Generalization of Plaskota's bound for Gaussian process learning curves

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

Abstract

[**Note:** This paper is an extended version of the manuscript *Learning curves for Gaussian process regression: Approximations and bounds* by Sollich and Halees. The only difference is the addition of Appendix B, which gives the derivation of the generalized version of Plaskota's bound. The remainder of the paper has been left in place to provide the proper context.]

We consider the problem of calculating learning curves (i.e., average generalization performance) of Gaussian processes used for regression. On the basis of a simple expression for the generalization error, in terms of the eigenvalue decomposition of the covariance function, we derive a number of approximation schemes. We identify where these become exact, and compare with existing bounds on learning curves; the new approximations, which can be used for any input space dimension, generally get substantially closer to the truth. We also study possible improvements to our approximations. Finally, we use a simple exactly solvable learning scenario to show that there are limits of principle on the quality of approximations and bounds expressible solely in terms of the eigenvalue spectrum of the covariance function.

1 Introduction: Gaussian processes

Within the neural networks community, there has in the last few years been a good deal of excitement about the use of Gaussian processes as an alternative to feedforward networks (see *e.g.* (Williams and Rasmussen, 1996; Williams, 1997; Barber and Williams, 1997; Goldberg et al., 1998; Sollich, 1999a; Malzahn and Opper, 2001). The advantages of Gaussian processes are that prior assumptions about the problem to be learned are encoded in a very transparent way, and that inference—at least in the case of regression that we will consider—is relatively straightforward; they are also 'non-parametric' in the sense that their effective number of parameters ('degrees of freedom') can grow arbitrarily large as more and more training data is collected. Interest in Gaussian processes has also been stimulated by the fact that they are at the heart of the large family of 'kernel machine' methods (see *e.g.* www.kernel-machines.org).

One crucial question for applications is then how 'fast' Gaussian processes 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. Several workers have derived bounds on $\epsilon(n)$ (Michelli and Wahba, 1981; Plaskota, 1990; Opper, 1997; Trecate et al., 1999; Opper and Vivarelli, 1999; Williams and Vivarelli, 2000) or studied its large n asymptotics (Silverman, 1985; Ritter, 1996). As we will illustrate below, however, the existing bounds are often far from tight; and asymptotic results will not necessarily apply for realistic sample sizes n. Our main aim in this paper is therefore to derive approximations to $\epsilon(n)$ which get closer to the true learning curves than existing bounds, and apply both for small and large n. We compare these approximations with existing bounds and the results of numerical simulations; possible improvements to the approximations are also discussed. Finally, we study an analytically solvable example scenario which sheds light on how tight bounds on learning curves can be made in principle. Summaries of the early stages of this work have appeared in the conference proceedings (Sollich, 1999a,b).

In its simplest form, the regression problem that we are considering 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' target outputs $\theta^*(x_l)$ due to corruption by noise. Given a test input x, we are then asked to come up with a prediction $\theta(x)$ for the corresponding output, expressed either in the simple form of a mean prediction $\hat{\theta}(x)$ plus error bars, or more comprehensively in terms of a 'predictive distribution' $P(\theta(x)|x, D)$. In a Bayesian setting, we do this by specifying a prior $P(\theta)$ over our 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 $P(\theta|D) \propto P(D|\theta)P(\theta)$. If we wanted to use a feedforward network for this task, we could proceed as follows: Specify candidate networks by a set of weights w, with prior probability $P(w)$. Each network defines a (stochastic) input-output relation described by the distribution of output y given input x (and weights w), $P(y|x, w)$. Multiplying over the whole data set, we get the probability of the observed data having been produced by the network with weights w: $P(D|w) = \prod_{l=1}^{n} P(y_l|x_l, w)$. Bayes' theorem then gives us the posterior, i.e., the probability of network w given the data, as $P(w|D) \propto P(D|w)P(w)$ up to an overall normalization factor. From this, finally, we get the predictive distribution $P(y|x, D) = \int dw P(y|x, w)P(w|D)$. This solves the regression problem in principle, but leaves us with a nasty integral over all possible network weights: the posterior $P(w|D)$ generally has a highly nontrivial structure, with many local peaks (corresponding to local minima in the training error). One therefore has to use sophisticated Monte Carlo integration techniques (Neal, 1993) or local approximations to $P(w|D)$ around its maxima (MacKay, 1992) to tackle this problem. Even once this has been done, one is still left with the question of how to interpret the results: We may for example want to select priors on the basis of the data, *e.g.* by making the prior $P(w|h)$ dependent on a set of hyperparameters h and choosing h such as to maximize the probability $P(D|h)$ of the data. Once we have found the 'optimal' prior we would then hope that it tells us something about the regression problem at hand (whether certain input components are irrelevant, for example). This would be easy if the prior told us directly how likely certain input-output functions are; instead we have to extract this information from the prior over weights, often a complicated process.

By contrast, for a Gaussian process it is an almost trivial task to obtain the posterior and the predictive distribution (see below). One reason for this is that the prior $P(\theta)$ is defined directly over input-output functions θ. How is this done? Any θ is uniquely determined by its output values $\theta(x)$ for all x from the input domain, and for a Gaussian process, these are simply assumed to have a joint Gaussian distribution (hence the name). This distribution can be specified by the mean values $\langle\theta(x)\rangle_\theta$ (which we assume to be zero in the following, as is commonly done), and the covariances $\langle\theta(x)\theta(x')\rangle_\theta = C(x, x')$; $C(x, x')$ is called the *covariance function* of the Gaussian process. It encodes in an easily interpretable way 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 $C(x, x') \propto \exp(-||x - x'||/l)$ produces very rough (non-differentiable) functions, while functions sampled from the radial basis function (RBF) prior with $C(x, x') \propto \exp[-||x - x'||^2/(2l^2)]$ are—in the mean-square sense—infinitely differentiable. (Intermediate priors yielding r times differentiable functions can also be defined by using modified Bessel functions as covariance functions; see (Stein, 1989).) Figure 1 illustrates these characteristics with two samples from the OU and RBF priors, respectively, over a two-dimensional input domain. The 'length scale' parameter l in the covariance functions also has an intuitive meaning: It corresponds directly to the distance in input space over which we expect our function to vary significantly. More complex properties can also be encoded; by replacing l with different length scales for each input component, for example, relevant (small l) and irrelevant (large l) inputs can be distinguished.

How does inference with Gaussian processes work? We only give a brief summary here and refer to existing reviews on the subject (see *e.g.* (Williams, 1998)) for details. It is simplest to assume 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 n training outputs $\{y_l\}$ and the function values $\theta(x)$ is then also Gaussian, with covariances given by

$$\langle y_l y_m \rangle \quad = \quad C(x_l, x_m) + \sigma^2 \delta_{lm} = (\mathbf{K})_{lm}$$

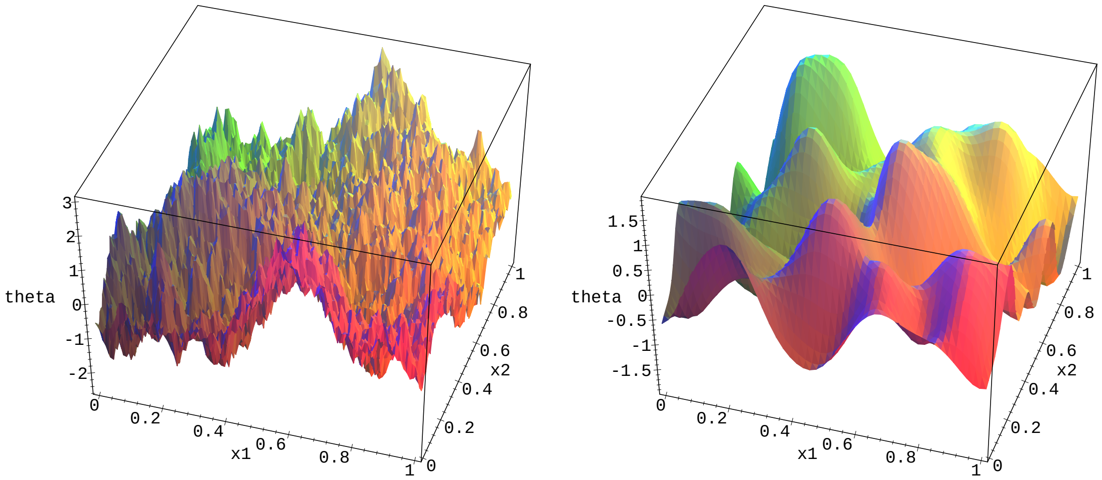

Figure 1: Samples drawn from Gaussian process priors over functions on $[0,1]^2$. Left: OU covariance function, $C(x,x') = \exp(-||x-x'||/l)$. Right: RBF covariance function, $C(x,x') = \exp[-||x-x'||^2/(2l^2)]$. The length scale $l = 0.1$ determines in both cases over what distance the functions vary significantly. Note the difference in roughness of the two functions; this is related to the behaviour of the covariance functions for $x \to x'$.

