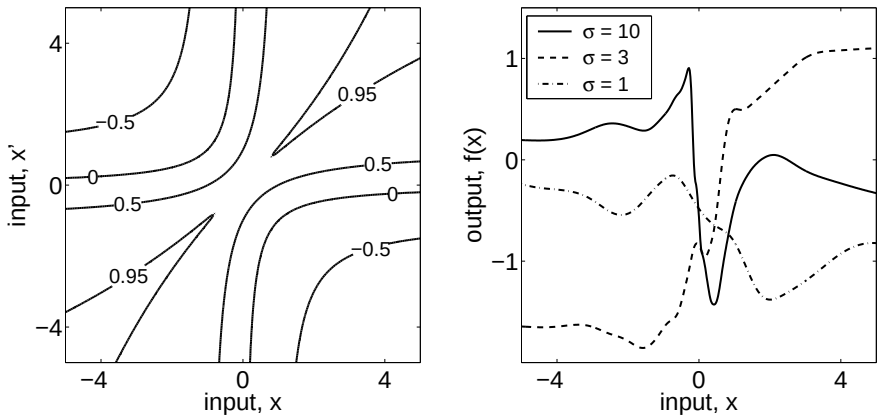
By evaluating $\mathbb{E}_{\mathbf{u}}[h(\mathbf{x};\mathbf{u})h(\mathbf{x}';\mathbf{u})]$ we can obtain the covariance function of the neural network. For example if we choose the error function $h(z) = \mathrm{erf}(z) = 2/\sqrt{\pi}\int_0^z e^{-t^2}\,dt$ as the transfer function, let $h(\mathbf{x};\mathbf{u}) = \mathrm{erf}(u_0 + \sum_{j=1}^D u_j x_j)$ and choose $\mathbf{u} \sim \mathcal{N}(0, \Sigma)$ then we obtain [Williams, 1998]

$$k_{\mathrm{NN}}(\mathbf{x}, \mathbf{x}') = \frac{2}{\pi} \sin^{-1}\Big(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1 + 2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1 + 2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\Big), \tag{4.29}$$

where $\tilde{\mathbf{x}} = (1, x_1, \ldots, x_d)^\top$ is an augmented input vector. This is a true "neural network" covariance function. The "sigmoid" kernel $k(\mathbf{x}, \mathbf{x}') = \tanh(a + b\mathbf{x}\cdot\mathbf{x}')$ has sometimes been proposed, but in fact this kernel is never positive definite and is thus not a valid covariance function, see, e.g. Schölkopf and Smola [2002, p. 113]. Figure 4.5 shows a plot of the neural network covariance function and samples from the prior. We have set $\Sigma = \mathrm{diag}(\sigma_0^2, \sigma^2)$. Samples from a GP with this covariance function can be viewed as superpositions of the functions $\mathrm{erf}(u_0 + ux)$, where σ_0^2 controls the variance of u_0 (and thus the amount of offset of these functions from the origin), and σ^2 controls u and thus the scaling on the x-axis. In Figure 4.5(b) we observe that the sample functions with larger σ vary more quickly. Notice that the samples display the non-stationarity of the covariance function in that for large values of $+x$ or $-x$ they should tend to a constant value, consistent with the construction as a superposition of sigmoid functions.

modulated squared exponential

Another interesting construction is to set $h(\mathbf{x};\mathbf{u}) = \exp(-|\mathbf{x} - \mathbf{u}|^2/2\sigma_g^2)$, where σ_g sets the scale of this Gaussian basis function. With $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma_u^2 I)$



(a), covariance (b), sample functions

Figure 4.5: Panel (a): a plot of the covariance function $k_{\mathrm{NN}}(x, x')$ for $\sigma_0 = 10$, $\sigma = 10$. Panel (b): samples drawn from the neural network covariance function with $\sigma_0 = 2$ and σ as shown in the legend. The samples were obtained using a discretization of the x-axis of 500 equally-spaced points.

we obtain

$$\begin{aligned} k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}') &= \frac{1}{(2\pi\sigma_u^2)^{d/2}} \int \exp\Big(-\frac{|\mathbf{x}-\mathbf{u}|^2}{2\sigma_g^2} - \frac{|\mathbf{x}'-\mathbf{u}|^2}{2\sigma_g^2} - \frac{\mathbf{u}^\top\mathbf{u}}{2\sigma_u^2}\Big)\, d\mathbf{u} \\ &= \Big(\frac{\sigma_e}{\sigma_u}\Big)^d \exp\Big(-\frac{\mathbf{x}^\top\mathbf{x}}{2\sigma_m^2}\Big) \exp\Big(-\frac{|\mathbf{x}-\mathbf{x}'|^2}{2\sigma_s^2}\Big) \exp\Big(-\frac{\mathbf{x}'^\top\mathbf{x}'}{2\sigma_m^2}\Big), \end{aligned} \tag{4.30}$$

where $1/\sigma_e^2 = 2/\sigma_g^2 + 1/\sigma_u^2$, $\sigma_s^2 = 2\sigma_g^2 + \sigma_g^4/\sigma_u^2$ and $\sigma_m^2 = 2\sigma_u^2 + \sigma_g^2$. This is in general a non-stationary covariance function, but if $\sigma_u^2 \to \infty$ (while scaling ω^2 appropriately) we recover the squared exponential $k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}') \propto \exp(-|\mathbf{x} - \mathbf{x}'|^2/4\sigma_g^2)$. For a finite value of σ_u^2, $k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}')$ comprises a squared exponential covariance function modulated by the Gaussian decay envelope function $\exp(-\mathbf{x}^\top\mathbf{x}/2\sigma_m^2)\exp(-\mathbf{x}'^\top\mathbf{x}'/2\sigma_m^2)$, cf. the vertical rescaling construction described in section 4.2.4.

