C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. www.GaussianProcess.org/gpml

Chapter 7

Theoretical Perspectives

This chapter covers a number of more theoretical issues relating to Gaussian processes. In section 2.6 we saw how GPR carries out a linear smoothing of the datapoints using the weight function. The form of the weight function can be understood in terms of the equivalent kernel, which is discussed in section 7.1.

As one gets more and more data, one would hope that the GP predictions would converge to the true underlying predictive distribution. This question of consistency is reviewed in section 7.2, where we also discuss the concepts of equivalence and orthogonality of GPs.

When the generating process for the data is assumed to be a GP it is particularly easy to obtain results for *learning curves* which describe how the accuracy of the predictor increases as a function of n, as described in section 7.3. An alternative approach to the analysis of generalization error is provided by the PAC-Bayesian analysis discussed in section 7.4. Here we seek to relate (with high probability) the error observed on the training set to the generalization error of the GP predictor.

Gaussian processes are just one of the many methods that have been developed for supervised learning problems. In section 7.5 we compare and contrast GP predictors with other supervised learning methods.

7.1 The Equivalent Kernel

In this section we consider regression problems. We have seen in section 6.2 that the posterior mean for GP regression can be obtained as the function which minimizes the functional

$$J[f] \;=\; \frac{1}{2}\|f\|_{\mathcal{H}}^2 + \frac{1}{2\sigma_n^2}\sum_{i=1}^{n}\big(y_i - f(\mathbf{x}_i)\big)^2, \tag{7.1}$$

where $\|f\|_{\mathcal{H}}$ is the RKHS norm corresponding to kernel k. Our goal is now to understand the behaviour of this solution as $n \to \infty$.

Let $\mu(\mathbf{x}, y)$ be the probability measure from which the data pairs $(\mathbf{x}_i, y_i)$ are generated. Observe that

$$\mathbb{E}\Big[\sum_{i=1}^{n}\big(y_i - f(\mathbf{x}_i)\big)^2\Big] \;=\; n\int\big(y - f(\mathbf{x})\big)^2\, d\mu(\mathbf{x}, y). \tag{7.2}$$

Let $\eta(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$ be the *regression function* corresponding to the probability measure μ. The variance around $\eta(\mathbf{x})$ is denoted $\sigma^2(\mathbf{x}) = \int (y - \eta(\mathbf{x}))^2 d\mu(y|\mathbf{x})$. Then writing $y - f = (y - \eta) + (\eta - f)$ we obtain

$$\int\big(y - f(\mathbf{x})\big)^2\, d\mu(\mathbf{x}, y) \;=\; \int\big(\eta(\mathbf{x}) - f(\mathbf{x})\big)^2\, d\mu(\mathbf{x}) + \int \sigma^2(\mathbf{x})\, d\mu(\mathbf{x}), \tag{7.3}$$

as the cross term vanishes due to the definition of $\eta(\mathbf{x})$.

As the second term on the right hand side of eq. (7.3) is independent of f, an idealization of the regression problem consists of minimizing the functional

$$J_\mu[f] \;=\; \frac{n}{2\sigma_n^2}\int\big(\eta(\mathbf{x}) - f(\mathbf{x})\big)^2\, d\mu(\mathbf{x}) + \frac{1}{2}\|f\|_{\mathcal{H}}^2. \tag{7.4}$$

The form of the minimizing solution is most easily understood in terms of the eigenfunctions $\{\phi_i(\mathbf{x})\}$ of the kernel k w.r.t. to $\mu(\mathbf{x})$, where $\int \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mu(\mathbf{x}) = \delta_{ij}$, see section 4.3. Assuming that the kernel is nondegenerate so that the ϕs form a complete orthonormal basis, we write $f(\mathbf{x}) = \sum_{i=1}^{\infty} f_i\phi_i(\mathbf{x})$. Similarly, $\eta(\mathbf{x}) = \sum_{i=1}^{\infty}\eta_i\phi_i(\mathbf{x})$, where $\eta_i = \int \eta(\mathbf{x})\phi_i(\mathbf{x})d\mu(\mathbf{x})$. Thus

$$J_\mu[f] \;=\; \frac{n}{2\sigma_n^2}\sum_{i=1}^{\infty}(\eta_i - f_i)^2 + \frac{1}{2}\sum_{i=1}^{\infty}\frac{f_i^2}{\lambda_i}. \tag{7.5}$$

This is readily minimized by differentiation w.r.t. each f_i to obtain

$$f_i \;=\; \frac{\lambda_i}{\lambda_i + \sigma_n^2/n}\eta_i. \tag{7.6}$$

Notice that the term $\sigma_n^2/n \to 0$ as $n \to \infty$ so that in this limit we would expect that $f(\mathbf{x})$ will converge to $\eta(\mathbf{x})$. There are two caveats: (1) we have assumed that $\eta(\mathbf{x})$ is sufficiently well-behaved so that it can be represented by the generalized Fourier series $\sum_{i=1}^{\infty}\eta_i\phi_i(\mathbf{x})$, and (2) we assumed that the kernel is nondegenerate. If the kernel is degenerate (e.g. a polynomial kernel) then f should converge to the best μ-weighted L_2 approximation to η within the span of the ϕ's. In section 7.2.1 we will say more about rates of convergence of f to η; clearly in general this will depend on the smoothness of η, the kernel k and the measure $\mu(\mathbf{x}, y)$.

From a Bayesian perspective what is happening is that the prior on f is being overwhelmed by the data as $n \to \infty$. Looking at eq. (7.6) we also see that if $\sigma_n^2 \gg n\lambda_i$ then f_i is effectively zero. This means that we cannot find out about the coefficients of eigenfunctions with small eigenvalues until we get sufficient amounts of data. Ferrari Trecate et al. [1999] demonstrated this by

showing that regression performance of a certain nondegenerate GP could be approximated by taking the first m eigenfunctions, where m was chosen so that $\lambda_m \simeq \sigma_n^2/n$. Of course as more data is obtained then m has to be increased.

Using the fact that $\eta_i = \int \eta(\mathbf{x}')\phi_i(\mathbf{x}')d\mu(\mathbf{x}')$ and defining $\sigma_{\text{eff}}^2 \triangleq \sigma_n^2/n$ we obtain

$$f(\mathbf{x}) \;=\; \sum_{i=1}^{\infty} \frac{\lambda_i \eta_i}{\lambda_i + \sigma_{\text{eff}}^2}\phi_i(\mathbf{x}) \;=\; \int \Big[\sum_{i=1}^{\infty} \frac{\lambda_i \phi_i(\mathbf{x})\phi_i(\mathbf{x}')}{\lambda_i + \sigma_{\text{eff}}^2}\Big]\eta(\mathbf{x}')\, d\mu(\mathbf{x}'). \qquad (7.7)$$

The term in square brackets in eq. (7.7) is the *equivalent kernel* for the smoothing problem; we denote it by $h_n(\mathbf{x}, \mathbf{x}')$. Notice the similarity to the vector-valued weight function $\mathbf{h}(\mathbf{x})$ defined in section 2.6. The difference is that there the prediction was obtained as a linear combination of a finite number of observations y_i with weights given by $h_i(\mathbf{x})$ while here we have a noisy function $y(\mathbf{x})$ instead, with $\bar{f}(\mathbf{x}') = \int h_n(\mathbf{x}, \mathbf{x}')y(\mathbf{x})d\mu(\mathbf{x})$. Notice that in the limit $n \to \infty$ (so that $\sigma_{\text{eff}}^2 \to 0$) the equivalent kernel tends towards the delta function.

equivalent kernel

The form of the equivalent kernel given in eq. (7.7) is not very useful in practice as it requires knowledge of the eigenvalues/functions for the combination of k and μ. However, in the case of stationary kernels we can use Fourier methods to compute the equivalent kernel. Consider the functional

$$J_\rho[f] \;=\; \frac{\rho}{2\sigma_n^2}\int (y(\mathbf{x}) - f(\mathbf{x}))^2\, d\mathbf{x} + \frac{1}{2}\|f\|_{\mathcal{H}}^2, \qquad (7.8)$$

where ρ has dimensions of the number of observations per unit of $\mathbf{x}$-space (length/area/volume etc. as appropriate). Using a derivation similar to eq. (7.6) we obtain

$$\tilde{h}(\mathbf{s}) = \frac{S_f(\mathbf{s})}{S_f(\mathbf{s}) + \sigma_n^2/\rho} = \frac{1}{1 + S_f^{-1}(\mathbf{s})\sigma_n^2/\rho}, \qquad (7.9)$$

where $S_f(\mathbf{s})$ is the power spectrum of the kernel k. The term σ_n^2/ρ corresponds to the power spectrum of a white noise process, as the delta function covariance function of white noise corresponds to a constant in the Fourier domain. This analysis is known as Wiener filtering; see, e.g. Papoulis [1991, sec. 14-1]. Equation (7.9) is the same as eq. (7.6) except that the discrete eigenspectrum has been replaced by a continuous one.