$$\langle y_l \theta(x) \rangle \quad = \quad C(x_l, x) \quad = \quad (\mathbf{k}(x))_l$$

where we 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 simply conditioning on the $\{y_l\}$. It is again Gaussian and has mean

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

and variance

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

Eqs. (1,2) solve the inference problem for Gaussian process: They provide us directly with the predictive distribution $P(\theta(x)|x,D)$. The posterior variance, eq. (2), in fact also gives us the expected generalization error (or Bayes error) at x. Why? If the teacher is θ^*, the squared deviation between our mean prediction and the teacher output[1] is $(\hat{\theta}(x) - \theta^*(x))^2$; averaging this over the posterior distribution of teachers $P(\theta^*|D)$ just gives (2). The underlying assumption is that our assumed Gaussian process prior is the true one from which teachers are actually generated (and that we are using the correct noise model). Otherwise, the expected generalization error is larger and given by a more complicated expression (Williams and Vivarelli, 2000). In line with most other work on the subject, we only consider the 'correct prior' case in the following. Averaging the generalization error at x over the distribution of inputs gives then

$$\epsilon(D) = \langle \epsilon(x,D) \rangle_x = \left\langle C(x,x) - \mathbf{k}(x)^{\mathrm{T}} \mathbf{K}^{-1} \mathbf{k}(x) \right\rangle_x \tag{3}$$

This form of the generalization error, which is well known (Michelli and Wahba, 1981; Opper, 1997; Williams, 1998; Williams and Vivarelli, 2000), still depends on the training inputs; the fact that the training *outputs* have dropped out already is a signature of the fact that Gaussian processes are *linear* predictors (compare (1)). Averaging over data sets yields the quantity we are after,

$$\epsilon = \langle \epsilon(D) \rangle_D \,. \tag{4}$$

[1] One can also measure the generalization by the squared deviation between the prediction $\hat{\theta}(x)$ and the *noisy* teacher output; this simply adds a term σ^2 to eq. (3).

This average expected generalization error (we will drop the 'average expected' in the following) only depends on the number of training examples n; the function $\epsilon(n)$ is called the *learning curve*. Its exact calculation is difficult because of the joint average in eqs. (3,4) over the training inputs x_l and the test input x.

Before proceeding with our calculation of the learning curve $\epsilon(n)$, let us try to gain some intuitive insight into its dependence on n. Consider a simple example scenario, where inputs x are one-dimensional and drawn randomly from the unit interval $[0,1]$, with uniform probability. For the covariance function we choose an RBF form, $C(x,x') = \exp[-|x-x'|^2/(2l^2)]$ with $l = 0.1$. Here we have taken the prior variance $C(x,x)$ as unity; as seems realistic for most applications we assume the noise level to be much smaller than this, $\sigma^2 = 0.05$. Figure 2 illustrates the x-dependence of the generalization error $\epsilon(x,D)$ for a small training set ($n = 2$): Each of the examples has made a 'dent' in $\epsilon(x,D)$, with a shape that is similar to that of the covariance function[2]. Outside the dents, $\epsilon(x,D)$ still has essentially its prior value, $\epsilon(x,D) = 1$; at the centre of each dent it is reduced to a much smaller value, $\epsilon(x,D) \approx \sigma^2/(1+\sigma^2)$ (this approximation holds as long as the different training inputs are sufficiently far away from each other). The generalization error $\epsilon(D)$ is therefore dominated by regions where no training examples have been seen; one has $\epsilon(D) \gg \sigma^2$, and the precise value of $\epsilon(D)$ depends only very little on σ^2 (assuming always that $\sigma^2 \ll 1$). Gradually, as n is increased, the covariance dents will cover the input space, so that $\epsilon(x,D)$ and $\epsilon(D)$ become of order σ^2; this situation is shown on the right of Figure 2. From this point onwards, further training examples essetially have the effect of averaging out noise, eventually making $\epsilon(D) \ll \sigma^2$ for large enough n. In summary, we expect the learning curve $\epsilon(n)$ to have two regimes: In the initial (small n) regime, where $\epsilon(n) \gg \sigma^2$, $\epsilon(n)$ is essentially independent of σ^2 and reflects mainly the geometrical distribution of covariance dents across the inputs space. In the asymptotic regime (n large enough such that $\epsilon(n) \ll \sigma^2$), on the other hand, the noise level σ^2 is important in controlling the size of $\epsilon(n)$ because learning arises mainly from the averaging out of noise in the training data.

The remainder of this paper is structured as follows. In the next section, we derive several approximations to GP learning curves, starting from an exact representation in terms of the eigenvalues and eigenfunctions of the covariance function. In Sec. 3, we compare these approximations to existing bounds and find, across a range of learning scenarios, that they generally get rather closer to the true learning curves. This section also includes two extensions of existing bounds which enlarge their domain of applicability. In Sec. 4 we investigate the potential for improving the accuracy of our approximations further. Finally, Sec. 5 deals with the question of whether there are limits of principle on the quality of learning curve approximations and bounds that only take the eigenvalues of the covariance function into account; we find that such limits do indeed exist. We conclude with a summary of our results and an overview of the challenges that remain.

2 Approximate learning curves

Calculating learning curves for Gaussian processes exactly is a difficult problem because of the joint average in (3,4) over the training inputs x_l and the test input x. Several workers have therefore derived upper and lower bounds on ϵ (Michelli and Wahba, 1981; Plaskota, 1990; Opper, 1997; Williams and Vivarelli, 2000) or studied the large n asymptotics of $\epsilon(n)$ (Silverman, 1985; Ritter, 1996). As we will illustrate below, however, the existing bounds are often far from tight; likewise, asymptotic results can only capture the large n regime defined above and will not necessarily apply for sample sizes n occurring in practice. We therefore now attempt to derive approximations to $\epsilon(n)$ which get closer to the true learning curves than existing bounds, and which are applicable both for small and large n.

As a starting point for an approximate calculation of $\epsilon(n)$, we first derive a representation of the generalization error in terms of the eigenvalue decomposition of the covariance function. Mercer's theorem (see *e.g.* (Wong, 1971)) tells us that the covariance function can be decomposed into its

[2]More precisely, the dents have the shape of the *square* of the covariance function: If the training inputs x_i are sufficiently far apart, then around each x_i we can neglect the influence of the other data points and apply (2) with $n = 1$, giving $\epsilon(x,D) \approx C(x,x) - C^2(x,x_i)/[C(x_i,x_i) + \sigma^2]$.