warping

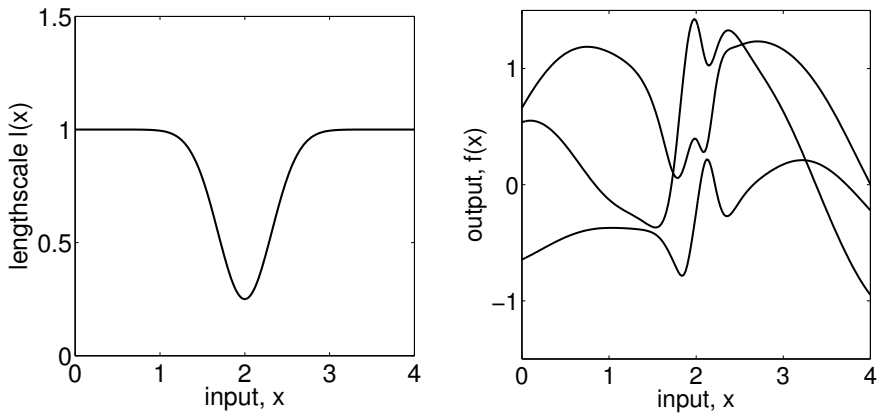
One way to introduce non-stationarity is to introduce an arbitrary non-linear mapping (or warping) $\mathbf{u}(\mathbf{x})$ of the input $\mathbf{x}$ and then use a stationary covariance function in $\mathbf{u}$-space. Note that $\mathbf{x}$ and $\mathbf{u}$ need not have the same dimensionality as each other. This approach was used by Sampson and Guttorp [1992] to model patterns of solar radiation in southwestern British Columbia using Gaussian processes.

periodic random function

Another interesting example of this warping construction is given in MacKay [1998] where the one-dimensional input variable x is mapped to the two-dimensional $\mathbf{u}(x) = (\cos(x), \sin(x))$ to give rise to a periodic random function of x. If we use the squared exponential kernel in $\mathbf{u}$-space, then

$$k(x, x') = \exp\Big(-\frac{2\sin^2\big(\frac{x-x'}{2}\big)}{\ell^2}\Big), \tag{4.31}$$

as $(\cos(x) - \cos(x'))^2 + (\sin(x) - \sin(x'))^2 = 4\sin^2(\frac{x-x'}{2})$.



(a) (b)

Figure 4.6: Panel (a) shows the chosen length-scale function $\ell(x)$. Panel (b) shows three samples from the GP prior using Gibbs' covariance function eq. (4.32). This figure is based on Fig. 3.9 in Gibbs [1997].

We have described above how to make an anisotropic covariance function by scaling different dimensions differently. However, we are not free to make these length-scales ℓ_d be functions of $\mathbf{x}$, as this will not in general produce a valid covariance function. Gibbs [1997] derived the covariance function

varying length-scale

$$k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} \Big(\frac{2\ell_d(\mathbf{x})\ell_d(\mathbf{x}')}{\ell_d^2(\mathbf{x}) + \ell_d^2(\mathbf{x}')} \Big)^{1/2} \exp\Big(-\sum_{d=1}^{D} \frac{(x_d - x_d')^2}{\ell_d^2(\mathbf{x}) + \ell_d^2(\mathbf{x}')} \Big), \qquad (4.32)$$

where each $\ell_i(\mathbf{x})$ is an arbitrary positive function of $\mathbf{x}$. Note that $k(\mathbf{x}, \mathbf{x}) = 1$ for all $\mathbf{x}$. This covariance function is obtained by considering a grid of N Gaussian basis functions with centres $\mathbf{c}_j$ and a corresponding length-scale on input dimension d which varies as a positive function $\ell_d(\mathbf{c}_j)$. Taking the limit as $N \to \infty$ the sum turns into an integral and after some algebra eq. (4.32) is obtained.

An example of a variable length-scale function and samples from the prior corresponding to eq. (4.32) are shown in Figure 4.6. Notice that as the length-scale gets shorter the sample functions vary more rapidly as one would expect. The large length-scale regions on either side of the short length-scale region can be quite strongly correlated. If one tries the converse experiment by creating a length-scale function $\ell(x)$ which has a longer length-scale region between two shorter ones then the behaviour may not be quite what is expected; on initially transitioning into the long length-scale region the covariance drops off quite sharply due to the prefactor in eq. (4.32), before stabilizing to a slower variation. See Gibbs [1997, sec. 3.10.3] for further details. Exercises 4.5.4 and 4.5.5 invite you to investigate this further.

Paciorek and Schervish [2004] have generalized Gibbs' construction to obtain non-stationary versions of arbitrary isotropic covariance functions. Let k_{S} be a

covariance function	expression	S	ND
constant	σ_0^2	$\surd$	
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$		
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$		
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$	$\surd$	$\surd$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{\ell} r\right)$	$\surd$	$\surd$
exponential	$\exp(-\frac{r}{\ell})$	$\surd$	$\surd$
γ-exponential	$\exp\left(-(\frac{r}{\ell})^\gamma\right)$	$\surd$	$\surd$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$	$\surd$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$		$\surd$

Table 4.1: Summary of several commonly-used covariance functions. The covariances are written either as a function of $\mathbf{x}$ and $\mathbf{x}'$, or as a function of $r = |\mathbf{x} - \mathbf{x}'|$. Two columns marked 'S' and 'ND' indicate whether the covariance functions are stationary and nondegenerate respectively. Degenerate covariance functions have finite rank, see section 4.3 for more discussion of this issue.

stationary, isotropic covariance function that is valid in every Euclidean space $\mathbb{R}^D$ for $D = 1, 2, \ldots$. Let $\Sigma(\mathbf{x})$ be a $D \times D$ matrix-valued function which is positive definite for all $\mathbf{x}$, and let $\Sigma_i \triangleq \Sigma(\mathbf{x}_i)$. (The set of Gibbs' $\ell_i(\mathbf{x})$ functions define a diagonal $\Sigma(\mathbf{x})$.) Then define the quadratic form

$$Q_{ij} \;=\; (\mathbf{x}_i - \mathbf{x}_j)^\top ((\Sigma_i + \Sigma_j)/2)^{-1} (\mathbf{x}_i - \mathbf{x}_j). \tag{4.33}$$

Paciorek and Schervish [2004] show that

$$k_{\mathrm{NS}}(\mathbf{x}_i, \mathbf{x}_j) \;=\; 2^{D/2} |\Sigma_i|^{1/4} |\Sigma_j|^{1/4} |\Sigma_i + \Sigma_j|^{-1/2} k_{\mathrm{S}}(\sqrt{Q_{ij}}), \tag{4.34}$$

is a valid non-stationary covariance function.