Wiener filtering

As can be observed in Figure 2.6, the equivalent kernel essentially gives a weighting to the observations locally around $\mathbf{x}$. Thus identifying ρ with $np(\mathbf{x})$ we can obtain an approximation to the equivalent kernel for stationary kernels when the width of the kernel is smaller than the length-scale of variations in $p(\mathbf{x})$. This form of analysis was used by Silverman [1984] for splines in one dimension.

7.1.1 Some Specific Examples of Equivalent Kernels

We first consider the OU process in 1-d. This has $k(r) = \exp(-\alpha|r|)$ (setting $\alpha = 1/\ell$ relative to our previous notation and $r = x - x'$), and power spectrum

$S(s) = 2\alpha/(4\pi^2 s^2 + \alpha^2)$. Let $v_n \triangleq \sigma_n^2/\rho$. Using eq. (7.9) we obtain

$$\tilde{h}(s) \;=\; \frac{2\alpha}{v_n(4\pi^2 s^2 + \beta^2)}, \tag{7.10}$$

where $\beta^2 = \alpha^2 + 2\alpha/v_n$. This again has the form of Fourier transform of an OU covariance function[1] and can be inverted to obtain $h(r) = \frac{\alpha}{v_n\beta} e^{-\beta|r|}$. In particular notice that as n increases (and thus v_n decreases) the inverse length-scale β of $h(r)$ increases; asymptotically $\beta \sim n^{1/2}$ for large n. This shows that the width of equivalent kernel for the OU covariance function will scale as $n^{-1/2}$ asymptotically. Similarly the width will scale as $p(\mathbf{x})^{-1/2}$ asymptotically.

A similar analysis can be carried out for the AR(2) Gaussian process in 1-d (see section B.2) which has a power spectrum $\propto (4\pi^2 s^2 + \alpha^2)^{-2}$ (i.e. it is in the Matérn class with $\nu = 3/2$). In this case we can show (using the Fourier relationships given by Papoulis [1991, p. 326]) that the width of the equivalent kernel scales as $n^{-1/4}$ asymptotically.

Analysis of the equivalent kernel has also been carried out for spline models. Silverman [1984] gives the explicit form of the equivalent kernel in the case of a one-dimensional cubic spline (corresponding to the regularizer $\|Pf\|^2 = \int (f'')^2 dx$). Thomas-Agnan [1996] gives a general expression for the equivalent kernel for the spline regularizer $\|Pf\|^2 = \int (f^{(m)})^2 dx$ in one dimension and also analyzes end-effects if the domain of interest is a bounded open interval. For the regularizer $\|Pf\|^2 = \int (\nabla^2 f)^2 d\mathbf{x}$ in two dimensions, the equivalent kernel is given in terms of the Kelvin function kei (Poggio et al. 1985, Stein 1991).

Silverman [1984] has also shown that for splines of order m in 1-d (corresponding to a roughness penalty of $\int (f^{(m)})^2 \, dx$) the width of the equivalent kernel will scale as $n^{-1/2m}$ asymptotically. In fact it can be shown that this is true for splines in $D > 1$ dimensions too, see exercise 7.7.1.

Another interesting case to consider is the squared exponential kernel, where $S(\mathbf{s}) = (2\pi\ell^2)^{D/2} \exp(-2\pi^2\ell^2|\mathbf{s}|^2)$. Thus

$$\tilde{h}_{\mathrm{SE}}(\mathbf{s}) = \frac{1}{1 + b\exp(2\pi^2\ell^2|\mathbf{s}|^2)}, \tag{7.11}$$

where $b = \sigma_n^2/\rho(2\pi\ell^2)^{D/2}$. We are unaware of an exact result in this case, but the following approximation due to Sollich and Williams [2005] is simple but effective. For large ρ (i.e. large n) b will be small. Thus for small $s = |\mathbf{s}|$ we have that $\tilde{h}_{\mathrm{SE}} \simeq 1$, but for large s it is approximately 0. The change takes place around the point s_c where $b\exp(2\pi^2\ell^2 s_c^2) = 1$, i.e. $s_c^2 = \log(1/b)/2\pi^2\ell^2$. As $\exp(2\pi^2\ell^2 s^2)$ grows quickly with s, the transition of $\tilde{h}_{\mathrm{SE}}$ between 1 and 0 can be expected to be rapid, and thus be well-approximated by a step function. By using the standard result for the Fourier transform of the step function we obtain

$$h_{\mathrm{SE}}(x) = 2s_c \mathrm{sinc}(2\pi s_c x) \tag{7.12}$$

[1] The fact that $\tilde{h}(s)$ has the same form as $S_f(s)$ is particular to the OU covariance function and is not generally the case.

for $D = 1$, where $\mathrm{sinc}(z) = \sin(z)/z$. A similar calculation in $D > 1$ using eq. (4.7) gives

$$h_{\mathrm{SE}}(r) = \left(\frac{s_c}{r}\right)^{D/2} J_{D/2}(2\pi s_c r). \tag{7.13}$$

Notice that s_c scales as $(\log(n))^{1/2}$ so that the width of the equivalent kernel will decay very slowly as n increases. Notice that the plots in Figure 2.6 show the sinc-type shape, although the sidelobes are not quite as large as would be predicted by the sinc curve (because the transition is smoother than a step function in Fourier space so there is less "ringing").

7.2 Asymptotic Analysis *

In this section we consider two asymptotic properties of Gaussian processes, consistency and equivalence/orthogonality.

7.2.1 Consistency

In section 7.1 we have analyzed the asymptotics of GP regression and have seen how the minimizer of the functional eq. (7.4) converges to the regression function as $n \to \infty$. We now broaden the focus by considering loss functions other than squared loss, and the case where we work directly with eq. (7.1) rather than the smoothed version eq. (7.4).

The set up is as follows: Let $\mathcal{L}(\cdot,\cdot)$ be a pointwise loss function. Consider a procedure that takes training data $\mathcal{D}$ and this loss function, and returns a function $f_{\mathcal{D}}(\mathbf{x})$. For a measurable function f, the risk (expected loss) is defined as

$$R_{\mathcal{L}}(f) \;=\; \int \mathcal{L}(y, f(\mathbf{x}))\, d\mu(\mathbf{x}, y). \tag{7.14}$$

Let $f^*_{\mathcal{L}}$ denote the function that minimizes this risk. For squared loss $f^*_{\mathcal{L}}(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$. For 0/1 loss with classification problems, we choose $f^*_{\mathcal{L}}(\mathbf{x})$ to be the class c at $\mathbf{x}$ such that $p(\mathcal{C}_c|\mathbf{x}) > p(\mathcal{C}_j|\mathbf{x})$ for all $j \neq c$ (breaking ties arbitrarily).

Definition 7.1 *We will say that a procedure that returns $f_{\mathcal{D}}$ is* consistent *for a given measure $\mu(\mathbf{x}, y)$ and loss function $\mathcal{L}$ if* consistency

$$R_{\mathcal{L}}(f_{\mathcal{D}}) \;\to\; R_{\mathcal{L}}(f^*_{\mathcal{L}}) \qquad \text{as } n \to \infty, \tag{7.15}$$

where convergence is assessed in a suitable manner, e.g. in probability. If $f_{\mathcal{D}}(\mathbf{x})$ is consistent for all Borel probability measures $\mu(\mathbf{x}, y)$ then it is said to be universally consistent. □

A simple example of a consistent procedure is the kernel regression method. As described in section 2.6 one obtains a prediction at test point $\mathbf{x}_*$ by computing $\hat{f}(\mathbf{x}_*) = \sum_{i=1}^n w_i y_i$ where $w_i = \kappa_i / \sum_{j=1}^n \kappa_j$ (the Nadaraya-Watson estimator). Let h be the width of the kernel κ and D be the dimension of the input

space. It can be shown that under suitable regularity conditions if $h \to 0$ and $nh^D \to \infty$ as $n \to \infty$ then the procedure is consistent; see e.g. [Györfi et al., 2002, Theorem 5.1] for the regression case with squared loss and Devroye et al. [1996, Theorem 10.1] for the classification case using 0/1 loss. An intuitive understanding of this result can be obtained by noting that $h \to 0$ means that only datapoints very close to $\mathbf{x}_*$ will contribute to the prediction (eliminating bias), while the condition $nh^D \to \infty$ means that a large number of datapoints will contribute to the prediction (eliminating noise/variance).