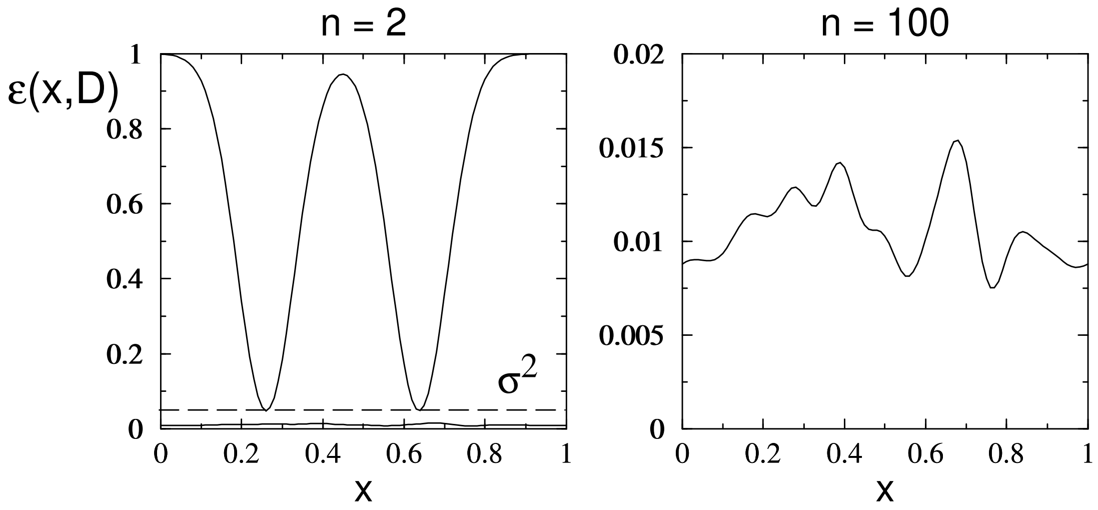

Figure 2: Generalization error $\epsilon(x, D)$ as a function of input position $x \in [0, 1]$, for noise level $\sigma^2 = 0.05$, RBF covariance function $C(x, x') = \exp[-|x - x'|^2/(2l^2)]$ with $l = 0.1$, for randomly drawn training sets D of size $n = 2$ (left) and $n = 100$ (right). To emphasize the difference in scale, the plot on the left actually also includes the results for $n = 100$, just visible below the dashed line at $\epsilon(x, D) = \sigma^2$.

eigenvalues λ_i and eigenfunctions $\phi_i(x)$:

$$C(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(x') \tag{5}$$

This is simply the analogue of the eigenvalue decomposition of a finite symmetric matrix. We assume here that eigenvalues and eigenfunctions are defined relative to the distribution over inputs x, *i.e.*,

$$\langle C(x, x') \phi_i(x') \rangle_{x'} = \lambda_i \phi_i(x) \tag{6}$$

The eigenfunctions are then orthogonal with respect to the same distribution, $\langle \phi_i(x) \phi_j(x) \rangle_x = \delta_{ij}$ (see *e.g.* (Williams and Seeger, 2000)). Now write the data-dependent generalization error (3) as

$$\epsilon(D) = \langle C(x, x) \rangle_x - \mathrm{tr}\, \left\langle \mathbf{k}(x) \mathbf{k}(x)^{\mathrm{T}} \right\rangle_x \mathbf{K}^{-1}$$

and perform the x-average:

$$\left\langle (\mathbf{k}(x)\mathbf{k}(x)^{\mathrm{T}})_{lm} \right\rangle_x = \sum_{ij} \lambda_i \lambda_j \phi_i(x_l) \left\langle \phi_i(x) \phi_j(x) \right\rangle_x \phi_j(x_m) = \sum_i \lambda_i^2 \phi_i(x_l) \phi_i(x_m).$$

This suggests introducing the diagonal matrix $(\mathbf{\Lambda})_{ij} = \lambda_i \delta_{ij}$ and the 'design matrix' $(\mathbf{\Psi})_{li} = \phi_i(x_l)$, so that $\left\langle \mathbf{k}(x)\mathbf{k}(x)^{\mathrm{T}} \right\rangle_x = \mathbf{\Psi} \mathbf{\Lambda}^2 \mathbf{\Psi}^{\mathrm{T}}$. One then also has

$$\langle C(x, x) \rangle_x = \mathrm{tr}\, \mathbf{\Lambda} \tag{7}$$

and the matrix $\mathbf{K}$ is expressed as

$$\mathbf{K} = \sigma^2 \mathbf{I} + \mathbf{\Psi} \mathbf{\Lambda} \mathbf{\Psi}^{\mathrm{T}}$$

with $\mathbf{I}$ being the identity matrix. Collecting these results, we have

$$\epsilon(D) = \mathrm{tr}\, \mathbf{\Lambda} - \mathrm{tr}\, (\sigma^2 \mathbf{I} + \mathbf{\Psi} \mathbf{\Lambda} \mathbf{\Psi}^{\mathrm{T}})^{-1} \mathbf{\Psi} \mathbf{\Lambda}^2 \mathbf{\Psi}^{\mathrm{T}}$$

Now one can apply the Woodbury formula (Press et al., 1992), which states that $(\mathbf{A}+\mathbf{U}\mathbf{V}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I}+\mathbf{V}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{A}^{-1}$ for matrices $\mathbf{A}$, $\mathbf{U}$, $\mathbf{V}$ of appropriate size. Setting $\mathbf{A}=\mathbf{\Lambda}^{-1}$, $\mathbf{U}=\mathbf{V}=\sigma^{-1}\mathbf{\Psi}$ and using the cyclic invariance of the trace, one then finds[3] $\epsilon(n) = \langle \epsilon(D) \rangle_D$ with

$$\epsilon(D) = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + \sigma^{-2}\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi})^{-1} = \mathrm{tr}\,\mathbf{\Lambda}(\mathbf{I}+\sigma^{-2}\mathbf{\Lambda}\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi})^{-1} \tag{8}$$

The advantage of this (still exact) representation of the generalization error is that the average over the test input x has already been carried out, and that the remaining dependence on the training data is contained entirely in the matrix $\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi}$. It also includes as a special case the well-known result for linear regression (see *e.g.* (Sollich, 1994)); $\mathbf{\Lambda}^{-1}$ and $\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi}$ can be interpreted as suitably generalized versions of the weight decay (matrix) and input correlation matrix.

Starting from (8), one can now derive approximate expressions for the learning curve $\epsilon(n)$. The most naive approach is to neglect entirely the fluctuations in $\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi}$ over different data sets and replace it by its average, which is simply $\left\langle (\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi})_{ij} \right\rangle_D = \sum_{l=1}^{n} \langle \phi_i(x_l)\phi_j(x_l)\rangle_D = n\delta_{ij}$. This leads to

$$\epsilon_{\mathrm{OV}}(n) = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + \sigma^{-2} n \mathbf{I})^{-1}. \tag{9}$$

While this is not, in general, a good *approximation*, it was shown by Opper and Vivarelli to be a lower *bound* (called OV bound below) on the learning curve (Opper and Vivarelli, 1999). It becomes tight in the large noise limit $\sigma^2 \to \infty$ at constant n/σ^2: The fluctuations of the elements of the matrix $\sigma^{-2}\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi}$ then become vanishingly small (of order $\sqrt{n}\sigma^{-2} = (n/\sigma^2)/\sqrt{n} \to 0$) and so replacing $\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi}$ by its average is justified. By a similar argument, one also expects that (for any fixed $\sigma^2 > 0$) the OV bound will become increasing tight as n increases.