In chapter 2 we described the linear regression model in feature space $f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}$. O'Hagan [1978] suggested making $\mathbf{w}$ a function of $\mathbf{x}$ to allow for different values of $\mathbf{w}$ to be appropriate in different regions. Thus he put a Gaussian process prior on $\mathbf{w}$ of the form $\mathrm{cov}(\mathbf{w}(\mathbf{x}), \mathbf{w}(\mathbf{x}')) = W_0 k_w(\mathbf{x}, \mathbf{x}')$ for some positive definite matrix W_0, giving rise to a prior on $f(\mathbf{x})$ with covariance $k_f(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top W_0 \boldsymbol{\phi}(\mathbf{x}') k_w(\mathbf{x}, \mathbf{x}')$.

Wiener process

Finally we note that the Wiener process with covariance function $k(x, x') = \min(x, x')$ is a fundamental non-stationary process. See section B.2.1 and texts such as Grimmett and Stirzaker [1992, ch. 13] for further details.

4.2.4 Making New Kernels from Old

In the previous sections we have developed many covariance functions some of which are summarized in Table 4.1. In this section we show how to combine or modify existing covariance functions to make new ones.

The sum of two kernels is a kernel. Proof: consider the random process $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$, where $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are independent. Then $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$. This construction can be used e.g. to add together kernels with different characteristic length-scales. sum

The product of two kernels is a kernel. Proof: consider the random process $f(\mathbf{x}) = f_1(\mathbf{x})f_2(\mathbf{x})$, where $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are independent. Then $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$.[8] A simple extension of this argument means that $k^p(\mathbf{x}, \mathbf{x}')$ is a valid covariance function for $p \in \mathbb{N}$. product

Let $a(\mathbf{x})$ be a given deterministic function and consider $g(\mathbf{x}) = a(\mathbf{x})f(\mathbf{x})$ where $f(\mathbf{x})$ is a random process. Then $\mathrm{cov}(g(\mathbf{x}), g(\mathbf{x}')) = a(\mathbf{x})k(\mathbf{x}, \mathbf{x}')a(\mathbf{x}')$. Such a construction can be used to normalize kernels by choosing $a(\mathbf{x}) = k^{-1/2}(\mathbf{x}, \mathbf{x})$ (assuming $k(\mathbf{x}, \mathbf{x}) > 0\ \forall \mathbf{x}$), so that vertical rescaling

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = \frac{k(\mathbf{x}, \mathbf{x}')}{\sqrt{k(\mathbf{x}, \mathbf{x})}\sqrt{k(\mathbf{x}', \mathbf{x}')}}. \tag{4.35}$$

This ensures that $k(\mathbf{x}, \mathbf{x}) = 1$ for all $\mathbf{x}$.

We can also obtain a new process by convolution (or blurring). Consider an arbitrary fixed kernel $h(\mathbf{x}, \mathbf{z})$ and the map $g(\mathbf{x}) = \int h(\mathbf{x}, \mathbf{z})f(\mathbf{z})\, d\mathbf{z}$. Then clearly $\mathrm{cov}(g(\mathbf{x}), g(\mathbf{x}')) = \int h(\mathbf{x}, \mathbf{z})k(\mathbf{z}, \mathbf{z}')h(\mathbf{x}', \mathbf{z}')\, d\mathbf{z}\, d\mathbf{z}'$. convolution

If $k(\mathbf{x}_1, \mathbf{x}_1')$ and $k(\mathbf{x}_2, \mathbf{x}_2')$ are covariance functions over different spaces $\mathcal{X}_1$ and $\mathcal{X}_2$, then the direct sum $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}_1, \mathbf{x}_1') + k_2(\mathbf{x}_2, \mathbf{x}_2')$ and the tensor product $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}_1, \mathbf{x}_1')k_2(\mathbf{x}_2, \mathbf{x}_2')$ are also covariance functions (defined on the product space $\mathcal{X}_1 \times \mathcal{X}_2$), by virtue of the sum and product constructions. direct sum, tensor product

The direct sum construction can be further generalized. Consider a function $f(\mathbf{x})$, where $\mathbf{x}$ is D-dimensional. An *additive* model [Hastie and Tibshirani, 1990] has the form $f(\mathbf{x}) = c + \sum_{i=1}^{D} f_i(x_i)$, i.e. a linear combination of functions of one variable. If the individual f_i's are taken to be independent stochastic processes, then the covariance function of f will have the form of a direct sum. If we now admit interactions of two variables, so that $f(\mathbf{x}) = c + \sum_{i=1}^{D} f_i(x_i) + \sum_{ij,j<i} f_{ij}(x_i, x_j)$ and the various f_i's and f_{ij}'s are independent stochastic processes, then the covariance function will have the form $k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{D} k_i(x_i, x_i') + \sum_{i=2}^{D} \sum_{j=1}^{i-1} k_{ij}(x_i, x_j; x_i', x_j')$. Indeed this process can be extended further to provide a *functional ANOVA*[9] decomposition, ranging from a simple additive model up to full interaction of all D input variables. (The sum can also be truncated at some stage.) Wahba [1990, ch. 10] and Stitson et al. [1999] suggest using tensor products for kernels with interactions so that in the example above $k_{ij}(x_i, x_j; x_i', x_j')$ would have the form $k_i(x_i; x_i')k_j(x_j; x_j')$. Note that if D is large then the large number of pairwise (or higher-order) terms may be problematic; Plate [1999] has investigated using a combination of additive GP models plus a general covariance function that permits full interactions. additive model, functional ANOVA

[8] If f_1 and f_2 are Gaussian processes then the product f will not in general be a Gaussian process, but there exists a GP with this covariance function.