It will first be useful to consider why we might hope that GPR and GPC should be universally consistent. As discussed in section 7.1, the key property is that a non-degenerate kernel will have an infinite number of eigenfunctions forming an orthonormal set. Thus from generalized Fourier analysis a linear combination of eigenfunctions $\sum_{i=1}^{\infty} c_i \phi_i(\mathbf{x})$ should be able to represent a sufficiently well-behaved target function f_L^*. However, we have to estimate the infinite number of coefficients $\{c_i\}$ from the noisy observations. This makes it clear that we are playing a game involving infinities which needs to be played with care, and there are some results [Diaconis and Freedman, 1986, Freedman, 1999, Grünwald and Langford, 2004] which show that in certain circumstances Bayesian inference in infinite-dimensional objects can be inconsistent.

However, there are some positive recent results on the consistency of GPR and GPC. Choudhuri et al. [2005] show that for the binary classification case under certain assumptions GPC is consistent. The assumptions include smoothness on the mean and covariance function of the GP, smoothness on $\mathbb{E}[y|\mathbf{x}]$ and an assumption that the domain is a bounded subset of $\mathbb{R}^D$. Their result holds for the class of response functions which are c.d.f.s of a unimodal symmetric density; this includes the probit and logistic functions.

For GPR, Choi and Schervish [2004] show that for a one-dimensional input space of finite length under certain assumptions consistency holds. Here the assumptions again include smoothness of the mean and covariance function of the GP and smoothness of $\mathbb{E}[y|\mathbf{x}]$. An additional assumption is that the noise has a normal or Laplacian distribution (with an unknown variance which is inferred).

There are also some consistency results relating to the functional

$$J_{\lambda_n}[f] \;=\; \frac{\lambda_n}{2}\|f\|_{\mathcal{H}}^2 + \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\big(y_i, f(\mathbf{x}_i)\big), \tag{7.16}$$

where $\lambda_n \to 0$ as $n \to \infty$. Note that to agree with our previous formulations we would set $\lambda_n = 1/n$, but other decay rates on λ_n are often considered.

In the splines literature, Cox [1984] showed that for regression problems using the regularizer $\|f\|_m^2 = \sum_{k=0}^{m} \|O^k f\|^2$ (using the definitions in eq. (6.10)) consistency can be obtained under certain technical conditions. Cox and O'Sullivan [1990] considered a wide range of problems (including regression problems with squared loss and classification using logistic loss) where the solution is obtained by minimizing the regularized risk using a spline smoothness term. They showed that if $f_{\mathcal{L}}^* \in \mathcal{H}$ (where $\mathcal{H}$ is the RKHS corresponding to the spline

regularizer) then as $n \to \infty$ and $\lambda_n \to 0$ at an appropriate rate, one gets convergence of $f_{\mathcal{D}}$ to $f^*_{\mathcal{L}}$.

More recently, Zhang [2004, Theorem 4.4] has shown that for the classification problem with a number of different loss functions (including logistic loss, hinge loss and quadratic loss) and for general RKHSs with a nondegenerate kernel, that if $\lambda_n \to 0$, $\lambda_n n \to \infty$ and $\mu(\mathbf{x}, y)$ is sufficiently regular then the classification error of $f_{\mathcal{D}}$ will converge to the Bayes optimal error in probability as $n \to \infty$. Similar results have also been obtained by Steinwart [2005] with various rates on the decay of λ_n depending on the smoothness of the kernel. Bartlett et al. [2003] have characterized the loss functions that lead to universal consistency.

Above we have focussed on regression and classification problems. However, similar analyses can also be given for other problems such as density estimation and deconvolution; see Wahba [1990, chs. 8, 9] for references. Also we have discussed consistency using a fixed decay rate for λ_n. However, it is also possible to analyze the asymptotics of methods where λ_n is set in a data-dependent way, e.g. by cross-validation;[2] see Wahba [1990, sec. 4.5] and references therein for further details.

Consistency is evidently a desirable property of supervised learning procedures. However, it is an asymptotic property that does not say very much about how a given prediction procedure will perform on a particular problem with a given dataset. For instance, note that we only required rather general properties of the kernel function (e.g. non-degeneracy) for some of the consistency results. However, the choice of the kernel can make a huge difference to how a procedure performs in practice. Some analyses related to this issue are given in section 7.3.

7.2.2 Equivalence and Orthogonality

The presentation in this section is based mainly on Stein [1999, ch. 4]. For two probability measures μ_0 and μ_1 defined on a measurable space $(\Omega, \mathcal{F})$,[3] μ_0 is said to be *absolutely continuous* w.r.t. μ_1 if for all $A \in \mathcal{F}$, $\mu_1(A) = 0$ implies $\mu_0(A) = 0$. If μ_0 is absolutely continuous w.r.t. μ_1 and μ_1 is absolutely continuous w.r.t. μ_0 the two measures are said to be *equivalent*, written $\mu_0 \equiv \mu_1$. μ_0 and μ_1 are said to be *orthogonal*, written $\mu_0 \perp \mu_1$, if there exists an $A \in \mathcal{F}$ such that $\mu_0(A) = 1$ and $\mu_1(A) = 0$. (Note that in this case we have $\mu_0(A^c) = 0$ and $\mu_1(A^c) = 1$, where A^c is the complement of A.) The dichotomy theorem for Gaussian processes (due to Hajek [1958] and, independently, Feldman [1958]) states that two Gaussian processes are either equivalent or orthogonal.

Equivalence and orthogonality for Gaussian measures μ_0, μ_1 with corresponding probability densities p_0, p_1, can be characterized in terms of the

[2] Cross validation is discussed in section 5.3.

[3] See section A.7 for background on measurable spaces.

symmetrized Kullback-Leibler divergence $\mathrm{KL}_{\mathrm{sym}}$ between them, given by

$$\mathrm{KL}_{\mathrm{sym}}(p_0, p_1) = \int (p_0(\mathbf{f}) - p_1(\mathbf{f})) \log \frac{p_0(\mathbf{f})}{p_1(\mathbf{f})} d\mathbf{f}. \tag{7.17}$$

The measures are equivalent if $\mathrm{KL}_{\mathrm{sym}} < \infty$ and orthogonal otherwise. For two finite-dimensional Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_0, K_0)$ and $\mathcal{N}(\boldsymbol{\mu}_1, K_1)$ we have [Kullback, 1959, sec. 9.1]

$$\begin{aligned} \mathrm{KL}_{\mathrm{sym}} &= \tfrac{1}{2} \operatorname{tr}(K_0 - K_1)(K_1^{-1} - K_0^{-1}) \\ &\quad + \tfrac{1}{2} \operatorname{tr}(K_1^{-1} + K_0^{-1})(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top. \end{aligned} \tag{7.18}$$

This expression can be simplified considerably by simultaneously diagonalizing K_0 and K_1. Two finite-dimensional Gaussian distributions are equivalent if the null spaces of their covariance matrices coincide, and are orthogonal otherwise.

Things can get more interesting if we consider infinite-dimensional distributions, i.e. Gaussian processes. Consider some closed subset $R \in \mathbb{R}^D$. Choose some finite number n of $\mathbf{x}$-points in R and let $\mathbf{f} = (f_1, \ldots, f_n)^\top$ denote the values corresponding to these inputs. We consider the $\mathrm{KL}_{\mathrm{sym}}$-divergence as above, but in the limit $n \to \infty$. $\mathrm{KL}_{\mathrm{sym}}$ can now diverge if the rates of decay of the eigenvalues of the two processes are not the same. For example, consider zero-mean periodic processes with period 1 where the eigenvalue λ_j^i indicates the amount of power in the sin/cos terms of frequency $2\pi j$ for process $i = 0, 1$. Then using eq. (7.18) we have

$$\mathrm{KL}_{\mathrm{sym}} = \frac{(\lambda_0^0 - \lambda_0^1)^2}{\lambda_0^0 \lambda_0^1} + 2 \sum_{j=1}^{\infty} \frac{(\lambda_j^0 - \lambda_j^1)^2}{\lambda_j^0 \lambda_j^1} \tag{7.19}$$

(see also [Stein, 1999, p. 119]). Some corresponding results for the equivalence or orthogonality of non-periodic Gaussian processes are given in Stein [1999, pp. 119-122]. Stein (p. 109) gives an example of two equivalent Gaussian processes on $\mathbb{R}$, those with covariance functions $\exp(-r)$ and $1/2 \exp(-2r)$. (It is easy to check that for large s these have the same power spectrum.)

