Gaussian Process Regression with Mismatched Models

Peter Sollich
Department of Mathematics, King's College London
Strand, London WC2R 2LS, U.K. Email peter.sollich@kcl.ac.uk

Abstract

Learning curves for Gaussian process regression are well understood when the 'student' model happens to match the 'teacher' (true data generation process). I derive approximations to the learning curves for the more generic case of *mismatched models*, and find very rich behaviour: For large input space dimensionality, where the results become exact, there are universal (student-independent) plateaux in the learning curve, with transitions in between that can exhibit arbitrarily many over-fitting maxima; over-fitting can occur even if the student estimates the teacher noise level correctly. In lower dimensions, plateaux also appear, and the learning curve remains dependent on the mismatch between student and teacher even in the asymptotic limit of a large number of training examples. Learning with excessively strong smoothness assumptions can be particularly dangerous: For example, a student with a standard radial basis function covariance function will learn a rougher teacher function only logarithmically slowly. All predictions are confirmed by simulations.

1 Introduction

There has in the last few years been a good deal of excitement about the use of Gaussian processes (GPs) as an alternative to feedforward networks [1]. GPs make prior assumptions about the problem to be learned very transparent, and even though they are non-parametric models, inference—at least in the case of regression considered below—is relatively straightforward. One crucial question for applications is then how 'fast' GPs learn, i.e. how many training examples are needed to achieve a certain level of generalization performance. The typical (as opposed to worst case) behaviour is captured in the *learning curve*, which gives the average generalization error ϵ as a function of the number of training examples n. Good bounds and approximations for $\epsilon(n)$ are now available [1, 2, 3, 4, 5], but these are mostly restricted to the case where the 'student' model exactly matches the true 'teacher' generating the data[1]. In practice, such a match is unlikely, and so it is

[1]The exception is the elegant work of Malzahn and Opper [2], which uses a statistical physics framework to derive approximate learning curves that also apply for any *fixed* target function. However, this framework has not yet to my knowledge been exploited to

important to understand how GPs learn if there is some model mismatch. This is the aim of this paper.

In its simplest form, the regression problem is this: We are trying to learn a function θ_* which maps inputs x (real-valued vectors) to (real-valued scalar) outputs $\theta_*(x)$. We are given a set of training data D, consisting of n input-output pairs (x^l, y^l); the training outputs y^l may differ from the 'clean' teacher outputs $\theta_*(x^l)$ due to corruption by noise. Given a test input x, we are then asked to come up with a prediction $\hat{\theta}(x)$, plus error bar, for the corresponding output $\theta(x)$. In a Bayesian setting, we do this by specifying a prior $P(\theta)$ over hypothesis functions, and a likelihood $P(D|\theta)$ with which each θ could have generated the training data; from this we deduce the posterior distribution $P(\theta|D) \propto P(D|\theta)P(\theta)$. For a GP, the prior is defined directly over input-output functions θ; this is simpler than for a Bayesian feedforward net since no weights are involved which would have to be integrated out. Any θ is uniquely determined by its output values $\theta(x)$ for all x from the input domain, and for a GP, these are assumed to have a joint Gaussian distribution (hence the name). If we set the means to zero as is commonly done, this distribution is fully specified by the *covariance function* $\langle\theta(x)\theta(x')\rangle_\theta = C(x, x')$. The latter transparently encodes prior assumptions about the function to be learned. Smoothness, for example, is controlled by the behaviour of $C(x, x')$ for $x' \to x$: The Ornstein-Uhlenbeck (OU) covariance function $C(x, x') = \exp(-|x - x'|/l)$ produces very rough (non-differentiable) functions, while functions sampled from the radial basis function (RBF) prior with $C(x, x') = \exp[-|x - x'|^2/(2l^2)]$ are infinitely differentiable. Here l is a lengthscale parameter, corresponding directly to the distance in input space over which we expect significant variation in the function values.