To derive better approximations, it is useful to see how the matrix $\mathcal{G} = (\mathbf{\Lambda}^{-1} + \sigma^{-2}\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi})^{-1}$ changes when a new example is added to the training set. This change can be expressed as

$$\begin{aligned} \mathcal{G}(n{+}1) - \mathcal{G}(n) &= \left[\mathcal{G}^{-1}(n) + \sigma^{-2}\boldsymbol{\psi}\boldsymbol{\psi}^{\mathrm{T}}\right]^{-1} - \mathcal{G}(n) \\ &= -\frac{\mathcal{G}(n)\boldsymbol{\psi}\boldsymbol{\psi}^{\mathrm{T}}\mathcal{G}(n)}{\sigma^2 + \boldsymbol{\psi}^{\mathrm{T}}\mathcal{G}(n)\boldsymbol{\psi}} \end{aligned} \tag{10}$$

in terms of the vector $\boldsymbol{\psi}$ with elements $(\boldsymbol{\psi})_i = \phi_i(x_{n+1})$. To get exact learning curves, one would have to average this update formula over both the new training input x_{n+1} and all previous ones. This is difficult, but progress can be made by neglecting correlations of numerator and denominator in (10), averaging them separately instead. Also treating n as a continuous variable, this yields the approximation

$$\frac{\partial \mathbf{G}(n)}{\partial n} = -\frac{\left\langle \mathcal{G}^2(n) \right\rangle}{\sigma^2 + \mathrm{tr}\,\mathbf{G}(n)} \tag{11}$$

where we have introduced the notation $\mathbf{G} = \langle \mathcal{G} \rangle$. If we also neglect fluctuations in $\mathcal{G}$, approximating $\left\langle \mathcal{G}^2 \right\rangle = \mathbf{G}^2$, we have $\partial \mathbf{G}/\partial n = -\mathbf{G}^2/(\sigma^2 + \mathrm{tr}\,\mathbf{G})$. Since at $n=0$, the solution $\mathbf{G} = \mathbf{\Lambda}$ is diagonal, it will remain so for all n, and one can rewrite the equation for $\mathbf{G}(n)$ as $\partial \mathbf{G}^{-1}/\partial n = (\sigma^2 + \mathrm{tr}\,\mathbf{G})^{-1}\mathbf{I}$. $\mathbf{G}^{-1}(n)$ thus differs from $\mathbf{G}^{-1}(0) = \mathbf{\Lambda}^{-1}$ only by a multiple of the identity matrix, and can be represented as $\mathbf{G}^{-1}(n) = \mathbf{\Lambda}^{-1} + \sigma^{-2} n' \mathbf{I}$, where n' has to obey $\sigma^{-2} dn'/dn = [\sigma^2 + \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + \sigma^{-2} n' \mathbf{I})]^{-1}$. Integrating, one finds for n' the equation

$$n' + \mathrm{tr}\,\ln(\mathbf{I} + \sigma^{-2} n' \mathbf{\Lambda}) = n.$$

and the generalization error is given by

$$\epsilon_{\mathrm{UC}}(n) = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + \sigma^{-2} n' \mathbf{I})^{-1} \tag{12}$$

By comparison with (9), n' can be thought of as an 'effective number of training examples'.

The subscript UC in (12) stands for Upper Continuous (*i.e.*, treating n as continuous) approximation. A better approximation with a lower value is obtained by retaining fluctuations in $\mathcal{G}$. As

[3] If the covariance function has zero eigenvalues, the inverse $\mathbf{\Lambda}^{-1}$ does not exist, and the second form of $\epsilon(D)$ given in (8) must be used; similar alternative forms, though not explicitly written, exist for all following results.

in the case of the linear perceptron (Sollich, 1994), this can be achieved by introducing an auxiliary offset parameter v into the definition of

$$\mathcal{G}^{-1} = v\mathbf{I} + \mathbf{\Lambda}^{-1} + \sigma^{-2}\mathbf{\Psi}^{\mathrm{T}}\mathbf{\Psi}. \tag{13}$$

One can then write

$$-\mathrm{tr}\,\left\langle \mathcal{G}^2 \right\rangle = \frac{\partial}{\partial v}\mathrm{tr}\,\langle \mathcal{G} \rangle = \partial\epsilon/\partial v \tag{14}$$

and obtains from (11) the partial differential equation

$$\frac{\partial \epsilon}{\partial n} - \frac{1}{\sigma^2 + \epsilon}\frac{\partial \epsilon}{\partial v} = 0 \tag{15}$$

This can be solved for $\epsilon(n, v)$ using the methods of characteristic curves (see App. A). Resetting the auxiliary parameter v to zero yields the *L*ower *C*ontinuous approximation to the learning curve, which is given by the self-consistency equation

$$\epsilon_{\mathrm{LC}}(n) = \mathrm{tr}\,\left(\mathbf{\Lambda}^{-1} + \frac{n}{\sigma^2 + \epsilon_{\mathrm{LC}}}\mathbf{I}\right)^{-1}. \tag{16}$$

It is easy to show that $\epsilon_{\mathrm{LC}} \leq \epsilon_{\mathrm{UC}}$. One can also check that both approximations converge to the exact result (9) in the large noise limit (as defined above). Encouragingly, we see that the LC approximation reflects our intuition about the difference between the initial and asymptotic regimes of the learning curve: For $\epsilon \gg \sigma^2$, we can simplify (16) to

$$\epsilon_{\mathrm{LC}}(n) = \mathrm{tr}\,\left(\mathbf{\Lambda}^{-1} + \frac{n}{\epsilon_{\mathrm{LC}}}\mathbf{I}\right)^{-1}$$

where as expected the noise level σ^2 has dropped out. In the opposite limit $\epsilon \ll \sigma^2$, on the other hand, we have

$$\epsilon_{\mathrm{LC}}(n) = \mathrm{tr}\,\left(\mathbf{\Lambda}^{-1} + \frac{n}{\sigma^2}\mathbf{I}\right)^{-1} \tag{17}$$

which—again as expected—retains the noise level σ^2 as an important parameter. Eq. (17) also shows that ϵ_{LC} approaches the OV lower bound (from above) for sufficiently large n.

We conclude this section with a brief qualitative discussion of the expected n-dependence of ϵ_{LC} (which below will turn out to be the more accurate of our two approximations). Obviously this n-dependence depends on the spectrum of eigenvalues λ_i; below we always assume that these are arranged in decreasing order. Consider then first the asymptotic regime $\epsilon \ll \sigma^2$, where ϵ_{LC} and ϵ_{OV} become identical. One then shows easily that for eigenvalues decaying as a power-law, $\lambda_i \sim i^{-r}$, the asymptotic learning curve scales as[4] $\epsilon_{\mathrm{LC}} \sim (n/\sigma^2)^{-(r-1)/r}$; this is in agreement with known exact results (Silverman, 1985; Ritter, 1996). In the initial regime $\epsilon \gg \sigma^2$, on the other hand, one can take $\sigma^2 \to 0$ and finds then a faster decay[5] of the generalization error, $\epsilon_{\mathrm{LC}} \sim n^{-(r-1)}$. We are not aware of exact results pertaining to this regime, except for the OU case in $d = 1$ which has $r = 2$ and for which thus $\epsilon_{\mathrm{LC}} \sim n^{-1}$, in agreement with an exact calculation (Manfred Opper, private communication).

3 Comparisons with bounds and numerical simulations

We now compare the LC and UC approximations with existing bounds, and with the 'true' learning curves as obtained by numerical simulations. A lower bound on the generalization error was given

[4]Among the scenarios studied in the next section, the OU covariance function provides a concrete example of this kind of behaviour: In $d = 1$ dimension it has, from (44), $\lambda_i \sim i^{-2}$ and thus $r = 2$. The RBF covariance function, on the other hand, has eigenvalues decaying faster than any power law, corresponding to $r \to \infty$ and thus $\epsilon_{\mathrm{LC}} \sim \sigma^2/n$ (up to logarithmic corrections) in the asymptotic regime.

[5]For covariance functions with eigenvalues decaying faster than a power law, the behaviour in the initial regime is nontrivial; for the RBF covariance function in $d = 1$ with uniform inputs, for example, we find (for large n) $\epsilon_{\mathrm{LC}} \sim n\exp(-cn^2)$ with some constant c.

by Michelli and Wahba (Michelli and Wahba, 1981) as

$$\epsilon(n) \geq \epsilon_{\mathrm{MW}}(n) = \sum_{i=n+1}^{\infty} \lambda_i \tag{18}$$

This bound is derived for the noiseless case by allowing 'generalized observations' (projections of $\theta^*(x)$ along the first n eigenfunctions of $C(x, x')$), and so is unlikely to be tight for the case of 'real' observations at discrete input points. Given that the bound is derived from the $\sigma^2 \to 0$ limit, it can only be useful in the initial (small n, $\epsilon \gg \sigma^2$) regime of the learning curve. There, it confirms the conclusion of our intuitive discussion above that the learning curve has a lower limit below which it will not drop even for $\sigma^2 \to 0$.