We now turn to the consequences of equivalence for the model selection problem. Suppose that we know that either $\mathcal{GP}_0$ or $\mathcal{GP}_1$ is the correct model. Then if $\mathcal{GP}_0 \equiv \mathcal{GP}_1$ then it is not possible to determine which model is correct with probability 1. However, under a Bayesian setting all this means is if we have prior probabilities π_0 and $\pi_1 = 1 - \pi_0$ on these two hypotheses, then after observing some data $\mathcal{D}$ the posterior probabilities $p(\mathcal{GP}_i|\mathcal{D})$ (for $i = 0, 1$) will not be 0 or 1, but could be heavily skewed to one model or the other.

The other important observation is to consider the predictions made by $\mathcal{GP}_0$ or $\mathcal{GP}_1$. Consider the case where $\mathcal{GP}_0$ is the correct model and $\mathcal{GP}_1 \equiv \mathcal{GP}_0$. Then Stein [1999, sec. 4.3] shows that the predictions of $\mathcal{GP}_1$ are asymptotically optimal, in the sense that the expected relative prediction error between $\mathcal{GP}_1$ and $\mathcal{GP}_0$ tends to 0 as $n \to \infty$ under some technical conditions. Stein's Corollary 9 (p. 132) shows that this conclusion remains true under additive noise if the un-noisy GPs are equivalent. One caveat about equivalence is although the predictions of $\mathcal{GP}_1$ are asymptotically optimal when $\mathcal{GP}_0$ is the correct model and $\mathcal{GP}_0 \equiv \mathcal{GP}_1$, one would see differing predictions for finite n.

C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. www.GaussianProcess.org/gpml

7.3 Average-case Learning Curves *

In section 7.2 we have discussed the asymptotic properties of Gaussian process predictors and related methods. In this section we will say more about the speed of convergence under certain specific assumptions. Our goal will be to obtain a *learning curve* describing the generalization error as a function of the training set size n. This is an average-case analysis, averaging over the choice of target functions (drawn from a GP) and over the $\mathbf{x}$ locations of the training points.

In more detail, we first consider a target function f drawn from a Gaussian process. n locations are chosen to make observations at, giving rise to the training set $\mathcal{D} = (X, \mathbf{y})$. The y_is are (possibly) noisy observations of the underlying function f. Given a loss function $\mathcal{L}(\cdot, \cdot)$ which measures the difference between the prediction for f and f itself, we obtain an estimator $f_{\mathcal{D}}$ for f. Below we will use the squared loss, so that the posterior mean $\bar{f}_{\mathcal{D}}(\mathbf{x})$ is the estimator. Then the *generalization error* (given f and $\mathcal{D}$) is given by generalization error

$$E^g_{\mathcal{D}}(f) \;=\; \int \mathcal{L}(f(\mathbf{x}_*), \bar{f}_{\mathcal{D}}(\mathbf{x}_*)) p(\mathbf{x}_*)\, d\mathbf{x}_*. \tag{7.20}$$

As this is an expected loss it is technically a risk, but the term generalization error is commonly used.

$E^g_{\mathcal{D}}(f)$ depends on both the choice of f and on X. (Note that $\mathbf{y}$ depends on the choice of f, and also on the noise, if present.) The first level of averaging we consider is over functions f drawn from a GP prior, to obtain

$$E^g(X) \;=\; \int E^g_{\mathcal{D}}(f) p(f)\, df. \tag{7.21}$$

It will turn out that for regression problems with Gaussian process priors and predictors this average can be readily calculated. The second level of averaging assumes that the $\mathbf{x}$-locations of the training set are drawn i.i.d. from $p(\mathbf{x})$ to give

$$E^g(n) \;=\; \int E^g(X) p(\mathbf{x}_1) \ldots p(\mathbf{x}_n)\, dx_1 \ldots\, dx_n. \tag{7.22}$$

A plot of $E^g(n)$ against n is known as a *learning curve*. learning curve

Rather than averaging over X, an alternative is to minimize $E^g(X)$ w.r.t. X. This gives rise to the *optimal experimental design* problem. We will not say more about this problem here, but it has been subject to a large amount of investigation. An early paper on this subject is by Ylvisaker [1975]. These questions have been addressed both in the statistical literature and in theoretical numerical analysis; for the latter area the book by Ritter [2000] provides a useful overview.

We now proceed to develop the average-case analysis further for the specific case of GP predictors and GP priors for the regression case using squared loss. Let f be drawn from a zero-mean GP with covariance function k_0 and noise level σ_0^2. Similarly the predictor assumes a zero-mean process, but covariance

function k_1 and noise level σ_1^2. At a particular test location $\mathbf{x}_*$, averaging over f we have

$$\begin{aligned}
&\mathbb{E}[(f(\mathbf{x}_*) - \mathbf{k}_1(\mathbf{x}_*)^\top K_{1,y}^{-1}\mathbf{y})^2] \qquad (7.23)\\
&\quad = \mathbb{E}[f^2(\mathbf{x}_*)] - 2\mathbf{k}_1(\mathbf{x}_*)^\top K_{1,y}^{-1}\mathbb{E}[f(\mathbf{x}_*)\mathbf{y}] + \mathbf{k}_1(\mathbf{x}_*)^\top K_{1,y}^{-1}\mathbb{E}[\mathbf{y}\mathbf{y}^\top]K_{1,y}^{-1}\mathbf{k}_1(\mathbf{x}_*)\\
&\quad = k_0(\mathbf{x}_*,\mathbf{x}_*) - 2\mathbf{k}_1(\mathbf{x}_*)^\top K_{1,y}^{-1}\mathbf{k}_0(\mathbf{x}_*) + \mathbf{k}_1(\mathbf{x}_*)^\top K_{1,y}^{-1}K_{0,y}K_{1,y}^{-1}\mathbf{k}_1(\mathbf{x}_*)
\end{aligned}$$

where $K_{i,y} = K_{i,f} + \sigma_i^2$ for $i = 0,\ 1$, i.e. the covariance matrix including the assumed noise. If $k_1 = k_0$ so that the predictor is correctly specified then the above expression reduces to $k_0(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_0(\mathbf{x}_*)^\top K_{0,y}^{-1}\mathbf{k}_0(\mathbf{x}_*)$, the predictive variance of the GP.

Averaging the error over $p(\mathbf{x}_*)$ we obtain

$$\begin{aligned}
E^g(X) &= \int \mathbb{E}[(f(\mathbf{x}_*) - \mathbf{k}_1(\mathbf{x}_*)^\top K_{1,y}^{-1}\mathbf{y})^2]p(\mathbf{x}_*)\,d\mathbf{x}_* \qquad (7.24)\\
&= \int k_0(\mathbf{x}_*,\mathbf{x}_*)p(\mathbf{x}_*)\,d\mathbf{x}_* - 2\,\mathrm{tr}\left(K_{1,y}^{-1}\int \mathbf{k}_0(\mathbf{x}_*)\mathbf{k}_1(\mathbf{x}_*)^\top p(\mathbf{x}_*)\,d\mathbf{x}_*\right)\\
&\quad + \mathrm{tr}\left(K_{1,y}^{-1}K_{0,y}K_{1,y}^{-1}\int \mathbf{k}_1(\mathbf{x}_*)\mathbf{k}_1(\mathbf{x})^\top p(\mathbf{x}_*)\,d\mathbf{x}_*\right).
\end{aligned}$$

For some choices of $p(\mathbf{x}_*)$ and covariance functions these integrals will be analytically tractable, reducing the computation of $E^g(X)$ to a $n \times n$ matrix computation.

To obtain $E^g(n)$ we need to perform a final level of averaging over X. In general this is difficult even if $E^g(X)$ can be computed exactly, but it is sometimes possible, e.g. for the noise-free OU process on the real line, see section 7.6.