There are good reviews on how inference with GPs works [1, 6], so I only give a brief summary here. The student assumes that outputs y are generated from the 'clean' values of a hypothesis function $\theta(x)$ by adding Gaussian noise of x-independent variance σ^2. The joint distribution of a set of training outputs $\{y^l\}$ and the function values $\theta(x)$ is then also Gaussian, with covariances given (under the student model) by

$$\langle y^l y^m\rangle = C(x^l, x^m) + \sigma^2\delta_{lm} = (\mathbf{K})_{lm}, \qquad \langle y^l\theta(x)\rangle = C(x^l, x) = (\mathbf{k}(x))_l$$

Here I have defined an $n \times n$ matrix $\mathbf{K}$ and an x-dependent n-component vector $\mathbf{k}(x)$. The posterior distribution $P(\theta|D)$ is then obtained by conditioning on the $\{y^l\}$; it is again Gaussian and has mean and variance

$$\langle\theta(x)\rangle_{\theta|D} \equiv \hat{\theta}(x|D) = \mathbf{k}(x)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} \tag{1}$$

$$\langle(\theta(x) - \hat{\theta}(x))^2\rangle_{\theta|D} = C(x, x) - \mathbf{k}(x)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}(x) \tag{2}$$

From the student's point of view, this solves the inference problem: The best prediction for $\theta(x)$ on the basis of the data D is $\hat{\theta}(x|D)$, with a (squared) error bar given by (2). The squared deviation between the prediction and the teacher is $[\hat{\theta}(x|D) - \theta_*(x)]^2$; the average generalization error (which, as a function of n, defines the learning curve) is obtained by averaging this over the posterior distribution of teachers, all datasets, and the test input x:

$$\epsilon = \langle\langle\langle[\hat{\theta}(x|D) - \theta_*(x)]^2\rangle_{\theta_*|D}\rangle_D\rangle_x \tag{3}$$

Now of course the student does not know the true posterior of the teacher; to estimate ϵ, she must assume that it is identical to the student posterior, giving from (2)

$$\hat{\epsilon} = \langle\langle\langle[\hat{\theta}(x|D) - \theta(x)]^2\rangle_{\theta|D}\rangle_D\rangle_x = \langle\langle C(x, x) - \mathbf{k}(x)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}(x)\rangle_{\{x^l\}}\rangle_x \tag{4}$$

---

consider systematically the effects of having a mismatch between the teacher prior over target functions and the prior assumed by the student.

where in the last expression I have replaced the average over D by one over the training inputs since the outputs no longer appear. If the student model matches the true teacher model, ϵ and $\hat{\epsilon}$ coincide and give the Bayes error, i.e. the best achievable (average) generalization performance for the given teacher.

I assume in what follows that the teacher is also a GP, but with a possibly different covariance function $C_*(x,x')$ and noise level σ_*^2. This allows eq. (3) for ϵ to be simplified, since by exact analogy with the argument for the student posterior

$$\langle\theta_*(x)\rangle_{\theta_*|D} = \mathbf{k}_*(x)^{\mathrm{T}}\mathbf{K}_*^{-1}\mathbf{y}, \quad \langle\theta_*^2(x)\rangle_{\theta_*|D} = \langle\theta_*(x)\rangle_{\theta_*|D}^2 + C_*(x,x) - \mathbf{k}_*(x)^{\mathrm{T}}\mathbf{K}_*^{-1}\mathbf{k}_*(x)$$

and thus, abbreviating $\mathbf{a}(x) = \mathbf{K}^{-1}\mathbf{k}(x) - \mathbf{K}_*^{-1}\mathbf{k}_*(x)$,

$$\epsilon = \langle\langle \mathbf{a}(x)^{\mathrm{T}}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{a}(x) + C_*(x,x) - \mathbf{k}_*(x)^{\mathrm{T}}\mathbf{K}_*^{-1}\mathbf{k}_*(x)\rangle_D\rangle_x$$

Conditional on the training inputs, the training outputs have a Gaussian distribution given by the true (teacher) model; hence $\langle\mathbf{y}\mathbf{y}^{\mathrm{T}}\rangle_{\{y^l\}|\{x^l\}} = \mathbf{K}_*$, giving

$$\epsilon = \langle\langle C_*(x,x) - 2\mathbf{k}_*(x)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}(x) + \mathbf{k}(x)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{K}_*\mathbf{K}^{-1}\mathbf{k}(x)\rangle_{\{x^l\}}\rangle_x \tag{5}$$

2 Calculating the learning curves

An exact calculation of the learning curve $\epsilon(n)$ is difficult because of the joint average in (5) over the training inputs X and the test input x. A more convenient starting point is obtained if (using Mercer's theorem) we decompose the covariance function into its eigenfunctions $\phi_i(x)$ and eigenvalues Λ_i, defined w.r.t. the input distribution so that $\langle C(x,x')\phi_i(x')\rangle_{x'} = \Lambda_i\phi_i(x)$ with the corresponding normalization $\langle\phi_i(x)\phi_j(x)\rangle_x = \delta_{ij}$. Then

$$C(x,x') = \sum_{i=1}^{\infty}\Lambda_i\phi_i(x)\phi_i(x'), \quad \text{and similarly } C_*(x,x') = \sum_{i=1}^{\infty}\Lambda_i^*\phi_i(x)\phi_i(x') \tag{6}$$