Plaskota (Plaskota, 1990) generalized the MW approach to the noisy case and obtained the following improved lower bound:

$$\epsilon(n) \geq \epsilon_{\mathrm{Pl}}(n) = \min_{\{\eta_i\}} \sum_{i=1}^{n} \frac{\lambda_i \sigma^2}{\eta_i + \sigma^2} + \sum_{i=n+1}^{\infty} \lambda_i \tag{19}$$

where the minimum is over all non-negative $\eta_1, \ldots, \eta_n$ obeying $\sum_{i=1}^{n} \eta_i = S \equiv \sum_{l=1}^{n} C(x_l, x_l)$. Plaskota only derived this bound for covariance functions for which the prior variance $C(x, x)$ is independent of x. We call these 'uniform' covariance functions; due to the general identity $\langle C(x, x) \rangle_x = \mathrm{tr}\,\mathbf{\Lambda}$, they obey $C(x, x) = \mathrm{tr}\,\mathbf{\Lambda}$ (and hence $S = n\,\mathrm{tr}\,\mathbf{\Lambda}$). We recap Plaskota's proof in App. B and also show there that it extends to general covariance functions in the form stated above. The Plaskota bound is close to the MW bound in the small n regime (equivalent to $\sigma^2 \to 0$); for larger n it becomes substantially larger. It therefore has the potential to be useful for both small and large n. Note that, in contrast to all other bounds discussed in this paper, the MW and Plaskota bounds are in fact 'single data set' (worst case) bounds: They apply to $\epsilon(D)$ for *any* data set D of the given size n, rather than just to the average[6] $\epsilon(n) = \langle \epsilon(D) \rangle$.

Opper used information theoretic methods to obtain a different lower bound (Opper, 1997), but we will not consider this because the more recent OV bound (9) is always tighter. Note that the OV bound incorrectly suggest that ϵ decreases to zero for $\sigma^2 \to 0$ at fixed n. It therefore becomes void for small n (where $\epsilon \gg \sigma^2$) and is expected to be of use only in the asymptotic regime of large n.

There is also an *U*pper bound due to *O*pper (Opper, 1997),

$$\tilde{\epsilon}(n) \leq \epsilon_{\mathrm{UO}}(n) = (\sigma^{-2} n)^{-1}\, \mathrm{tr}\, \ln(\mathbf{I} + \sigma^{-2} n \mathbf{\Lambda}) + \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + \sigma^{-2} n \mathbf{I})^{-1} \tag{20}$$

Here $\tilde{\epsilon}$ is a modified version of ϵ which (in the rescaled version that we are using) becomes identical to ϵ in the limit of small generalization errors ($\epsilon \ll \sigma^2$), but never gets larger that $2\sigma^2$; for small n in particular, $\epsilon(n)$ can therefore actually be much *larger* than $\tilde{\epsilon}(n)$ and its bound (20). For this reason, and because in our simulations we never get very far into the asymptotic regime $\epsilon \ll \sigma^2$, we do not display the UO bound in the graphs below.

The UO bound is complemented by an upper bound due to *W*illiams and *V*ivarelli (Williams and Vivarelli, 2000), which never decays below values around σ^2 and is therefore mainly useful in the initial regime $\epsilon \gg \sigma^2$. It applies for one-dimensional inputs x and stationary covariance functions—for which $C(x, x') = C_s(x - x')$ is a function of $x - x'$ alone—and reads:

$$\epsilon(n) \leq \epsilon_{\mathrm{WV}}(n) = C_s(0) - \frac{1}{C_s(0) + \sigma^2} \int_0^{\infty} da\, f_n(a)\, C_s^2(a) \tag{21}$$

with

$$f_n(a) = 2(1-a)^n \Theta(1-a) + 2(n-1)(1-2a)^n \Theta(1-2a) \tag{22}$$

and where the Heaviside step functions Θ (defined as $\Theta(z) = 1$ for $z > 0$ and $= 0$ otherwise) in the two terms imply that only values of a up to 1 and $1/2$, respectively, contribute to the integral in (21).

[6] Our generalized version of the Plaskota bound depends on the specific data set only through the value of $S = \sum_{l=1}^{n} C(x_l, x_l)$. To obtain an average case bound one would need to average over the distribution of S.

The function $f_n(a)$ is a normalized distribution over a which for $n \to \infty$ becomes peaked around $a = 0$, implying that the asymptotic value of the bound is $\epsilon_{\mathrm{WV}}(n \to \infty) = C_s(0)\sigma^2/[C_s(0) + \sigma^2] \approx \sigma^2$ for $\sigma^2 \ll C_s(0)$. The derivation of the bound is based on the insight that $\epsilon(x, D)$ always decreases as more examples are added; it can therefore be upper bounded for any given x by the smallest $\epsilon(x, D')$ that would result from training on any data set D' comprising only a *single* example from the original training set D. The idea can be generalized to using the smallest $\epsilon(x, D')$ obtainable from any *two* of the training examples, but this does not significantly improve the bound (Williams and Vivarelli, 2000).

As stated above in (21,22), the WV bound applies only to the case of a uniform input distributions over the unit interval $[0, 1]$. However, it is relatively straightforward to extend the approach to general (one-dimensional) input distributions $P(x)$; only the data set average becomes technically a little more complicated. We omit the details and only quote the result: Eq. (21) remains valid if the expression (22) for $f_n(a)$ is generalized to

$$\begin{aligned} f_n(a) &= n \left\langle P(x-a)[1 - Q(x)]^{n-1} + P(x+a)[Q(x)]^{n-1} \right\rangle_x \\ &\quad + n(n-1) \left\langle \Theta(x' - x - 2a)[P(x+a) + P(x'-a)][1 + Q(x) - Q(x')]^{n-2} \right\rangle_{x,x'} \end{aligned} \tag{23}$$

where $Q(z) = \int_{-\infty}^{z} dx\ P(x)$ is the cumulative distribution function. In the simpler scenario considered by Williams and Vivarelli this can be shown to reduce to (22), while in the most general case the numerical evaluation of the bound requires a triple integral (over x, x' and a).

Finally, there is one more upper bound, due to *T*recate, *W*illiams and *O*pper (Trecate et al., 1999); based on the generalization error achieved by a 'suboptimal' Gaussian regressor, they showed

$$\epsilon(n) \le \epsilon_{\mathrm{TWO}}(n) = \mathrm{tr}\,\mathbf{\Lambda} - n \sum_i \frac{\lambda_i}{c_i}$$

where

$$c_i = (n-1)\lambda_i + \sigma^2 + \left\langle C(x,x)\phi_i^2(x) \right\rangle_x. \tag{24}$$

For a uniform covariance function, the average in c_i becomes $\mathrm{tr}\,\mathbf{\Lambda} \left\langle \phi_i^2(x) \right\rangle_x = \mathrm{tr}\,\mathbf{\Lambda}$ and the bound simplifies to

$$\epsilon_{\mathrm{TWO}}(n) = \sum_i \lambda_i \frac{\mathrm{tr}\,\mathbf{\Lambda} + \sigma^2 - \lambda_i}{\mathrm{tr}\,\mathbf{\Lambda} + \sigma^2 + (n-1)\lambda_i}$$