[9] ANOVA stands for analysis of variance, a statistical technique that analyzes the interactions between various attributes.

4.3 Eigenfunction Analysis of Kernels

We first define eigenvalues and eigenfunctions and discuss Mercer's theorem which allows us to express the kernel (under certain conditions) in terms of these quantities. Section 4.3.1 gives the analytical solution of the eigenproblem for the SE kernel under a Gaussian measure. Section 4.3.2 discusses how to compute approximate eigenfunctions numerically for cases where the exact solution is not known.

It turns out that Gaussian process regression can be viewed as Bayesian linear regression with a possibly infinite number of basis functions, as discussed in chapter 2. One possible basis set is the *eigenfunctions* of the covariance function. A function $\phi(\cdot)$ that obeys the integral equation

$$\int k(\mathbf{x}, \mathbf{x}')\phi(\mathbf{x})\, d\mu(\mathbf{x}) \;=\; \lambda\phi(\mathbf{x}'), \tag{4.36}$$

eigenvalue, eigenfunction

is called an eigenfunction of kernel k with eigenvalue λ with respect to measure[10] μ. The two measures of particular interest to us will be (i) Lebesgue measure over a compact subset $\mathcal{C}$ of $\mathbb{R}^D$, or (ii) when there is a density $p(\mathbf{x})$ so that $d\mu(\mathbf{x})$ can be written $p(\mathbf{x})d\mathbf{x}$.

In general there are an infinite number of eigenfunctions, which we label $\phi_1(\mathbf{x})$, $\phi_2(\mathbf{x}), \ldots$ We assume the ordering is chosen such that $\lambda_1 \geq \lambda_2 \geq \ldots$. The eigenfunctions are orthogonal with respect to μ and can be chosen to be normalized so that $\int \phi_i(\mathbf{x})\phi_j(\mathbf{x})\, d\mu(\mathbf{x}) = \delta_{ij}$ where δ_{ij} is the Kronecker delta.

Mercer's theorem

Mercer's theorem (see, e.g. König, 1986) allows us to express the kernel k in terms of the eigenvalues and eigenfunctions.

Theorem 4.2 *(Mercer's theorem). Let $(\mathcal{X}, \mu)$ be a finite measure space and $k \in L_\infty(\mathcal{X}^2, \mu^2)$ be a kernel such that $T_k : L_2(\mathcal{X}, \mu) \to L_2(\mathcal{X}, \mu)$ is positive definite (see eq. (4.2)). Let $\phi_i \in L_2(\mathcal{X}, \mu)$ be the normalized eigenfunctions of T_k associated with the eigenvalues $\lambda_i > 0$. Then:*

1. *the eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ are absolutely summable*
2. $$k(\mathbf{x}, \mathbf{x}') \;=\; \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x})\phi_i^*(\mathbf{x}'), \tag{4.37}$$

   holds μ^2 almost everywhere, where the series converges absolutely and uniformly μ^2 almost everywhere. □

This decomposition is just the infinite-dimensional analogue of the diagonalization of a Hermitian matrix. Note that the sum may terminate at some value $N \in \mathbb{N}$ (i.e. the eigenvalues beyond N are zero), or the sum may be infinite. We have the following definition [Press et al., 1992, p. 794]

[10] For further explanation of measure see Appendix A.7.

Definition 4.1 *A* degenerate *kernel has only a finite number of non-zero eigenvalues.* □

A degenerate kernel is also said to have finite rank. If a kernel is not degenerate it is said to be *nondegenerate*. As an example a N-dimensional linear regression model in feature space (see eq. (2.10)) gives rise to a degenerate kernel with at most N non-zero eigenvalues. (Of course if the measure only puts weight on a finite number of points n in $\mathbf{x}$-space then the eigendecomposition is simply that of a $n \times n$ matrix, even if the kernel is nondegenerate.)

degenerate, nondegenerate kernel

The statement of Mercer's theorem above referred to a finite measure μ. If we replace this with Lebesgue measure and consider a stationary covariance function, then directly from Bochner's theorem eq. (4.5) we obtain

$$k(\mathbf{x}-\mathbf{x}') \;=\; \int_{\mathbb{R}^D} e^{2\pi i \mathbf{s}\cdot(\mathbf{x}-\mathbf{x}')}\, d\mu(\mathbf{s}) \;=\; \int_{\mathbb{R}^D} e^{2\pi i \mathbf{s}\cdot\mathbf{x}} \Big(e^{2\pi i \mathbf{s}\cdot\mathbf{x}'}\Big)^* \, d\mu(\mathbf{s}). \quad (4.38)$$

The complex exponentials $e^{2\pi i \mathbf{s}\cdot\mathbf{x}}$ are the eigenfunctions of a stationary kernel w.r.t. Lebesgue measure. Note the similarity to eq. (4.37) except that the summation has been replaced by an integral.

The rate of decay of the eigenvalues gives important information about the smoothness of the kernel. For example Ritter et al. [1995] showed that in 1-d with μ uniform on $[0, 1]$, processes which are r-times mean-square differentiable have $\lambda_i \propto i^{-(2r+2)}$ asymptotically. This makes sense as "rougher" processes have more power at high frequencies, and so their eigenvalue spectrum decays more slowly. The same phenomenon can be read off from the power spectrum of the Matérn class as given in eq. (4.15).

Hawkins [1989] gives the exact eigenvalue spectrum for the OU process on $[0, 1]$. Widom [1963; 1964] gives an asymptotic analysis of the eigenvalues of stationary kernels taking into account the effect of the density $d\mu(\mathbf{x}) = p(\mathbf{x})d\mathbf{x}$; Bach and Jordan [2002, Table 3] use these results to show the effect of varying $p(\mathbf{x})$ for the SE kernel. An exact eigenanalysis of the SE kernel under the Gaussian density is given in the next section.