For simplicity I assume here that the student and teacher covariance functions have the *same* eigenfunctions (but different eigenvalues). This is not as restrictive as it may seem; several examples are given below. The averages over the test input x in (5) are now easily carried out: E.g. for the last term we need

$$\langle(\mathbf{k}(x)\mathbf{k}(x)^{\mathrm{T}})_{lm}\rangle_x = \sum_{ij}\Lambda_i\Lambda_j\phi_i(x^l)\langle\phi_i(x)\phi_j(x)\rangle_x\phi_j(x^m) = \sum_i \Lambda_i^2\phi_i(x^l)\phi_i(x^m)$$

Introducing the diagonal eigenvalue matrix $(\mathbf{\Lambda})_{ij} = \Lambda_i\delta_{ij}$ and the 'design matrix' $(\Phi)_{li} = \phi_i(x^l)$, this reads $\langle\mathbf{k}(x)\mathbf{k}(x)^{\mathrm{T}}\rangle_x = \Phi\mathbf{\Lambda}^2\Phi^{\mathrm{T}}$. Similarly, for the second term in (5), $\langle\mathbf{k}(x)\mathbf{k}_*^{\mathrm{T}}(x)\rangle_x = \Phi\mathbf{\Lambda}\mathbf{\Lambda}_*\Phi^{\mathrm{T}}$, and $\langle C_*(x,x)\rangle_x = \operatorname{tr}\mathbf{\Lambda}_*$. This gives, dropping the training inputs subscript from the remaining average,

$$\epsilon = \langle \operatorname{tr}\mathbf{\Lambda}_* - 2\operatorname{tr}\Phi\mathbf{\Lambda}\mathbf{\Lambda}_*\Phi^{\mathrm{T}}\mathbf{K}^{-1} + \operatorname{tr}\Phi\mathbf{\Lambda}^2\Phi^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{K}_*\mathbf{K}^{-1}\rangle$$

In this new representation we also have $\mathbf{K} = \sigma^2\mathbf{I} + \Phi\mathbf{\Lambda}\Phi^{\mathrm{T}}$ and similarly for $\mathbf{K}_*$; for the inverse of $\mathbf{K}$ we can use the Woodbury formula to write $\mathbf{K}^{-1} = \sigma^{-2}[\mathbf{I} - \sigma^{-2}\Phi\mathcal{G}\Phi^{\mathrm{T}}]$, where $\mathcal{G} = (\mathbf{\Lambda}^{-1} + \sigma^{-2}\Phi^{\mathrm{T}}\Phi)^{-1}$. Inserting these results, one finds after some algebra that

$$\epsilon = \sigma_*^2\sigma^{-2}\left[\langle\operatorname{tr}\mathcal{G}\rangle - \langle\operatorname{tr}\mathcal{G}\mathbf{\Lambda}^{-1}\mathcal{G}\rangle\right] + \langle\operatorname{tr}\mathcal{G}\mathbf{\Lambda}_*\mathbf{\Lambda}^{-2}\mathcal{G}\rangle \tag{7}$$

which for the matched case reduces to the known result for the Bayes error [4]

$$\hat{\epsilon} = \langle\operatorname{tr}\mathcal{G}\rangle \tag{8}$$

Eqs. (7,8) are still exact. We now need to tackle the remaining averages over training inputs. Two of these are of the form $\langle \mathrm{tr}\, \mathcal{G}\mathbf{M}\mathcal{G}\rangle$; if we generalize the definition of $\mathcal{G}$ to $\mathcal{G} = (\mathbf{\Lambda}^{-1} + v\mathbf{I} + w\mathbf{M} + \sigma^{-2}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}$ and define $g = \langle \mathrm{tr}\, \mathcal{G}\rangle$, then they reduce to $\langle \mathrm{tr}\, \mathcal{G}\mathbf{M}\mathcal{G}\rangle = -\partial g/\partial w$. (The derivative is taken at $v = w = 0$; the idea behind introducing v will become clear shortly.) So it is sufficient to calculate g. To do this, consider how $\mathcal{G}$ changes when a new example is added to the training set. One has