We now compare the quality of these bounds and our approximations with numerical simulations of learning curves. All the theoretical expressions require knowledge of the eigenvalue spectrum of the covariance function, so we focus on situations where this is known analytically. We consider three scenarios: For the first two, we assume that inputs x are drawn from the d-dimensional unit hypercube $[0, 1]^d$, and that the input density is uniform. As covariance functions we use the RBF function $C(x, x') = \exp[-||x - x'||^2/(2l^2)]$ and the OU function $C(x, x') = \exp(-||x - x'||/l)$, to have the extreme cases of smooth and rough functions to be learned; both contain a tunable length scale l. To be precise, we use slightly modified versions of the RBF and OU covariance functions (using what physicists call 'periodic boundary conditions') which make the eigenvalue calculations analytically tractable; the details are explained in App. C. In the third scenario we explore the effect of a non-uniform input distribution by considering inputs x drawn from a d-dimensional (zero mean) isotropic Gaussian distribution $P(x) \propto \exp[-||x||^2/(2\sigma_x^2)]$, for an RBF covariance function. Details of the eigenvalue spectrum for this case can also be found in App. C. Note that in all three cases, the covariance function is uniform, *i.e.* has a constant variance $C(x, x)$; we have fixed this to unity without loss of generality. This leaves three variable parameters: the input space dimension d, the noise level σ^2 and the length scale l. As explained above we generically expect the prior variance to be significantly larger than the noise on the training data, so we only consider values of $\sigma^2 < 1$. The length scale l should also obey $l < 1$; otherwise the covariance functions $C(x, x')$ would be almost constant across the input space, corresponding to a trivial GP prior of essentially x-independent functions. We in fact choose the length scale l for each d in such a way as to get a reasonable decay of the learning curve within the range of $n = 0 \ldots 300$ that can be

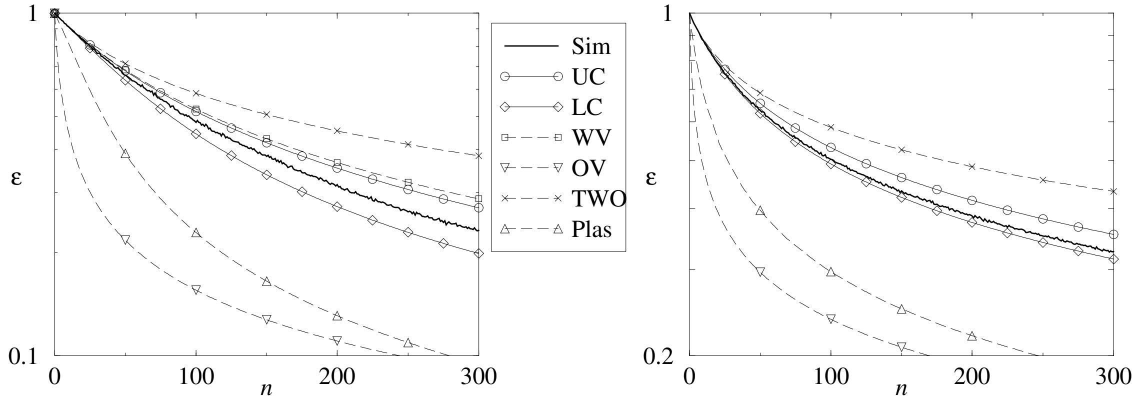

Figure 3: Learning curve for a GP with OU covariance function and inputs uniformly drawn from $x \in [0,1]^d$, at noise level $\sigma^2 = 0.05$. Left: $d = 1$, length scale $l = 0.01$. Right: $d = 2$, $l = 0.1$.

conveniently simulated numerically. To see why this is necessary, note that each covariance 'dent' covers a fraction of order l^d of the input space, so that the number of examples n needed to see a significant reduction in generalization error ϵ will scale as $(1/l)^d$. This quickly becomes very large as d increases unless l is increased simultaneously. (The effect of larger l leading to a faster decay of the learning curve was also observed in (Williams and Vivarelli, 2000).)

In the following figures, we show the lower bounds (Plaskota, OV), the non-asymptotic upper bounds (TWO and, for $d = 1$ with uniform input distribution, WV), and our approximations (LC and UC). The true learning curve as obtained from numerical simulations is also shown. For the numerical simulations, we built up training sets by randomly drawing training inputs from the specified input distribution. For each new training input, the matrix inverse $\mathbf{K}^{-1}$ has to be recalculated. By partitioning the matrix into its elements corresponding to the old and new inputs, this inversion can be performed with $\mathcal{O}(n^2)$ operations (see *e.g.* (Press et al., 1992)), as opposed to $\mathcal{O}(n^3)$ if the inverse is calculated from scratch every time. With $\mathbf{K}^{-1}$ known, the generalization error $\epsilon(D)$ was then calculated from (3), with the average over x estimated by an average over randomly sampled test inputs. This process was repeated up to our chosen $n_{\max} = 300$; the results for $\epsilon(D)$ were then averaged over a number of training set realizations to obtain the learning curve $\epsilon(n)$. In all the graphs shown, the size of the error bars on the simulated learning curve is of the order of the visible fluctuations in the curve.

In Figure 3 we show the results for an OU covariance function with inputs from $[0,1]^d$, for $d = 1$ (left) and $d = 2$ (right). One observes that the lower bounds (Plaskota and OV) are rather loose in both cases. The TWO upper bound is also far from tight; the WV upper bound is better where it can be defined (for $d = 1$). Our approximations, LC and UC, are closer to the true learning curve than any of the bounds and in fact appear to bracket it.

Similar comments apply to Figure 4, which displays the results for an RBF covariance function with inputs from $[0,1]$. Because functions from an RBF prior are much smoother than those from an OU prior, they are easier to learn and the generalization error ϵ shows a more rapid decrease with n. This makes visible, within the range of n shown, the anticipated change in behaviour as ϵ crosses over from the initial ($\epsilon \gg \sigma^2$) to the asymptotic ($\epsilon \ll \sigma^2$) regime. The LC approximation and, to a less quantitative extent, the Plaskota bound both capture this change. By contrast, the OV bound (as expected from its general properties discussed above) only shows the right qualitative behaviour in the asymptotic regime.

Figure 5 shows corresponding results in higher dimension ($d = 4$), at two different noise levels σ^2. One observes in particular that the OV lower bound becomes looser as σ^2 decreases; this is as expected since for $\sigma^2 \to 0$ the bound actually becomes void ($\epsilon_{\mathrm{OV}} \to 0$). The Plaskota bound also appears to get looser for lower σ^2, though not as dramatically. (Note that the kinks in the Plaskota curve are not an artifact: For larger d the multiplicities of the different eigenvalues can be quite

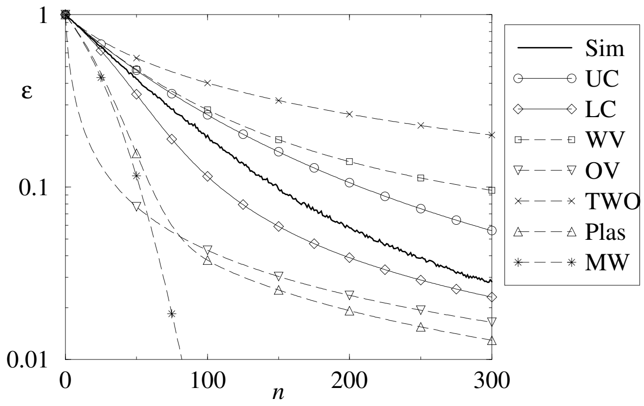

Figure 4: Learning curve for a GP with RBF covariance function and inputs uniformly drawn from $x \in [0,1]^d$, at noise level $\sigma^2 = 0.05$; dimension $d = 1$, length scale $l = 0.01$. We also show the MW bound here to show how it is first close to the Plaskota bound but then "misses" the change in behaviour where the generalization error crosses over into the asymptotic regime ($\epsilon \ll \sigma^2$).