4.3.1 An Analytic Example *

For the case that $p(x)$ is a Gaussian and for the squared-exponential kernel $k(x, x') = \exp(-(x-x')^2/2\ell^2)$, there are analytic results for the eigenvalues and eigenfunctions, as given by Zhu et al. [1998, sec. 4]. Putting $p(x) = \mathcal{N}(x|0, \sigma^2)$ we find that the eigenvalues λ_k and eigenfunctions ϕ_k (for convenience let $k = 0, 1, \dots$) are given by

$$\lambda_k \;=\; \sqrt{\frac{2a}{A}}\, B^k, \quad (4.39)$$

$$\phi_k(x) \;=\; \exp\big(-(c-a)x^2\big) H_k\big(\sqrt{2c}x\big), \quad (4.40)$$

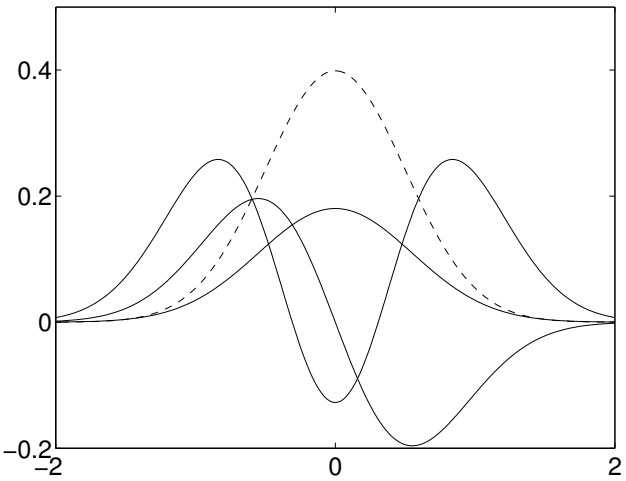


Figure 4.7: The first 3 eigenfunctions of the squared exponential kernel w.r.t. a Gaussian density. The value of $k = 0,\ 1,\ 2$ is equal to the number of zero-crossings of the function. The dashed line is proportional to the density $p(x)$.

where $H_k(x) = (-1)^k \exp(x^2) \frac{d^k}{dx^k} \exp(-x^2)$ is the kth order Hermite polynomial (see Gradshteyn and Ryzhik [1980, sec. 8.95]), $a^{-1} = 4\sigma^2$, $b^{-1} = 2\ell^2$ and

$$c = \sqrt{a^2 + 2ab}, \qquad A = a + b + c, \qquad B = b/A. \tag{4.41}$$

Hints on the proof of this result are given in exercise 4.5.9. A plot of the first three eigenfunctions for $a = 1$ and $b = 3$ is shown in Figure 4.7.

The result for the eigenvalues and eigenfunctions is readily generalized to the multivariate case when the kernel and Gaussian density are products of the univariate expressions, as the eigenfunctions and eigenvalues will simply be products too. For the case that a and b are equal on all D dimensions, the degeneracy of the eigenvalue $(\frac{2a}{A})^{D/2} B^k$ is $\binom{k+D-1}{D-1}$ which is $\mathcal{O}(k^{D-1})$. As $\sum_{j=0}^{k} \binom{j+D-1}{D-1} = \binom{k+D}{D}$ we see that the $\binom{k+D}{D}$'th eigenvalue has a value given by $(\frac{2a}{A})^{D/2} B^k$, and this can be used to determine the rate of decay of the spectrum.

4.3.2 Numerical Approximation of Eigenfunctions

The standard numerical method for approximating the eigenfunctions and eigenvalues of eq. (4.36) is to use a numerical routine to approximate the integral (see, e.g. Baker [1977, ch. 3]). For example letting $d\mu(\mathbf{x}) = p(\mathbf{x})d\mathbf{x}$ in eq. (4.36) one could use the approximation

$$\lambda_i \phi_i(\mathbf{x}') \;=\; \int k(\mathbf{x}, \mathbf{x}') p(\mathbf{x}) \phi_i(\mathbf{x})\, d\mathbf{x} \;\simeq\; \frac{1}{n} \sum_{l=1}^{n} k(\mathbf{x}_l, \mathbf{x}') \phi_i(\mathbf{x}_l), \tag{4.42}$$

where the $\mathbf{x}_l$'s are sampled from $p(\mathbf{x})$. Plugging in $\mathbf{x}' = \mathbf{x}_l$ for $l = 1, \ldots, n$ into eq. (4.42) we obtain the matrix eigenproblem

$$K\mathbf{u}_i \;=\; \lambda_i^{\text{mat}}\mathbf{u}_i, \tag{4.43}$$

where K is the $n \times n$ Gram matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, λ_i^{mat} is the ith matrix eigenvalue and $\mathbf{u}_i$ is the corresponding eigenvector (normalized so that $\mathbf{u}_i^\top \mathbf{u}_i = 1$). We have $\phi_i(\mathbf{x}_j) \sim \sqrt{n}(\mathbf{u}_i)_j$ where the $\sqrt{n}$ factor arises from the differing normalizations of the eigenvector and eigenfunction. Thus $\frac{1}{n}\lambda_i^{\text{mat}}$ is an obvious estimator for λ_i for $i = 1, \ldots, n$. For fixed n one would expect that the larger eigenvalues would be better estimated than the smaller ones. The theory of the numerical solution of eigenvalue problems shows that for a fixed i, $\frac{1}{n}\lambda_i^{\text{mat}}$ will converge to λ_i in the limit that $n \to \infty$ [Baker, 1977, Theorem 3.4]. It is also possible to study the convergence further; for example it is quite easy using the properties of principal components analysis (PCA) in feature space to show that for any l, $1 \le l \le n$, $\mathbb{E}_n[\frac{1}{n}\sum_{i=1}^{l}\lambda_i^{\text{mat}}] \ge \sum_{i=1}^{l}\lambda_i$ and $\mathbb{E}_n[\frac{1}{n}\sum_{i=l+1}^{n}\lambda_i^{\text{mat}}] \le \sum_{i=l+1}^{N}\lambda_i$, where $\mathbb{E}_n$ denotes expectation with respect to samples of size n drawn from $p(\mathbf{x})$. For further details see Shawe-Taylor and Williams [2003].