$$\mathcal{G}(n+1) - \mathcal{G}(n) = \left[\mathcal{G}^{-1}(n) + \sigma^{-2}\psi\psi^{\mathrm{T}}\right]^{-1} - \mathcal{G}(n) = -\frac{\mathcal{G}(n)\psi\psi^{\mathrm{T}}\mathcal{G}(n)}{\sigma^2 + \psi^{\mathrm{T}}\mathcal{G}(n)\psi} \tag{9}$$

in terms of the vector ψ with elements $(\psi)_i = \phi_i(x_{n+1})$, using again the Woodbury formula. To obtain the change in g we need the average of (9) over both the new training input x_{n+1} and all previous ones. This cannot be done exactly, but we can approximate by averaging numerator and denominator separately; taking the trace then gives $g(n+1) - g(n) = -\langle \mathrm{tr}\, \mathcal{G}^2(n)\rangle/[\sigma^2 + g(n)]$. Now, using our auxiliary parameter v, $-\langle \mathrm{tr}\, \mathcal{G}^2\rangle = \partial g/\partial v$; if we also approximate n as continuous, we get the simple partial differential equation $\partial g/\partial n = (\partial g/\partial v)/(\sigma^2 + g)$ with the initial condition $g|_{n=0} = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + v\mathbf{I} + w\mathbf{M})^{-1}$. Solving this using the method of characteristics [7] gives a self-consistent equation for g,

$$g = \mathrm{tr}\left[\mathbf{\Lambda}^{-1} + \left(v + \frac{n}{\sigma^2 + g}\right)\mathbf{I} + w\mathbf{M}\right]^{-1} \tag{10}$$

The Bayes error (8) is $\hat{\epsilon} = g|_{v=w=0}$ and therefore obeys

$$\hat{\epsilon} = \mathrm{tr}\, \mathbf{G}, \qquad \mathbf{G}^{-1} = \mathbf{\Lambda}^{-1} + \frac{n}{\sigma^2 + \hat{\epsilon}}\mathbf{I} \tag{11}$$

within our approximation (called 'LC' in [4]). To obtain ϵ, we differentiate both sides of (10) w.r.t. w, set $v = w = 0$ and rearrange to give

$$\langle \mathrm{tr}\, \mathcal{G}\mathbf{M}\mathcal{G}\rangle = -\partial g/\partial w = (\mathrm{tr}\, \mathbf{M}\mathbf{G}^2)/[1 - (\mathrm{tr}\, \mathbf{G}^2)n/(\sigma^2 + \hat{\epsilon})^2]$$

Using this result in (7), with $\mathbf{M} = \mathbf{\Lambda}^{-1}$ and $\mathbf{M} = \mathbf{\Lambda}^{-1}\mathbf{\Lambda}_*\mathbf{\Lambda}^{-1}$, we find after some further simplifications the final (approximate) result for the learning curve:

$$\epsilon = \hat{\epsilon}\, \frac{\sigma_*^2\, \mathrm{tr}\, \mathbf{G}^2 + n^{-1}(\sigma^2 + \hat{\epsilon})^2\, \mathrm{tr}\, \mathbf{\Lambda}_*\mathbf{\Lambda}^{-2}\mathbf{G}^2}{\sigma^2\, \mathrm{tr}\, \mathbf{G}^2 + n^{-1}(\sigma^2 + \hat{\epsilon})^2\, \mathrm{tr}\, \mathbf{\Lambda}^{-1}\mathbf{G}^2} \tag{12}$$

which transparently shows how in the matched case ϵ and $\hat{\epsilon}$ become identical.

3 Examples

I now apply the result for the learning curve (11,12) to some exemplary learning scenarios. First, consider inputs x which are binary vectors[2] with d components $x_a \in \{-1, 1\}$, and assume that the input distribution is uniform. We consider covariance functions for student and teacher which depend on the product $x \cdot x'$ only; this includes the standard choices (e.g. OU and RBF) which depend on the Euclidean distance $|x - x'|$, since $|x - x'|^2 = 2d - 2x \cdot x'$. All these have the same eigenfunctions [9], so our above assumption is satisfied. The eigenfunctions are indexed by subsets ρ of $\{1, 2 \ldots d\}$ and given explicitly by $\phi_\rho(x) = \prod_{a\in\rho} x_a$. The