The form of $E^g(X)$ can be simplified considerably if we express the covariance functions in terms of their eigenfunction expansions. In the case that $k_0 = k_1$ we use the definition $k(\mathbf{x},\mathbf{x}') = \sum_i \lambda_i\phi_i(\mathbf{x})\phi_i(\mathbf{x}')$ and $\int k(\mathbf{x},\mathbf{x}')\phi_i(\mathbf{x})p(\mathbf{x})\,d\mathbf{x} = \lambda_i\phi_i(\mathbf{x}')$. Let Λ be a diagonal matrix of the eigenvalues and Φ be the $N \times n$ design matrix, as defined in section 2.1.2. Then from eq. (7.24) we obtain

$$\begin{aligned}
E^g(X) &= \mathrm{tr}(\Lambda) - \mathrm{tr}((\sigma_n^2 I + \Phi^\top\Lambda\Phi)^{-1}\Phi^\top\Lambda^2\Phi)\\
&= \mathrm{tr}(\Lambda^{-1} + \sigma_n^{-2}\Phi\Phi^\top)^{-1},
\end{aligned} \qquad (7.25)$$

where the second line follows through the use of the matrix inversion lemma eq. (A.9) (or directly if we use eq. (2.11)), as shown in Sollich [1999] or Opper and Vivarelli [1999]. Using the fact that $\mathbb{E}_X[\Phi\Phi^\top] = nI$, a naïve approximation would replace $\Phi\Phi^\top$ inside the trace with its expectation; in fact Opper and Vivarelli [1999] showed that this gives a lower bound, so that

$$E^g(n) \geq \mathrm{tr}(\Lambda^{-1} + n\sigma_n^{-2}I)^{-1} = \sigma^2\sum_{i=1}^{N}\frac{\lambda_i}{\sigma_n^2 + n\lambda_i}. \qquad (7.26)$$

Examining the asymptotics of eq. (7.26), we see that for each eigenvalue where $\lambda_i \gg \sigma_n^2/n$ we add σ_n^2/n onto the bound on the generalization error. As we saw

in section 7.1, more eigenfunctions "come into play" as n increases, so the rate of decay of $E^g(n)$ is slower than $1/n$. Sollich [1999] derives a number of more accurate approximations to the learning curve than eq. (7.26).

For the noiseless case with $k_1 = k_0$, there is a simple lower bound $E^g(n) \geq \sum_{i=n+1}^{\infty} \lambda_i$ due to Micchelli and Wahba [1981]. This bound is obtained by demonstrating that the optimal n pieces of information are the projections of the random function f onto the first n eigenfunctions. As observations which simply consist of function evaluations will not in general provide such information this is a lower bound. Plaskota [1996] generalized this result to give a bound on the learning curve if the observations are noisy.

Some asymptotic results for the learning curves are known. For example, in Ritter [2000, sec. V.2] covariance functions obeying Sacks-Ylvisaker conditions[4] of order r in 1-d are considered. He shows that for an optimal sampling of the input space the generalization error goes as $\mathcal{O}(n^{-(2r+1)/(2r+2)})$ for the noisy problem. Similar rates can also be found in Sollich [2002] for random designs. For the noise-free case Ritter [2000, p. 103] gives the rate as $\mathcal{O}(n^{-(2r+1)})$.

One can examine the learning curve not only asymptotically but also for small n, where typically the curve has a roughly linear decrease with n. Williams and Vivarelli [2000] explained this behaviour by observing that the introduction of a datapoint $\mathbf{x}_1$ reduces the variance locally around $\mathbf{x}_1$ (assuming a stationary covariance function). The addition of another datapoint at $\mathbf{x}_2$ will also create a "hole" there, and so on. With only a small number of datapoints it is likely that these holes will be far apart so their contributions will add, thus explaining the initial linear trend.

Sollich [2002] has also investigated the mismatched case where $k_0 \neq k_1$. This can give rise to a rich variety of behaviours in the learning curves, including plateaux. Stein [1999, chs. 3, 4] has also carried out some analysis of the mismatched case.

Although we have focused on GP regression with squared loss, we note that Malzahn and Opper [2002] have developed more general techniques that can be used to analyze learning curves for other situations such as GP classification.

7.4 PAC-Bayesian Analysis *

In section 7.3 we gave an *average-case* analysis of generalization, taking the average with respect to a GP prior over functions. In this section we present a different kind of analysis within the *probably approximately correct* (PAC) framework due to Valiant [1984]. Seeger [2002; 2003] has presented a PAC-Bayesian analysis of generalization in Gaussian process classifiers and we get to this in a number of stages; we first present an introduction to the PAC framework (section 7.4.1), then describe the PAC-Bayesian approach (section

PAC

[4] Roughly speaking, a stochastic process which possesses r MS derivatives but not $r+1$ is said to satisfy Sacks-Ylvisaker conditions of order r; in 1-d this gives rise to a spectrum $\lambda_i \propto i^{-(2r+2)}$ asymptotically. The OU process obeys Sacks-Ylvisaker conditions of order 0.

7.4.2) and then finally the application to GP classification (section 7.4.3). Our presentation is based mainly on Seeger [2003].

7.4.1 The PAC Framework

Consider a fixed measure $\mu(\mathbf{x}, y)$. Given a loss function $\mathcal{L}$ there exists a function $\eta(\mathbf{x})$ which minimizes the expected risk. By running a learning algorithm on a data set $\mathcal{D}$ of size n drawn i.i.d. from $\mu(\mathbf{x}, y)$ we obtain an estimate $f_\mathcal{D}$ of η which attains an expected risk $R_\mathcal{L}(f_\mathcal{D})$. We are not able to evaluate $R_\mathcal{L}(f_\mathcal{D})$ as we do not know μ. However, we do have access to the empirical distribution of the training set $\hat{\mu}(\mathbf{x}, y) = \frac{1}{n}\sum_i \delta(\mathbf{x}-\mathbf{x}_i)\delta(y-y_i)$ and can compute the empirical risk $\hat{R}_\mathcal{L}(f_\mathcal{D}) = \frac{1}{n}\sum_i \mathcal{L}(y_i, f_\mathcal{D}(\mathbf{x}_i))$. Because the training set had been used to compute $f_\mathcal{D}$ we would expect $\hat{R}_\mathcal{L}(f_\mathcal{D})$ to underestimate $R_\mathcal{L}(f_\mathcal{D})$,[5] and the aim of the PAC analysis is to provide a bound on $R_\mathcal{L}(f_\mathcal{D})$ based on $\hat{R}_\mathcal{L}(f_\mathcal{D})$.

A PAC bound has the following format

$$p_\mathcal{D}\{R_\mathcal{L}(f_\mathcal{D}) \leq \hat{R}_\mathcal{L}(f_\mathcal{D}) + \mathrm{gap}(f_\mathcal{D}, \mathcal{D}, \delta)\} \;\geq\; 1-\delta, \tag{7.27}$$

where $p_\mathcal{D}$ denotes the probability distribution of datasets drawn i.i.d. from $\mu(\mathbf{x}, y)$, and $\delta \in (0, 1)$ is called the confidence parameter. The bound states that, averaged over draws of the dataset $\mathcal{D}$ from $\mu(\mathbf{x}, y)$, $R_\mathcal{L}(f_\mathcal{D})$ does not exceed the sum of $\hat{R}_\mathcal{L}(f_\mathcal{D})$ and the gap term with probability of at least $1-\delta$. The δ accounts for the "probably" in PAC, and the "approximately" derives from the fact that the gap term is positive for all n. It is important to note that PAC analyses are *distribution-free*, i.e. eq. (7.27) must hold for any measure μ.

There are two kinds of PAC bounds, depending on whether $\mathrm{gap}(f_\mathcal{D}, \mathcal{D}, \delta)$ actually depends on the particular sample $\mathcal{D}$ (rather than on simple statistics like n). Bounds that do depend on $\mathcal{D}$ are called data dependent, and those that do not are called data independent. The PAC-Bayesian bounds given below are data dependent.

It is important to understand the interpretation of a PAC bound and to clarify this we first consider a simpler case of statistical inference. We are given a dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ drawn i.i.d. from a distribution $\mu(\mathbf{x})$ that has mean $\mathbf{m}$. An estimate of $\mathbf{m}$ is given by the sample mean $\bar{\mathbf{x}} = \sum_i \mathbf{x}_i/n$. Under certain assumptions we can obtain (or put bounds on) the *sampling distribution* $p(\bar{\mathbf{x}}|\mathbf{m})$ which relates to the choice of dataset $\mathcal{D}$. However, if we wish to perform probabilistic inference for $\mathbf{m}$ we need to combine $p(\bar{\mathbf{x}}|\mathbf{m})$ with a prior distribution $p(\mathbf{m})$ and use Bayes' theorem to obtain the posterior.[6] The situation is similar (although somewhat more complex) for PAC bounds as these concern the sampling distribution of the expected and empirical risks of $f_\mathcal{D}$ w.r.t. $\mathcal{D}$.

[5] It is also possible to consider PAC analyses of other empirical quantities such as the cross-validation error (see section 5.3) which do not have this bias.