The Nyström method for approximating the ith eigenfunction (see Baker [1977] and Press et al. [1992, section 18.1]) is given by

Nyström method

$$\phi_i(\mathbf{x}') \;\simeq\; \frac{\sqrt{n}}{\lambda_i^{\text{mat}}}\mathbf{k}(\mathbf{x}')^\top \mathbf{u}_i, \tag{4.44}$$

where $\mathbf{k}(\mathbf{x}')^\top = (k(\mathbf{x}_1, \mathbf{x}'), \ldots, k(\mathbf{x}_n, \mathbf{x}'))$, which is obtained from eq. (4.42) by dividing both sides by λ_i. Equation 4.44 extends the approximation $\phi_i(\mathbf{x}_j) \simeq \sqrt{n}(\mathbf{u}_i)_j$ from the sample points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to all $\mathbf{x}$.

There is an interesting relationship between the kernel PCA method of Schölkopf et al. [1998] and the eigenfunction expansion discussed above. The eigenfunction expansion has (at least potentially) an infinite number of non-zero eigenvalues. In contrast, the kernel PCA algorithm operates on the $n \times n$ matrix K and yields n eigenvalues and eigenvectors. Eq. (4.42) clarifies the relationship between the two. However, note that eq. (4.44) is identical (up to scaling factors) to Schölkopf et al. [1998, eq. 4.1] which describes the projection of a new point $\mathbf{x}'$ onto the ith eigenvector in the kernel PCA feature space.

kernel PCA

4.4 Kernels for Non-vectorial Inputs

So far in this chapter we have assumed that the input $\mathbf{x}$ is a vector, measuring the values of a number of attributes (or features). However, for some learning problems the inputs are not vectors, but structured objects such as strings, trees or general graphs. For example, we may have a biological problem where we want to classify proteins (represented as strings of amino acid symbols).[11]

[11] Proteins are initially made up of 20 different amino acids, of which a few may later be modified bringing the total number up to 26 or 30.

Or our input may be parse-trees derived from a linguistic analysis. Or we may wish to represent chemical compounds as labelled graphs, with vertices denoting atoms and edges denoting bonds.

To follow the discriminative approach we need to extract some features from the input objects and build a predictor using these features. (For a classification problem, the alternative generative approach would construct class-conditional models over the objects themselves.) Below we describe two approaches to this feature extraction problem and the efficient computation of kernels from them: in section 4.4.1 we cover string kernels, and in section 4.4.2 we describe Fisher kernels. There exist other proposals for constructing kernels for strings, for example Watkins [2000] describes the use of pair hidden Markov models (HMMs that generate output symbols for two strings conditional on the hidden state) for this purpose.

4.4.1 String Kernels

We start by defining some notation for strings. Let $\mathcal{A}$ be a finite alphabet of characters. The concatenation of strings x and y is written xy and $|x|$ denotes the length of string x. The string s is a substring of x if we can write $x = usv$ for some (possibly empty) u, s and v.

Let $\phi_s(x)$ denote the number of times that substring s appears in string x. Then we define the kernel between two strings x and x' as

$$k(x, x') = \sum_{s \in \mathcal{A}^*} w_s \phi_s(x) \phi_s(x'), \tag{4.45}$$

where w_s is a non-negative weight for substring s. For example, we could set $w_s = \lambda^{|s|}$, where $0 < \lambda < 1$, so that shorter substrings get more weight than longer ones.

A number of interesting special cases are contained in the definition 4.45:

bag-of-characters

- Setting $w_s = 0$ for $|s| > 1$ gives the bag-of-characters kernel. This takes the feature vector for a string x to be the number of times that each character in $\mathcal{A}$ appears in x.

bag-of-words

- In text analysis we may wish to consider the frequencies of word occurrence. If we require s to be bordered by whitespace then a "bag-of-words" representation is obtained. Although this is a very simple model of text (which ignores word order) it can be surprisingly effective for document classification and retrieval tasks, see e.g. Hand et al. [2001, sec. 14.3]. The weights can be set differently for different words, e.g. using the "term frequency inverse document frequency" (TF-IDF) weighting scheme developed in the information retrieval area [Salton and Buckley, 1988].

k-spectrum kernel

- If we only consider substrings of length k, then we obtain the k-spectrum kernel [Leslie et al., 2003].

Importantly, there are efficient methods using suffix trees that can compute a string kernel $k(x, x')$ in time linear in $|x| + |x'|$ (with some restrictions on the weights $\{w_s\}$) [Leslie et al., 2003, Vishwanathan and Smola, 2003].

Work on string kernels was started by Watkins [1999] and Haussler [1999]. There are many further developments of the methods we have described above; for example Lodhi et al. [2001] go beyond substrings to consider subsequences of x which are not necessarily contiguous, and Leslie et al. [2003] describe mismatch string kernels which allow substrings s and s' of x and x' respectively to match if there are at most m mismatches between them. We expect further developments in this area, tailoring (or engineering) the string kernels to have properties that make sense in a particular domain.

The idea of string kernels, where we consider matches of substrings, can easily be extended to trees, e.g. by looking at matches of subtrees [Collins and Duffy, 2002].

Leslie et al. [2003] have applied string kernels to the classification of protein domains into SCOP[12] superfamilies. The results obtained were significantly better than methods based on either PSI-BLAST[13] searches or a generative hidden Markov model classifier. Similar results were obtained by Jaakkola et al. [2000] using a Fisher kernel (described in the next section). Saunders et al. [2003] have also described the use of string kernels on the problem of classifying natural language newswire stories from the Reuters-21578[14] database into ten classes.