[2]This scenario may seem strange, but simplifies the determination of the eigenfunctions and eigenvalues. For large d, one expects other distributions with continuously varying x and the same first- and second-order statistics ($\langle x_a\rangle = 0$, $\langle x_a x_b\rangle = \delta_{ab}$) to give similar results [8].

corresponding eigenvalues depend only on the size $s = |\rho|$ of the subsets and are therefore $\binom{d}{s}$-fold degenerate; letting $e = (1, 1 \ldots 1)$ be the 'all ones' input vector, they have the values $\Lambda_s = \langle C(x, e)\phi_\rho(x)\rangle_x$ (which can easily be evaluated as an average over two binomially distributed variables, counting the number of $+1$'s in x overall and among the x_a with $a \in \rho$). With the Λ_s and Λ_s^* determined, it is then a simple matter to evaluate the predicted learning curve (11,12) numerically. First, though, focus on the limit of large d, where much more can be said. If we write $C(x, x') = f(x \cdot x'/d)$, the eigenvalues become, for $d \to \infty$, $\Lambda_s = d^{-s} f^{(s)}(0)$ and the contribution to $C(x, x) = f(1)$ from the s-th eigenvalue block is $\lambda_s \equiv \binom{d}{s}\Lambda_s \to f^{(s)}(0)/s!$, consistent with $f(1) = \sum_{s=0}^{\infty} f^{(s)}(0)/s!$ The Λ_s, because of their scaling with d, become infinitely separated for $d \to \infty$. For training sets of size $n = \mathcal{O}(d^L)$, we then see from (11) that eigenvalues with $s > L$ contribute as if $n = 0$, since $\Lambda_s \gg n/(\sigma^2 + \hat{\epsilon})$; they have effectively not yet been learned. On the other hand, eigenvalues with $s < L$ are completely suppressed and have been learnt perfectly. We thus have a hierarchical learning scenario, where different scalings of n with d—as defined by L—correspond to different 'learning stages'. Formally, we can analyse the stages separately by letting $d \to \infty$ at a constant ratio $\alpha = n/\binom{d}{L}$ of the number of examples to the number of parameters to be learned at stage L (note $\binom{d}{L} = \mathcal{O}(d^L)$ for large d). An independent (replica) calculation along the lines of Ref. [8] shows that our approximation for the learning curve actually becomes *exact* in this limit. The resulting α-dependence of ϵ can be determined explicitly: Set $f_L = \sum_{s \geq L} \lambda_s$ (so that $f_0 = f(1)$) and similarly for f_L^*. Then for large α,

$$\epsilon = f_{L+1}^* + (f_{L+1}^* + \sigma_*^2)\alpha^{-1} + \mathcal{O}(\alpha^{-2}) \tag{13}$$

This implies that, during successive learning stages, (teacher) eigenvalues are learnt one by one and their contribution eliminated from the generalization error, giving plateaux in the learning curve at $\epsilon = f_1^*, f_2^*, \ldots$. These plateaux, as well as the asymptotic decay (13) towards them, are universal [8], i.e. student-independent. The (non-universal) behaviour for smaller α can also be fully characterized: Consider first the simple case of linear perceptron learning (see e.g. [7]), which corresponds to both student and teacher having simple dot-product covariance functions $C(x, x') = C_*(x, x') = x \cdot x'/d$. In this case there is only a single learning stage (only $\lambda_1 = \lambda_1^* = 1$ are nonzero), and $\epsilon = r(\alpha)$ decays from $r(0) = 1$ to $r(\infty) = 0$, with an over-fitting maximum around $\alpha = 1$ if σ^2 is sufficiently small compared to σ_*^2. In terms of this function $r(\alpha)$, the learning curve at stage L for *general* covariance functions is then *exactly* given by $\epsilon = f_{L+1}^* + \lambda_L^* r(\alpha)$ if in the evaluation of $r(\alpha)$ the effective noise levels $\tilde{\sigma}^2 = (f_{L+1} + \sigma^2)/\lambda_L$ and $\tilde{\sigma}_*^2 = (f_{L+1}^* + \sigma_*^2)/\lambda_L^*$ are used. Note how in $\tilde{\sigma}_*^2$, the contribution f_{L+1}^* from the not-yet-learned eigenvalues acts as effective noise, and is normalized by the amount of 'signal' $\lambda_L^* = f_L^* - f_{L+1}^*$ available at learning stage L. The analogous definition of $\tilde{\sigma}^2$ implies that, for small σ^2 and depending on the choice of student covariance function, there can be arbitrarily many learning stages L where $\tilde{\sigma}^2 \ll \tilde{\sigma}_*^2$, and therefore *arbitrarily many over-fitting maxima* in the resulting learning curves. From the definitions of $\tilde{\sigma}^2$ and $\tilde{\sigma}_*^2$ it is clear that this situation can occur *even if the student knows the exact teacher noise level*, i.e. even if $\sigma^2 = \sigma_*^2$.