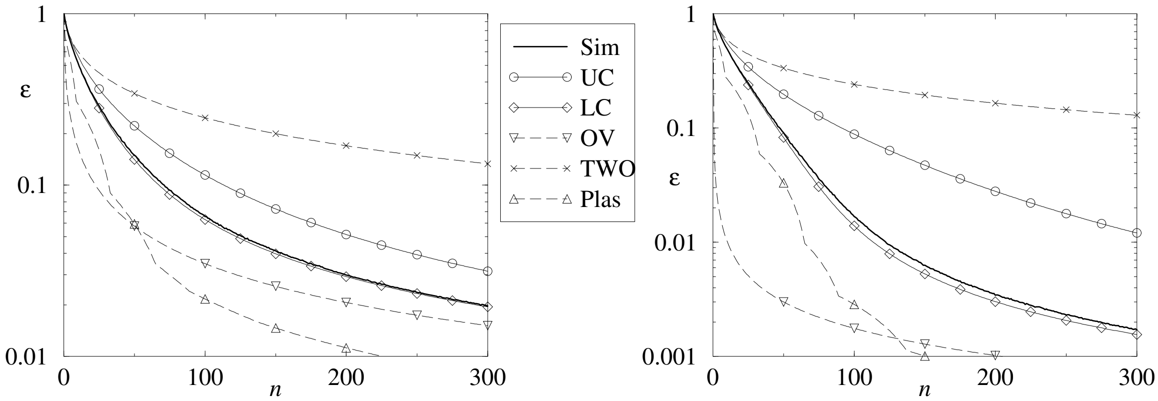

Figure 5: Learning curve for a GP with RBF covariance function and inputs uniformly drawn from $x \in [0,1]^d$, for dimension $d = 4$ and length scale $l = 0.3$. Left: noise level $\sigma^2 = 0.05$. Right: $\sigma^2 = 0.001$. Note that, for lower σ^2, the OV bound becomes looser as expected (it approaches zero for $\sigma^2 \to 0$).

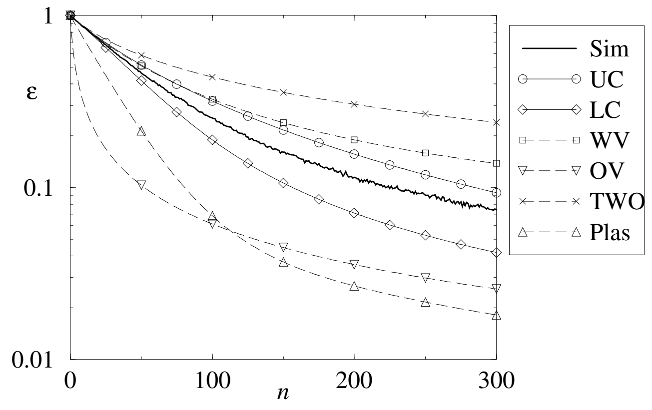

Figure 6: Learning curve for a GP with RBF covariance function (length scale $l = 0.01$) and inputs drawn from a Gaussian distribution in $d = 1$ dimension; noise level $\sigma^2 = 0.05$. Note that the LC approximation provides less of a good representation of the overall shape of the learning curve here than for the previous examples with uniform input distributions. The curve labelled WV shows our generalized version of the Williams-Vivarelli bound (see eq. (23)).

large; the value of ϵ_{Pl} can become dominated by one such block of degenerate eigenvalues, and kinks occur where the dominant block changes.) The TWO upper bound, finally, is only weakly affected by the value of σ^2 and quite loose throughout.

All results shown so far pertain to uniform input distributions (over $[0, 1]^d$). We now move to the last of our three scenarios, a GP with an RBF covariance function and inputs drawn from a Gaussian distribution (see App. C for details). In Figure 6 we see that in $d = 1$ the (generalized) WV bound is still reasonably tight, while the LC approximation now provides less of a good representation of the overall shape of the learning curve than for the case of uniform input distributions. However, as in all previous examples, the LC and UC approximations still bracket the true learning curve (and come closer to it than the bounds). One is thus lead to speculate whether the approximations we have derived are actually bounds. Figure 7 shows this not to be the case, however: In $d = 4$, the true learning curve drops visibly below the LC approximation in the small n regime, and so the latter cannot be a lower bound. The low noise case ($\sigma^2 = 0.001$) shown here illustrates once more that the OV lower bound ceases to be useful for small noise levels.

In summary, of the approximations which we have derived, the LC approximation performs best. While we know on theoretical grounds that it will be accurate for large noise levels σ^2, the examples shown above demonstrate that it produces predictions close to the true learning curves even for the more realistic case of low noise levels (compared to the prior variance). As a general trend, agreement appears to be better for the case of uniform input distributions.

It is interesting at this stage to make a connection to the recent work of Malzahn and Opper (Malzahn and Opper, 2001). They devised an elegant way of approaching the learning curve problem from the point of view of statistical physics, calculating the relevant partition function (which is an average over data sets) using a so-called Gaussian variational approximation. The result they find for the Bayes error is *identical* to the LC approximation under the condition that $\epsilon(x) = \langle\epsilon(x, D)\rangle_D$, the x-dependent generalization error averaged over all data sets, is independent of x. Otherwise, they find a result of the same functional form, $\epsilon(n) = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + \eta\mathbf{I})^{-1}$, but the self-consistency equation for η is more complicated than the simple relation $\eta = n/(\epsilon + \sigma^2)$ obtained from the LC approximation (16). The LC approximation would thus be expected to perform less well for such "non-uniform" scenarios. This agrees qualitatively with our above findings: For the scenario with a Gaussian input distribution, the LC approximation is of poorer quality than for the cases with uniform input distributions[7] over $[0, 1]^d$.

[7] It is easy to see that in these cases $\epsilon(x)$ is indeed independent of x; the absence of effects from the boundaries of

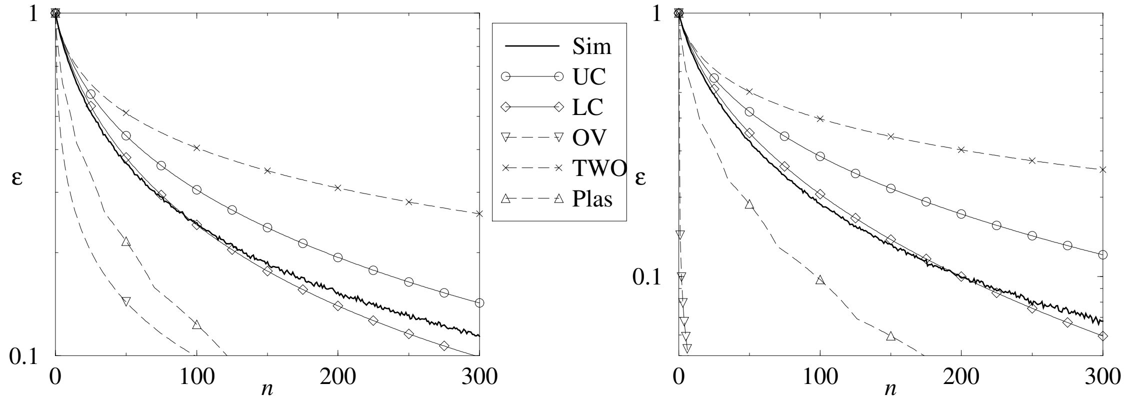

Figure 7: Learning curve for a GP with RBF covariance function (length scale $l = 0.3$) and inputs drawn from a Gaussian distribution in $d = 4$ dimensions. Left: noise level $\sigma^2 = 0.05$. Right: $\sigma^2 = 0.001$. The OV lower bound is very loose for this smaller noise level. Note that, in contrast to all previous examples, there is a range of n here where the true learning curve lies *below* the LC approximation.

4 Improving the approximations

In the previous section we saw that in our test scenarios the LC approximation (16) generally provides the closest theoretical approximation to the true learning curves. This may appear somewhat surprising, given that we made two rather drastic approximations in deriving (16): We treated the number of training examples n as a continuous variable, and we decoupled the average of the right hand side of (10) into separate averages over numerator and denominator. We now investigate whether the LC prediction for the learning curves can be further improved by removing these approximations.