4.4.2 Fisher Kernels

As explained above, our problem is that the input x is a structured object of arbitrary size e.g. a string, and we wish to extract features from it. The *Fisher kernel* (introduced by Jaakkola et al., 2000) does this by taking a generative model $p(x|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector of parameters, and computing the feature vector $\boldsymbol{\phi}_{\boldsymbol{\theta}}(x) = \nabla_{\boldsymbol{\theta}} \log p(x|\boldsymbol{\theta})$. $\boldsymbol{\phi}_{\boldsymbol{\theta}}(x)$ is sometimes called the *score vector*. score vector

Take, for example, a Markov model for strings. Let x_k be the kth symbol in string x. Then a Markov model gives $p(x|\boldsymbol{\theta}) = p(x_1|\boldsymbol{\pi}) \prod_{i=1}^{|x|-1} p(x_{i+1}|x_i, A)$, where $\boldsymbol{\theta} = (\boldsymbol{\pi}, A)$. Here $(\boldsymbol{\pi})_j$ gives the probability that x_1 will be the jth symbol in the alphabet $\mathcal{A}$, and A is a $|\mathcal{A}| \times |\mathcal{A}|$ stochastic matrix, with a_{jk} giving the probability that $p(x_{i+1} = k|x_i = j)$. Given such a model it is straightforward to compute the score vector for a given x.

It is also possible to consider other generative models $p(x|\boldsymbol{\theta})$. For example we might try a kth-order Markov model where x_i is predicted by the preceding k symbols. See Leslie et al. [2003] and Saunders et al. [2003] for an interesting discussion of the similarities of the features used in the k-spectrum kernel and the score vector derived from an order $k-1$ Markov model; see also exercise

[12] Structural classification of proteins database, `http://scop.mrc-lmb.cam.ac.uk/scop/`.

[13] Position-Specific Iterative Basic Local Alignment Search Tool, see `http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/psi1.html`.

[14] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`.

4.5.12. Another interesting choice is to use a hidden Markov model (HMM) as the generative model, as discussed by Jaakkola et al. [2000]. See also exercise 4.5.11 for a linear kernel derived from an isotropic Gaussian model for $\mathbf{x} \in \mathbb{R}^D$.

We define a kernel $k(x, x')$ based on the score vectors for x and x'. One simple choice is to set

$$k(x, x') \;=\; \boldsymbol{\phi}_{\boldsymbol{\theta}}(x) M^{-1} \boldsymbol{\phi}_{\boldsymbol{\theta}}(x'), \tag{4.46}$$

where M is a strictly positive definite matrix. Alternatively we might use the squared exponential kernel $k(x, x') = \exp(-\alpha |\boldsymbol{\phi}_{\boldsymbol{\theta}}(x) - \boldsymbol{\phi}_{\boldsymbol{\theta}}(x')|^2)$ for some $\alpha > 0$.

The structure of $p(x|\boldsymbol{\theta})$ as $\boldsymbol{\theta}$ varies has been studied extensively in information geometry (see, e.g. Amari, 1985). It can be shown that the manifold of $\log p(x|\boldsymbol{\theta})$ is Riemannian with a metric tensor which is the inverse of the *Fisher information matrix* F, where

Fisher information matrix

$$F \;=\; \mathbb{E}_x[\boldsymbol{\phi}_{\boldsymbol{\theta}}(x) \boldsymbol{\phi}_{\boldsymbol{\theta}}^\top(x)]. \tag{4.47}$$

Fisher kernel

Setting $M = F$ in eq. (4.46) gives the *Fisher kernel* . If F is difficult to compute then one might resort to setting $M = I$. The advantage of using the Fisher information matrix is that it makes arc length on the manifold invariant to reparameterizations of $\boldsymbol{\theta}$.

TOP kernel

The Fisher kernel uses a class-independent model $p(x|\boldsymbol{\theta})$. Tsuda et al. [2002] have developed the *tangent of posterior odds* (TOP) kernel based on $\nabla_{\boldsymbol{\theta}}(\log p(y = +1|x, \boldsymbol{\theta}) - \log p(y = -1|x, \boldsymbol{\theta}))$, which makes use of class-conditional distributions for the $\mathcal{C}_+$ and $\mathcal{C}_-$ classes.

4.5 Exercises

1. The OU process with covariance function $k(x - x') = \exp(-|x - x'|/\ell)$ is the unique stationary first-order Markovian Gaussian process (see Appendix B for further details). Consider training inputs $x_1 < x_2 \ldots < x_{n-1} < x_n$ on $\mathbb{R}$ with corresponding function values $\mathbf{f} = (f(x_1), \ldots, f(x_n))^\top$. Let x_l denote the nearest training input to the left of a test point x_*, and similarly let x_u denote the nearest training input to the right of x_*. Then the Markovian property means that $p(f(x_*)|\mathbf{f}) = p(f(x_*)|f(x_l), f(x_u))$. Demonstrate this by choosing some x-points on the line and computing the predictive distribution $p(f(x_*)|\mathbf{f})$ using eq. (2.19), and observing that non-zero contributions only arise from x_l and x_u. Note that this only occurs in the noise-free case; if one allows the training points to be corrupted by noise (equations 2.23 and 2.24) then all points will contribute in general.

2. Computer exercise: write code to draw samples from the neural network covariance function, eq. (4.29) in 1-d and 2-d. Consider the cases when $\mathrm{var}(u_0)$ is either 0 or non-zero. Explain the form of the plots obtained when $\mathrm{var}(u_0) = 0$.

3. Consider the random process $f(\mathbf{x}) = \text{erf}(u_0 + \sum_{i=1}^{D} u_j x_j)$, where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Show that this non-linear transform of a process with an inhomogeneous linear covariance function has the same covariance function as the erf neural network. However, note that this process is not a Gaussian process. Draw samples from the given process and compare them to your results from exercise 4.5.2.

4. Derive Gibbs' non-stationary covariance function, eq. (4.32).

5. Computer exercise: write code to draw samples from Gibbs' non-stationary covariance function eq. (4.32) in 1-d and 2-d. Investigate various forms of length-scale function $\ell(\mathbf{x})$.