Fig. 1(left) demonstrates that the above conclusions hold not just for $d \to \infty$; even for the cases shown, with $d = 10$, up to three over-fitting maxima are apparent. Our theory provides a very good description of the numerically simulated learning curves even though, at such small d, the predictions are still significantly different from those for $d \to \infty$ (see Fig. 1(right)) and therefore not guaranteed to be exact.

In the second example scenario, I consider continuous-valued input vectors, uni-

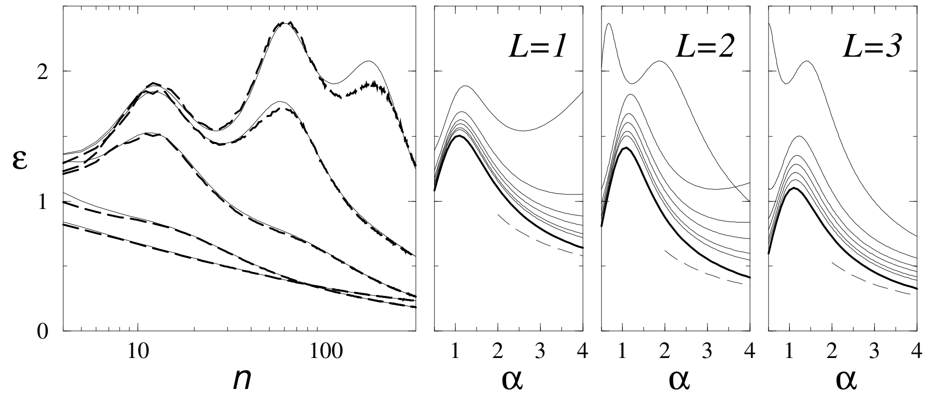

Figure 1: Left: Learning curves for RBF student and teacher, with uniformly distributed, binary input vectors with $d = 10$ components. Noise levels: Teacher $\sigma_*^2 = 1$, student $\sigma^2 = 10^{-4}, 10^{-3}, \ldots, 1$ (top to bottom). Length scales: Teacher $l_* = d^{1/2}$, student $l = 2d^{1/2}$. Dashed: numerical simulations, solid: theoretical prediction. Right: Learning curves for $\sigma^2 = 10^{-4}$ and increasing d (top to bottom: 10, 20, 30, 40, 60, 80, [bold] ∞). The x-axis shows $\alpha = n/\binom{d}{L}$, for learning stages $L = 1, 2, 3$; the dashed lines are the universal asymptotes (13) for $d \to \infty$.

formly distributed over the unit interval $[0, 1]$; generalization to d dimensions ($x \in [0, 1]^d$) is straightforward. For covariance functions which are stationary, i.e. dependent on x and x' only through $x - x'$, and assuming periodic boundary conditions (see [4] for details), one then again has covariance function-independent eigenfunctions. They are indexed by integers[3] q, with $\phi_q(x) = e^{2\pi i q x}$; the corresponding eigenvalues are $\Lambda_q = \int dx\, C(0, x) e^{-2\pi i q x}$. For the ('periodified') RBF covariance function $C(x, x') = \exp[-(x - x')^2/(2l^2)]$, for example, one has $\Lambda_q \propto \exp(-\tilde{q}^2/2)$, where $\tilde{q} = 2\pi l q$. The OU case $C(x, x') = \exp(-|x - x'|/l)$, on the other hand, gives $\Lambda_q \propto (1 + \tilde{q}^2)^{-1}$, thus $\Lambda_q \propto q^{-2}$ for large q. I also consider below covariance functions which interpolate in smoothness between the OU and RBF limits: E.g. the MB2 (modified Bessel) covariance $C(x, x') = e^{-a}(1 + a)$, with $a = |x - x'|/l$, yields functions which are once differentiable [5]; its eigenvalues $\Lambda_q \propto (1 + \tilde{q}^2)^{-2}$ show a faster asymptotic power law decay, $\Lambda_q \propto q^{-4}$, than those of the OU covariance function. To subsume all these cases I assume in the following analysis of the general shape of the learning curves that $\Lambda_q \propto q^{-r}$ (and similarly $\Lambda_q^* \propto q^{-r_*}$). Here $r = 2$ for OU, $r = 4$ for MB2, and (due to the faster-than-power law decay of its eigenvalues) effectively $r = \infty$ for RBF.