[6] In introductory treatments of frequentist statistics the logical hiatus of going from the sampling distribution to inference on the parameter of interest is often not well explained.

We might wish to make a conditional statement like

$$p_{\mathcal{D}}\{R_{\mathcal{L}}(f_{\mathcal{D}}) \leq r + \text{gap}(f_{\mathcal{D}}, \mathcal{D}, \delta) | \hat{R}_{\mathcal{L}}(f_{\mathcal{D}}) = r\} \geq 1 - \delta, \tag{7.28}$$

where r is a small value, but such a statement cannot be inferred directly from the PAC bound. This is because the gap might be heavily anti-correlated with $\hat{R}_{\mathcal{L}}(f_{\mathcal{D}})$ so that the gap is large when the empirical risk is small.

PAC bounds are sometimes used to carry out model selection—given a learning machine which depends on a (discrete or continuous) parameter vector $\boldsymbol{\theta}$, one can seek to minimize the generalization bound as a function of $\boldsymbol{\theta}$. However, this procedure may not be well-justified if the generalization bounds are loose. Let the *slack* denote the difference between the value of the bound and the generalization error. The danger of choosing $\boldsymbol{\theta}$ to minimize the bound is that if the slack depends on $\boldsymbol{\theta}$ then the value of $\boldsymbol{\theta}$ that minimizes the bound may be very different from the value of $\boldsymbol{\theta}$ that minimizes the generalization error. See Seeger [2003, sec. 2.2.4] for further discussion.

7.4.2 PAC-Bayesian Analysis

We now consider a Bayesian set up, with a prior distribution $p(\mathbf{w})$ over the parameters $\mathbf{w}$, and a "posterior" distribution $q(\mathbf{w})$. (Strictly speaking the analysis does not require $q(\mathbf{w})$ to be the posterior distribution, just some other distribution, but in practice we will consider q to be an (approximate) posterior distribution.) We also limit our discussion to binary classification with labels $\{-1, 1\}$, although more general cases can be considered, see Seeger [2003, sec. 3.2.2].

The predictive distribution for f_* at a test point $\mathbf{x}_*$ given $q(\mathbf{w})$ is $q(f_*|\mathbf{x}_*) = \int q(f_*|\mathbf{w}, \mathbf{x}_*) q(\mathbf{w}) d\mathbf{w}$, and the *predictive classifier* outputs $\text{sgn}(q(f_*|\mathbf{x}_*) - 1/2)$. The *Gibbs classifier* has also been studied in learning theory; given a test point $\mathbf{x}_*$ one draws a sample $\tilde{\mathbf{w}}$ from $q(\mathbf{w})$ and predicts the label using $\text{sgn}(f(\mathbf{x}_*; \tilde{\mathbf{w}}))$. The main reason for introducing the Gibbs classifier here is that the PAC-Bayesian theorems given below apply to Gibbs classifiers.

predictive classifier
Gibbs classifier

For a given parameter vector $\mathbf{w}$ giving rise to a classifier $c(\mathbf{x}; \mathbf{w})$, the expected risk and empirical risk are given by

$$R_{\mathcal{L}}(\mathbf{w}) = \int \mathcal{L}(y, c(\mathbf{x}; \mathbf{w}))\, d\mu(\mathbf{x}, y), \quad \hat{R}_{\mathcal{L}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, c(\mathbf{x}_i; \mathbf{w})). \tag{7.29}$$

As the Gibbs classifier draws samples from $q(\mathbf{w})$ we consider the averaged risks

$$R_{\mathcal{L}}(q) = \int R_{\mathcal{L}}(\mathbf{w}) q(\mathbf{w})\, d\mathbf{w}, \qquad \hat{R}_{\mathcal{L}}(q) = \int \hat{R}_{\mathcal{L}}(\mathbf{w}) q(\mathbf{w})\, d\mathbf{w}. \tag{7.30}$$

Theorem 7.1 *(McAllester's PAC-Bayesian theorem) For any probability measures p and q over $\mathbf{w}$ and for any bounded loss function $\mathcal{L}$ for which $\mathcal{L}(y, c(\mathbf{x})) \in [0, 1]$ for any classifier c and input $\mathbf{x}$ we have*

McAllester's PAC-Bayesian theorem

$$p_{\mathcal{D}}\Big\{R_{\mathcal{L}}(q) \leq \hat{R}_{\mathcal{L}}(q) + \sqrt{\frac{\text{KL}(q||p) + \log \frac{1}{\delta} + \log n + 2}{2n - 1}} \;\forall\, q\Big\} \geq 1 - \delta. \tag{7.31}$$

□

The proof can be found in McAllester [2003]. The Kullback-Leibler (KL) divergence $\mathrm{KL}(q||p)$ is defined in section A.5. An example of a loss function which obeys the conditions of the theorem is the 0/1 loss.

For the special case of 0/1 loss, Seeger [2002] gives the following tighter bound.

Seeger's PAC-Bayesian theorem

Theorem 7.2 *(Seeger's PAC-Bayesian theorem) For any distribution over* $\mathcal{X} \times \{-1, +1\}$ *and for any probability measures* p *and* q *over* $\mathbf{w}$ *the following bound holds for i.i.d. samples drawn from the data distribution*

$$p_{\mathcal{D}}\Big\{ \mathrm{KL}_{\mathrm{Ber}}(\hat{R}_{\mathcal{L}}(q)||R_{\mathcal{L}}(q)) \leq \frac{1}{n}(\mathrm{KL}(q||p) + \log\frac{n+1}{\delta}) \;\forall\, q \Big\} \;\geq\; 1-\delta. \quad (7.32)$$

□

Here $\mathrm{KL}_{\mathrm{Ber}}(\cdot||\cdot)$ is the KL divergence between two Bernoulli distributions (defined in eq. (A.22)). Thus the theorem bounds (with high probability) the KL divergence between $\hat{R}_{\mathcal{L}}(q)$ and $R_{\mathcal{L}}(q)$.

The PAC-Bayesian theorems above refer to a Gibbs classifier. If we are interested in the predictive classifier $\mathrm{sgn}(q(f_*|\mathbf{x}_*) - 1/2)$ then Seeger [2002] shows that if $q(f_*|\mathbf{x}_*)$ is symmetric about its mean then the expected risk of the predictive classifier is less than twice the expected risk of the Gibbs classifier. However, this result is based on a simple bounding argument and in practice one would expect that the predictive classifier will usually give better performance than the Gibbs classifier. Recent work by Meir and Zhang [2003] provides some PAC bounds directly for Bayesian algorithms (like the predictive classifier) whose predictions are made on the basis of a data-dependent posterior distribution.

7.4.3 PAC-Bayesian Analysis of GP Classification

To apply this bound to the Gaussian process case we need to compute the KL divergence $\mathrm{KL}(q||p)$ between the posterior distribution $q(\mathbf{w})$ and the prior distribution $p(\mathbf{w})$. Although this could be considered w.r.t. the weight vector $\mathbf{w}$ in the eigenfunction expansion, in fact it turns out to be more convenient to consider the latent function value $f(\mathbf{x})$ at every possible point in the input space $\mathcal{X}$ as the parameter. We divide this (possibly infinite) vector into two parts, (1) the values corresponding to the training points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, denoted $\mathbf{f}$, and (2) those at the remaining points in $\mathbf{x}$-space (the test points) $\mathbf{f}_*$.