6. Show that the SE process is infinitely MS differentiable and that the OU process is not MS differentiable.

7. Prove that the eigenfunctions of a symmetric kernel are orthogonal w.r.t. the measure μ.

8. Let $\tilde{k}(\mathbf{x}, \mathbf{x}') = p^{1/2}(\mathbf{x})k(\mathbf{x}, \mathbf{x}')p^{1/2}(\mathbf{x}')$, and assume $p(\mathbf{x}) > 0$ for all $\mathbf{x}$. Show that the eigenproblem $\int \tilde{k}(\mathbf{x}, \mathbf{x}')\tilde{\phi}_i(\mathbf{x})d\mathbf{x} = \tilde{\lambda}_i\tilde{\phi}_i(\mathbf{x}')$ has the same eigenvalues as $\int k(\mathbf{x}, \mathbf{x}')p(\mathbf{x})\phi_i(\mathbf{x})d\mathbf{x} = \lambda_i\phi_i(\mathbf{x}')$, and that the eigenfunctions are related by $\tilde{\phi}_i(\mathbf{x}) = p^{1/2}(\mathbf{x})\phi_i(\mathbf{x})$. Also give the matrix version of this problem (Hint: introduce a diagonal matrix P to take the rôle of $p(\mathbf{x})$). The significance of this connection is that it can be easier to find eigenvalues of symmetric matrices than general matrices.

9. Apply the construction in the previous exercise to the eigenproblem for the SE kernel and Gaussian density given in section 4.3.1, with $p(x) = \sqrt{2a/\pi}\exp(-2ax^2)$. Thus consider the modified kernel given by $\tilde{k}(x, x') = \exp(-ax^2)\exp(-b(x-x')^2)\exp(-a(x')^2)$. Using equation 7.374.8 in Gradshteyn and Ryzhik [1980]:

$$\int_{-\infty}^{\infty} \exp\big(-(x-y)^2\big) H_n(\alpha x)\, dx \;=\; \sqrt{\pi}(1-\alpha^2)^{n/2} H_n\Big(\frac{\alpha y}{(1-\alpha^2)^{1/2}}\Big),$$

verify that $\tilde{\phi}_k(x) = \exp(-cx^2)H_k(\sqrt{2c}x)$, and thus confirm equations 4.39 and 4.40.

10. Computer exercise: The analytic form of the eigenvalues and eigenfunctions for the SE kernel and Gaussian density are given in section 4.3.1. Compare these exact results to those obtained by the Nyström approximation for various values of n and choice of samples.

11. Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$. Consider the Fisher kernel derived from this model with respect to variation of $\boldsymbol{\mu}$ (i.e. regard σ^2 as a constant). Show that:

$$\left.\frac{\partial \log p(\mathbf{x}|\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}\right|_{\boldsymbol{\mu}=\mathbf{0}} \;=\; \frac{\mathbf{x}}{\sigma^2}$$

and that $F = \sigma^{-2}I$. Thus the Fisher kernel for this model with $\boldsymbol{\mu} = \mathbf{0}$ is the linear kernel $k(\mathbf{x}, \mathbf{x}') = \frac{1}{\sigma^2}\mathbf{x} \cdot \mathbf{x}'$.

12. Consider a $k-1$ order Markov model for strings on a finite alphabet. Let this model have parameters $\theta_{t|s_1,\ldots,s_{k-1}}$ denoting the probability $p(x_i = t|x_{i-1} = s_1, \ldots, x_{k-1} = s_{k-1})$. Of course as these are probabilities they obey the constraint that $\sum_{t'} \theta_{t'|s_1,\ldots,s_{k-1}} = 1$. Enforcing this constraint can be achieved automatically by setting

$$\theta_{t|s_1,\ldots,s_{k-1}} = \frac{\theta_{t,s_1,\ldots,s_{k-1}}}{\sum_{t'} \theta_{t',s_1,\ldots,s_{k-1}}},$$

where the $\theta_{t,s_1,\ldots,s_{k-1}}$ parameters are now independent, as suggested in [Jaakkola et al., 2000]. The current parameter values are denoted $\boldsymbol{\theta}^0$. Let the current values of $\theta^0_{t,s_1,\ldots,s_{k-1}}$ be set so that $\sum_{t'} \theta^0_{t',s_1,\ldots,s_{k-1}} = 1$, i.e. that $\theta^0_{t,s_1,\ldots,s_{k-1}} = \theta^0_{t|s_1,\ldots,s_{k-1}}$.

Show that $\log p(x|\boldsymbol{\theta}) = \sum n_{t,s_1,\ldots,s_{k-1}} \log \theta_{t|s_1,\ldots,s_{k-1}}$ where $n_{t,s_1,\ldots,s_{k-1}}$ is the number of instances of the substring $s_{k-1} \ldots s_1 t$ in x. Thus, following Leslie et al. [2003], show that

$$\left.\frac{\partial \log p(x|\boldsymbol{\theta})}{\partial \theta_{t,s_1,\ldots,s_{k-1}}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^0} = \frac{n_{t,s_1,\ldots,s_{k-1}}}{\theta^0_{t|s_1,\ldots,s_{k-1}}} - n_{s_1,\ldots,s_{k-1}},$$

where $n_{s_1,\ldots,s_{k-1}}$ is the number of instances of the substring $s_{k-1} \ldots s_1$ in x. As $n_{s_1,\ldots,s_{k-1}} \theta^0_{t|s_1,\ldots,s_{k-1}}$ is the expected number of occurrences of the string $s_{k-1} \ldots s_1 t$ given the count $n_{s_1,\ldots,s_{k-1}}$, the Fisher score captures the degree to which this string is over- or under-represented relative to the model. For the k-spectrum kernel the relevant feature is $\phi_{s_{k-1}\ldots,s_1,t}(x) = n_{t,s_1,\ldots,s_{k-1}}$.