From (11,12), it is clear that the n-dependence of the Bayes error $\hat{\epsilon}$ has a strong effect on the true generalization error ϵ. From previous work [4], we know that $\hat{\epsilon}(n)$ has two regimes: For small n, where $\hat{\epsilon} \gg \sigma^2$, $\hat{\epsilon}$ is dominated by regions in input space which are too far from the training examples to have significant correlation with them, and one finds $\hat{\epsilon} \propto n^{-(r-1)}$. For much larger n, learning is essentially against noise, and one has a slower decay $\hat{\epsilon} \propto (n/\sigma^2)^{-(r-1)/r}$. These power laws can be derived from (11) by approximating factors such as $[\Lambda_q^{-1} + n/(\sigma^2 + \hat{\epsilon})]^{-1}$ as equal to either Λ_q or to 0, depending on whether $n/(\sigma^2 + \hat{\epsilon}) <$ or $> \Lambda_q^{-1}$. With the same technique, one can estimate the behaviour of ϵ from (12). In the *small* n-regime, one finds $\epsilon \approx c_1 \sigma_*^2 + c_2 n^{-(r_* - 1)}$, with prefactors c_1, c_2 depending on the student. Note

[3]Since $\Lambda_q = \Lambda_{-q}$, one can assume $q \geq 0$ if all Λ_q for $q > 0$ are taken as doubly degenerate.

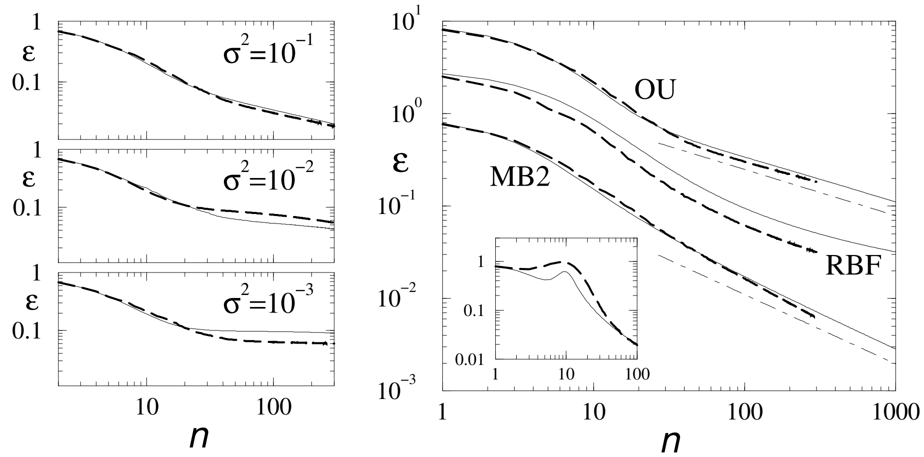

Figure 2: Learning curves for inputs x uniformly distributed over $[0,1]$. Teacher: MB2 covariance function, lengthscale $l_* = 0.1$, noise level $\sigma_*^2 = 0.1$; student lengthscale $l = 0.1$ throughout. Dashed: simulations, solid: theory. Left: OU student with σ^2 as shown. The predicted plateau appears as σ^2 decreases. Right: Students with $\sigma^2 = 0.1$ and covariance function as shown; for clarity, the RBF and OU results have been multiplied by $\sqrt{10}$ and 10, respectively. Dash-dotted lines show the predicted asymptotic power laws for MB2 and OU; the RBF data have a persistent upward curvature consistent with the predicted logarithmic decay. Inset: RBF student with $\sigma^2 = 10^{-3}$, showing the occurrence of over-fitting maxima.