The key observation is that all methods we have described for dealing with GP classification problems produce a posterior approximation $q(\mathbf{f}|\mathbf{y})$ which is defined at the training points. (This is an approximation for Laplace's method and for EP; MCMC methods sample from the exact posterior.) This posterior over $\mathbf{f}$ is then extended to the test points by setting $q(\mathbf{f}, \mathbf{f}_*|\mathbf{y}) = q(\mathbf{f}|\mathbf{y})p(\mathbf{f}_*|\mathbf{f})$. Of course for the prior distribution we have a similar decomposition $p(\mathbf{f}, \mathbf{f}_*) =$

$p(\mathbf{f})p(\mathbf{f}_*|\mathbf{f})$. Thus the KL divergence is given by

$$\begin{aligned} \mathrm{KL}(q||p) &= \int q(\mathbf{f}|\mathbf{y})p(\mathbf{f}_*|\mathbf{f}) \log \frac{q(\mathbf{f}|\mathbf{y})p(\mathbf{f}_*|\mathbf{f})}{p(\mathbf{f})p(\mathbf{f}_*|\mathbf{f})} \, d\mathbf{f} d\mathbf{f}_* \\ &= \int q(\mathbf{f}|\mathbf{y}) \log \frac{q(\mathbf{f}|\mathbf{y})}{p(\mathbf{f})} d\mathbf{f}, \end{aligned} \tag{7.33}$$

as shown e.g. in Seeger [2002]. Notice that this has reduced a rather scary infinite-dimensional integration to a more manageable n-dimensional integration; in the case that $q(\mathbf{f}|\mathbf{y})$ is Gaussian (as for the Laplace and EP approximations), this KL divergence can be computed using eq. (A.23). For the Laplace approximation with $p(\mathbf{f}) = \mathcal{N}(\mathbf{0}, K)$ and $q(\mathbf{f}|\mathbf{y}) = \mathcal{N}(\hat{\mathbf{f}}, A^{-1})$ this gives

$$\mathrm{KL}(q||p) = \tfrac{1}{2}\log|K| + \tfrac{1}{2}\log|A| + \tfrac{1}{2}\operatorname{tr}\left(A^{-1}(K^{-1} - A)\right) + \tfrac{1}{2}\hat{\mathbf{f}}^\top K^{-1}\hat{\mathbf{f}}. \tag{7.34}$$

Seeger [2002] has evaluated the quality of the bound produced by the PAC-Bayesian method for a Laplace GPC on the task of discriminating handwritten 2s and 3s from the MNIST handwritten digits database.[7] He reserved a test set of 1000 examples and used training sets of size 500, 1000, 2000, 5000 and 9000. The classifications were replicated ten times using draws of the training sets from a pool of 12089 examples. We quote example results for $n = 5000$ where the training error was 0.0187 ± 0.0016, the test error was 0.0195 ± 0.0011 and the PAC-Bayesian bound on the generalization error (evaluated for $\delta = 0.01$) was 0.076 ± 0.002. (The $\pm$ figures denote a 95% confidence interval.) The classification results are for the Gibbs classifier; for the predictive classifier the test error rate was 0.0171 ± 0.0016. Thus the generalization error is around 2%, while the PAC bound is 7.6%. Many PAC bounds struggle to predict error rates below 100%(!), so this is an impressive and highly non-trivial result. Further details and experiments can be found in Seeger [2002].

7.5 Comparison with Other Supervised Learning Methods

The focus of this book is on Gaussian process methods for supervised learning. However, there are many other techniques available for supervised learning such as linear regression, logistic regression, decision trees, neural networks, support vector machines, kernel smoothers, k-nearest neighbour classifiers, etc., and we need to consider the relative strengths and weaknesses of these approaches.

Supervised learning is an *inductive* process—given a finite training set we wish to infer a function f that makes predictions for all possible input values. The additional assumptions made by the learning algorithm are known as its *inductive bias* (see e.g. Mitchell [1997, p. 43]). Sometimes these assumptions are explicit, but for other algorithms (e.g. for decision tree induction) they can be rather more implicit.

inductive bias

[7] See http://yann.lecun.com/exdb/mnist.

C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. www.GaussianProcess.org/gpml

However, for all their variety, supervised learning algorithms are based on the idea that similar input patterns will usually give rise to similar outputs (or output distributions), and it is the precise notion of similarity that differentiates the algorithms. For example some algorithms may do feature selection and decide that there are input dimensions that are irrelevant to the predictive task. Some algorithms may construct new features out of those provided and measure similarity in this derived space. As we have seen, many regression techniques can be seen as linear smoothers (see section 2.6) and these techniques vary in the definition of the weight function that is used.

One important distinction between different learning algorithms is how they relate to the question of universal consistency (see section 7.2.1). For example a linear regression model will be inconsistent if the function that minimizes the risk cannot be represented by a linear function of the inputs. In general a model with a finite-dimensional parameter vector will not be universally consistent. Examples of such models are linear regression and logistic regression with a finite-dimensional feature vector, and neural networks with a fixed number of hidden units. In contrast to these *parametric* models we have *non-parametric* models (such as k-nearest neighbour classifiers, kernel smoothers and Gaussian processes and SVMs with nondegenerate kernels) which do not compress the training data into a finite-dimensional parameter vector. An intermediate position is taken by *semi-parametric* models such as neural networks where the number of hidden units k is allowed to increase as n increases. In this case universal consistency results can be obtained [Devroye et al., 1996, ch. 30] under certain technical conditions and growth rates on k.

Although universal consistency is a "good thing", it does not necessarily mean that we should only consider procedures that have this property; for example if on a specific problem we knew that a linear regression model was consistent for that problem then it would be very natural to use it.

neural networks

In the 1980's there was a large surge in interest in artificial neural networks (ANNs), which are feedforward networks consisting of an input layer, followed by one or more layers of non-linear transformations of weighted combinations of the activity from previous layers, and an output layer. One reason for this surge of interest was the use of the backpropagation algorithm for training ANNs. Initial excitement centered around that fact that training non-linear networks was possible, but later the focus came onto the generalization performance of ANNs, and how to deal with questions such as how many layers of hidden units to use, how many units there should be in each layer, and what type of non-linearities should be used, etc.

For a particular ANN the search for a good set of weights for a given training set is complicated by the fact that there can be local optima in the optimization problem; this can cause significant difficulties in practice. In contrast for Gaussian process regression and classification the posterior for the latent variables is convex.

Bayesian neural networks

One approach to the problems raised above was to put ANNs in a Bayesian framework, as developed by MacKay [1992a] and Neal [1996]. This gives rise

to posterior distributions over weights for a given architecture, and the use of the marginal likelihood (see section 5.2) for model comparison and selection. In contrast to Gaussian process regression the marginal likelihood for a given ANN model is not analytically tractable, and thus approximation techniques such as the Laplace approximation [MacKay, 1992a] and Markov chain Monte Carlo methods [Neal, 1996] have to be used. Neal's observation [1996] that certain ANNs with one hidden layer converge to a Gaussian process prior over functions (see section 4.2.3) led us to consider GPs as alternatives to ANNs.

MacKay [2003, sec. 45.7] raises an interesting question whether in moving from neural networks to Gaussian processes we have "thrown the baby out with the bathwater?". This question arises from his statements that "neural networks were meant to be intelligent models that discovered features and patterns in data", while "Gaussian processes are simply smoothing devices". Our answer to this question is that GPs give us a computationally attractive method for dealing with the smoothing problem for a given kernel, and that issues of feature discovery etc. can be addressed through methods to select the kernel function (see chapter 5 for more details on how to do this). Note that using a distance function $r^2(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^\top M(\mathbf{x} - \mathbf{x}')$ with M having a low-rank form $M = \Lambda\Lambda^\top + \Psi$ as in eq. (4.22), features are described by the columns of Λ. However, some of the non-convexity of the neural network optimization problem now returns, as optimizing the marginal likelihood in terms of the parameters of M may well have local optima.

linear and logistic regression

As we have seen from chapters 2 and 3 linear regression and logistic regression with Gaussian priors on the parameters are a natural starting point for the development of Gaussian process regression and Gaussian process classification. However, we need to enhance the flexibility of these models, and the use of non-degenerate kernels opens up the possibility of universal consistency.

kernel smoothers and classifiers

Kernel smoothers and classifiers have been described in sections 2.6 and 7.2.1. At a high level there are similarities between GP prediction and these methods as a kernel is placed on every training example and the prediction is obtained through a weighted sum of the kernel functions, but the details of the prediction and the underlying logic differ. Note that the GP prediction view gives us much more, e.g. error bars on the predictions and the use of the marginal likelihood to set parameters in the kernel (see section 5.2). On the other hand the computational problem that needs to be solved to carry out GP prediction is more demanding than that for simple kernel-based methods.

Kernel smoothers and classifiers are non-parametric methods, and consistency can often be obtained under conditions where the width h of the kernel tends to zero while $nh^D \to \infty$. The equivalent kernel analysis of GP regression (section 7.1) shows that there are quite close connections between the kernel regression method and GPR, but note that the equivalent kernel automatically reduces its width as n grows; in contrast the decay of h has to be imposed for kernel regression. Also, for some kernel smoothing and classification algorithms the width of the kernel is increased in areas of low observation density; for example this would occur in algorithms that consider the k nearest neighbours of a test point. Again notice from the equivalent kernel analysis that the width

of the equivalent kernel is larger in regions of low density, although the exact dependence on the density will depend on the kernel used.

regularization networks, splines, SVMs and RVMs

The similarities and differences between GP prediction and regularization networks, splines, SVMs and RVMs have been discussed in chapter 6.