that the contribution proportional to σ_*^2 is automatically negligible in the matched case (since then $\epsilon = \hat{\epsilon} \gg \sigma^2 = \sigma_*^2$ for small n); if there is a model mismatch, however, and if the small-n regime extends far enough, it will become significant. This is the case for small σ^2; indeed, for $\sigma^2 \to 0$, the 'small n' criterion $\hat{\epsilon} \gg \sigma^2$ is satisfied for any n. Our theory thus predicts the appearance of plateaux in the learning curves, becoming more pronounced as σ^2 decreases; Fig. 2(left) confirms this[4]. Numerical evaluation also shows that for small σ^2, over-fitting maxima may occur before the plateau is reached, consistent with simulations; see inset in Fig. 2(right). In the *large* n-regime ($\hat{\epsilon} \ll \sigma^2$), our theory predicts that the generalization error decays as a power law. If the student assumes a rougher function than the teacher provides ($r < r_*$), the asymptotic power law exponent $\epsilon \propto n^{-(r-1)/r}$ is determined by the student alone. In the converse case, the asymptotic decay is $\epsilon \propto n^{-(r_*-1)/r}$ and can be very slow, actually becoming logarithmic for an RBF student ($r \to \infty$). For $r = r_*$, the fastest decay for given r_* is obtained, as expected from the properties of the Bayes error. The simulation data in Fig. 2 are compatible with these predictions (though the simulations cover too small a range of n to allow exponents to be determined precisely). It should be stressed that the above results imply that there is no asymptotic regime of large training sets in which the learning curve assumes a universal form, in contrast to the case of parametric models where the generalization error decays as $\epsilon \propto 1/n$ for sufficiently large n independently of model mismatch (as long as the problem is learnable at all). This conclusion may seem counter-intuitive, but becomes clear if one remembers that a GP covariance function with an infinite number of nonzero eigenvalues Λ_i always has arbitrarily many eigenvalues

[4]If $\sigma^2 = 0$ exactly, the plateau will extend to $n \to \infty$. With hindsight, this is clear: a GP with an infinite number of nonzero eigenvalues has no limit on the number of its 'degrees of freedom' and can fit perfectly any amount of noisy training data, without ever learning the true teacher function.

that are arbitrarily close to zero (since the Λ_i are positive and $\sum_i \Lambda_i = \langle C(x,x) \rangle$ is finite). Whatever n, there are therefore many eigenvalues for which $\Lambda_i^{-1} \gg n/\sigma^2$, corresponding to degrees of freedom which are still mainly determined by the prior rather than the data (compare (11)). In other words, a regime where the data completely overwhelms the mismatched prior—and where the learning curve could therefore become independent of model mismatch—can never be reached.

In summary, the above approximate theory makes a number of non-trivial predictions for GP learning with mismatched models, all borne out by simulations: for large input space dimensions, the occurrence of multiple over-fitting maxima; in lower dimensions, the generic presence of plateaux in the learning curve if the student assumes too small a noise level σ^2, and strong effects of model mismatch on the asymptotic learning curve decay. The behaviour is much richer than for the matched case, and could guide the choice of (student) priors in real-world applications of GP regression; RBF students, for example, run the risk of very slow logarithmic decay of the learning curve if the target (teacher) is less smooth than assumed.

An important issue for future work—some of which is in progress—is to analyse to which extent hyperparameter tuning (e.g. via evidence maximization) can make GP learning robust against some forms of model mismatch, e.g. a misspecified functional form of the covariance function. One would like to know, for example, whether a data-dependent adjustment of the lengthscale of an RBF covariance function would be sufficient to avoid the logarithmically slow learning of rough target functions.

References

[1] See e.g. D J C MacKay, Gaussian Processes, Tutorial at *NIPS 10*; recent papers by Csató *et al.* (*NIPS 12*), Goldberg/Williams/Bishop (*NIPS 10*), Williams and Barber/Williams (*NIPS 9*), Williams/Rasmussen (*NIPS 8*); and references below.

[2] D Malzahn and M Opper. In *NIPS 13*, pages 273–279; also in *NIPS 14*.

[3] C A Michelli and G Wahba. In Z Ziegler, editor, *Approximation theory and applications*, pages 329–348. Academic Press, 1981; M Opper. In I K Kwok-Yee *et al.*, editors, *Theoretical Aspects of Neural Computation*, pages 17–23. Springer, 1997.

[4] P Sollich. In *NIPS 11*, pages 344–350.

[5] C K I Williams and F Vivarelli. *Mach. Learn.*, 40:77–102, 2000.

[6] C K I Williams. In M I Jordan, editor, *Learning and Inference in Graphical Models*, pages 599–621. Kluwer Academic, 1998.

[7] P Sollich. *J. Phys. A*, 27:7771–7784, 1994.

[8] M Opper and R Urbanczik. *Phys. Rev. Lett.*, 86:4410–4413, 2001.

[9] R Dietrich, M Opper, and H Sompolinsky. *Phys. Rev. Lett.*, 82:2975–2978, 1999.