* 7.6 Appendix: Learning Curve for the Ornstein-Uhlenbeck Process

We now consider the calculation of the learning curve for the OU covariance function $k(r) = \exp(-\alpha|r|)$ on the interval $[0, 1]$, assuming that the training x's are drawn from the uniform distribution $U(0, 1)$. Our treatment is based on Williams and Vivarelli [2000].[8] We first calculate $E^g(X)$ for a fixed design, and then integrate over possible designs to obtain $E^g(n)$.

In the absence of noise the OU process is Markovian (as discussed in Appendix B and exercise 4.5.1). We consider the interval $[0, 1]$ with points $x_1 < x_2 \ldots < x_{n-1} < x_n$ placed on this interval. Also let $x_0 = 0$ and $x_{n+1} = 1$. Due to the Markovian nature of the process the prediction at a test point x depends only on the function values of the training points immediately to the left and right of x. Thus in the i-th interval (counting from 0) the bounding points are x_i and x_{i+1}. Let this interval have length δ_i.

Using eq. (7.24) we have

$$E^g(X) \;=\; \int_0^1 \sigma_f^2(x)\, dx \;=\; \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} \sigma_f^2(x)\, dx, \tag{7.35}$$

where $\sigma_f^2(x)$ is the predictive variance at input x. Using the Markovian property we have in interval i (for $i = 1, \ldots, n-1$) that $\sigma_f^2(x) = k(0) - \mathbf{k}(x)^\top K^{-1}\mathbf{k}(x)$ where K is the 2×2 Gram matrix

$$K \;=\; \begin{pmatrix} k(0) & k(\delta_i) \\ k(\delta_i) & k(0) \end{pmatrix} \tag{7.36}$$

and $\mathbf{k}(x)$ is the corresponding vector of length 2. Thus

$$K^{-1} \;=\; \frac{1}{\Delta_i} \begin{pmatrix} k(0) & -k(\delta_i) \\ -k(\delta_i) & k(0) \end{pmatrix}, \tag{7.37}$$

where $\Delta_i = k^2(0) - k^2(\delta_i)$ and

$$\sigma_f^2(x) \;=\; k(0) - \frac{1}{\Delta_i}[k(0)(k^2(x_{i+1}-x) + k^2(x-x_i)) - 2k(\delta_i)k(x-x_i)k(x_{i+1}-x)]. \tag{7.38}$$

Thus

$$\int_{x_i}^{x_{i+1}} \sigma_f^2(x)dx = \delta_i k(0) - \frac{2}{\Delta_i}(I_1(\delta_i) - I_2(\delta_i)) \tag{7.39}$$

[8] CW thanks Manfred Opper for pointing out that the upper bound developed in Williams and Vivarelli [2000] is exact for the noise-free OU process.

where

$$I_1(\delta) \;=\; k(0)\int_0^\delta k^2(z)dz, \qquad I_2(\delta) \;=\; k(\delta)\int_0^\delta k(z)k(\delta - z)dz. \tag{7.40}$$

For $k(r) = \exp(-\alpha|r|)$ these equations reduce to $I_1(\delta) = (1 - e^{-2\alpha\delta})/(2\alpha)$, $I_2(\delta) = \delta e^{-2\alpha\delta}$ and $\Delta = 1 - e^{-2\alpha\delta}$. Thus

$$\int_{x_i}^{x_{i+1}} \sigma_f^2(x)dx \;=\; \delta_i - \frac{1}{\alpha} + \frac{2\delta_i e^{-2\alpha\delta_i}}{1 - e^{-2\alpha\delta_i}}. \tag{7.41}$$

This calculation is not correct in the first and last intervals where only x_1 and x_n are relevant (respectively). For the 0th interval we have that $\sigma_f^2(x) = k(0) - k^2(x_1 - x)/k(0)$ and thus

$$\int_0^{x_1} \sigma_f^2(x) \;=\; \delta_0 k(0) - \frac{1}{k(0)}\int_0^{x_1} k^2(x_1 - x)dx \tag{7.42}$$

$$\;=\; \delta_0 - \frac{1}{2\alpha}(1 - e^{-2\alpha\delta_0}), \tag{7.43}$$

and a similar result holds for $\int_{x_n}^1 \sigma_f^2(x)$.

Putting this all together we obtain

$$E^g(X) \;=\; 1 - \frac{n}{\alpha} + \frac{1}{2\alpha}(e^{-2\alpha\delta_0} + e^{-2\alpha\delta_n}) + \sum_{i=1}^{n-1} \frac{2\delta_i e^{-2\alpha\delta_i}}{1 - e^{-2\alpha\delta_i}}. \tag{7.44}$$

Choosing a regular grid so that $\delta_0 = \delta_n = 1/2n$ and $\delta_i = 1/n$ for $i = 1, \ldots, n-1$ it is straightforward to show (see exercise 7.7.4) that E^g scales as $\mathcal{O}(n^{-1})$, in agreement with the general Sacks-Ylvisaker result [Ritter, 2000, p. 103] when it is recalled that the OU process obeys Sacks-Ylvisaker conditions of order 0. A similar calculation is given in Plaskota [1996, sec. 3.8.2] for the Wiener process on $[0, 1]$ (note that this is also Markovian, but non-stationary).

We have now worked out the generalization error for a fixed design X. However to compute $E^g(n)$ we need to average $E^g(X)$ over draws of X from the uniform distribution. The theory of order statistics David [1970, eq. 2.3.4] tells us that $p(\delta) = n(1-\delta)^{n-1}$ for all the δ_i, $i = 0, \ldots, n$. Taking the expectation of $E^g(X)$ then turns into the problem of evaluating the one-dimensional integrals $\int e^{-2\alpha\delta}p(\delta)d\delta$ and $\int \delta e^{-2\alpha\delta}(1 - e^{-2\alpha\delta})^{-1}p(\delta)d\delta$. Exercise 7.7.5 asks you to compute these integrals numerically.

7.7 Exercises

1. Consider a spline regularizer with $S_f(\mathbf{s}) = c^{-1}|\mathbf{s}|^{-2m}$. (As we noted in section 6.3 this is not strictly a power spectrum as the spline is an improper prior, but it can be used as a power spectrum in eq. (7.9) for the

purposes of this analysis.) The equivalent kernel corresponding to this spline is given by

$$h(\mathbf{x}) \;=\; \int \frac{\exp(2\pi i \mathbf{s}\cdot\mathbf{x})}{1+\gamma|\mathbf{s}|^{2m}}\, d\mathbf{s}, \tag{7.45}$$

where $\gamma = c\sigma_n^2/\rho$. By changing variables in the integration to $|\mathbf{t}| = \gamma^{1/2m}|\mathbf{s}|$ show that the width of $h(\mathbf{x})$ scales as $n^{-1/2m}$.

2. Equation 7.45 gives the form of the equivalent kernel for a spline regularizer. Show that $h(\mathbf{0})$ is only finite if $2m > D$. (Hint: transform the integration to polar coordinates.) This observation was made by P. Whittle in the discussion of Silverman [1985], and shows the need for the condition $2m > D$ for spline smoothing.

3. Computer exercise: Space $n+1$ points out evenly along the interval $(-1/2, 1/2)$. (Take n to be even so that one of the sample points falls at 0.) Calculate the weight function (see section 2.6) corresponding to Gaussian process regression with a particular covariance function and noise level, and plot this for the point $x = 0$. Now compute the equivalent kernel corresponding to the covariance function (see, e.g. the examples in section 7.1.1), plot this on the same axes and compare results. Hint 1: Recall that the equivalent kernel is defined in terms of integration (see eq. (7.7)) so that there will be a scaling factor of $1/(n+1)$. Hint 2: If you wish to use large n (say > 1000), use the n_{grid} method described in section 2.6.

4. Consider $E^g(X)$ as given in eq. (7.44) and choose a regular grid design X so that $\delta_0 = \delta_n = 1/2n$ and $\delta_i = 1/n$ for $i = 1, \ldots, n-1$. Show that $E^g(X)$ scales as $\mathcal{O}(n^{-1})$ asymptotically. Hint: when expanding $1 - \exp(-2\alpha\delta_i)$, be sure to extend the expansion to sufficient order.

5. Compute numerically the expectation of $E^g(X)$ eq. (7.44) over random designs for the OU process example discussed in section 7.6. Make use of the fact [David, 1970, eq. 2.3.4] that $p(\delta) = n(1-\delta)^{n-1}$ for all the δ_i, $i = 0, \ldots, n$. Investigate the scaling behaviour of $E^g(n)$ w.r.t. n.