We begin with the effect of n, the number of training examples, taking only discrete (rather than continuous) values. Starting from (10), averaging numerator and denominator separately as before and introducing the auxiliary variable v as in (13,14), we obtain

$$\epsilon(n+1) - \epsilon(n) = \frac{1}{\sigma^2 + \epsilon}\frac{\partial \epsilon}{\partial v} \tag{25}$$

instead of (15). It is possible to interpolate between the two equations by writing

$$\frac{1}{\delta}[\epsilon(n+\delta) - \epsilon(n)] = \frac{1}{\sigma^2 + \epsilon}\frac{\partial \epsilon}{\partial v} \tag{26}$$

Then $\delta = 1$ corresponds to (25), which is the equation we wish to solve (discrete n), while in the limit $\delta \to 0$ we retrieve (15). To proceed, we treat δ as a perturbation parameter and assume that the solution of (26) can be expanded as

$$\epsilon = \epsilon_0 + \delta\epsilon_1 + \mathcal{O}(\delta^2)$$

where $\epsilon_0 \equiv \epsilon_{\mathrm{LC}}$. Expanding both sides of (26) to first order in δ yields

$$\frac{\partial \epsilon_0}{\partial n} + \delta\frac{\partial \epsilon_1}{\partial n} + \frac{1}{2}\delta\frac{\partial^2 \epsilon_0}{\partial n^2} + \mathcal{O}(\delta^2) = \left(\frac{1}{\sigma^2 + \epsilon_0} - \delta\,\frac{\epsilon_1}{(\sigma^2+\epsilon_0)^2}\right)\left(\frac{\partial \epsilon_0}{\partial v} + \delta\,\frac{\partial \epsilon_1}{\partial v}\right)$$

Comparing the coefficients of the $\mathcal{O}(\delta^0)$ terms gives us back (15) for ϵ_0, while from the $\mathcal{O}(\delta)$ terms we get

$$\frac{\partial \epsilon_1}{\partial n} - \frac{1}{\sigma^2 + \epsilon_0}\frac{\partial \epsilon_1}{\partial v} = -\frac{1}{2}\frac{\partial^2 \epsilon_0}{\partial n^2} - \frac{\epsilon_1}{(\sigma^2+\epsilon_0)^2}\frac{\partial \epsilon_0}{\partial v} \tag{27}$$

the hypercube comes from the periodic boundary conditions that we are using.

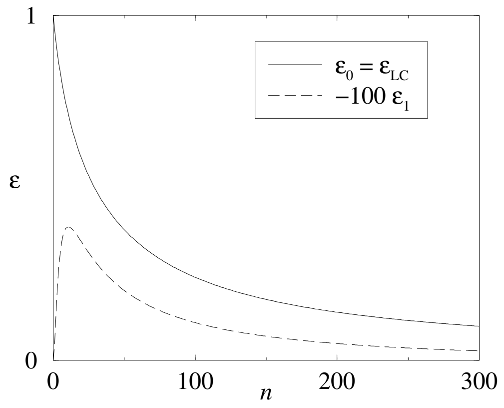

Figure 8: Solid line: LC approximation $\epsilon_{\rm LC}$ for the learning curve of a GP with RBF covariance function and Gaussian inputs. Parameters are as in Figure 7(left): Dimension $d = 4$, length scale $l = 0.3$, noise level $\sigma^2 = 0.05$. Dashed line: First order correction ϵ_1 arising from the discrete nature of the number of training examples n. Note that ϵ_1 has been multiplied by -100 to make it positive and visible on the scale of the values of $\epsilon_{\rm LC}$.

This can again be solved using characteristics (see App. A), with the result

$$\epsilon_1 = (\sigma^2 + \epsilon_0)\frac{n(a_2^2 - a_3)}{(1 - na_2)^2}, \qquad a_k = (\sigma^2 + \epsilon_0)^{-k}\ \mathrm{tr}\ \left(\mathbf{\Lambda}^{-1} + \frac{n}{\sigma^2 + \epsilon_0}\mathbf{I}\right)^{-k} \tag{28}$$

Setting δ back to 1 to have the case of discrete n in (26), we then have

$$\epsilon_{\rm LC1} = \epsilon_0 + \epsilon_1 \equiv \epsilon_{\rm LC} + \epsilon_1$$

as the improved LC approximation that takes into account the effects of discrete n (up to linear order in a perturbation expansion in δ). We see that the correction term (28) is zero at $n = 0$ as it must since $\epsilon_{\rm LC}$ gives the exact result $\epsilon = \mathrm{tr}\,\mathbf{\Lambda}$ there. It can also be shown that $\epsilon_1 < 0$ for all nonzero n. This can be understood as follows: The decrease of $\epsilon(n)$ becomes smaller (in absolute value) as n increases. Comparing (15) and (25), we see that the continuous n-approximation effectively averages the decrease term over the range $n \ldots n+1$ rather than evaluating it at n itself; it therefore produces a smaller decrease in $\epsilon(n)$. The true decrease for discrete n is larger, and so one expects the correction ϵ_1 to be negative, in agreement with our calculation.

In Figure 8 we show $\epsilon_{\rm LC}$ and ϵ_1 for one of the scenarios considered earlier; the results are typical also of what we find for other cases. The most striking observation is the smallness of ϵ_1: Its absolute value is of the order of 1% of $\epsilon_{\rm LC}$ or less, and consequently $\epsilon_{\rm LC}$ and $\epsilon_{\rm LC1}$ are indistinguishable on the scale of the plot. On the one hand, this is encouraging: Given that ϵ_1 is already so small, one would expect higher orders in a perturbation expansion in δ to yield even smaller corrections. Thus $\epsilon_{\rm LC1}$ is likely to be very close to the result that one would find if the discrete nature of n was taken into account exactly. On the other hand, we also conclude that treating n as discrete is *not* sufficient to turn the LC approximation into a lower bound on the learning curve; in Fig 6, for example, the curve for $\epsilon_{\rm LC1}$ would lie essentially on top of the one for $\epsilon_{\rm LC}$ and so still be significantly above the true learning curve for small n.

It is clear at this stage that in order to improve the LC approximation significantly one would have to address the decoupling of the numerator and denominator averages in (10). Generally, if a and b are random variables, one can evaluate the average of their ratio perturbatively as

$$\left\langle \frac{a}{b} \right\rangle = \left\langle \frac{\langle a\rangle + \Delta a}{\langle b\rangle + \Delta b} \right\rangle = \frac{\langle a\rangle}{\langle b\rangle} + \frac{\langle a\rangle}{\langle b\rangle^3}\left\langle (\Delta b)^2 \right\rangle - \frac{1}{\langle b\rangle^2}\left\langle \Delta a \Delta b\right\rangle + \ldots$$

up to second order in the fluctuations. (This idea was used in (Sollich, 1994) to calculate finite N-corrections to the $N \to \infty$ limit of a linear learning problem.) To apply this to the average of the right hand side of (10) over the new training input x_{n+1} and all previous ones, one would set

$$a = \mathcal{G}(n)\boldsymbol{\psi}\boldsymbol{\psi}^{\mathrm{T}}\mathcal{G}(n), \qquad b = \sigma^2 + \boldsymbol{\psi}^{\mathrm{T}}\mathcal{G}(n)\boldsymbol{\psi}$$

One then sees that averages such as $\langle ab\rangle$, required in $\langle \Delta a \Delta b\rangle = \langle ab\rangle - \langle a\rangle\langle b\rangle$, involve fourth-order averages $\langle \phi_i(x)\phi_j(x)\phi_k(x)\phi_l(x)\rangle_x$ of the components of $\boldsymbol{\psi}$. In contrast to the second-order averages $\langle \phi_i(x)\phi_j(x)\rangle = \delta_{ij}$, such fourth-order statistics of the eigenfunctions do not have a simple, covariance function-independent form. Even if they were known, however, one would end up with averages over the entries of the matrix $\mathcal{G}$ which cannot be reduced to $\epsilon = \mathrm{tr}\,\langle\mathcal{G}\rangle$ (*e.g.* by derivatives with respect to auxiliary parameters). Separate equations for the change of these averages with n would then be required, generating new averages and eventually an infinite hierarchy which cannot be closed. We thus conclude that a perturbative approach is of little use in improving the LC approximation beyond the decoupling of averages. The approach of (Malzahn and Opper, 2001) thus looks more hopeful as far as the derivation of systematic corrections to the approximation is concerned.

5 How good can bounds and approximations be?

In this final section we ask whether there are limits of principle on the quality of theoretical predictions (either bounds or approximations) for GP learning curves. Of course this question is meaningless unless we specify what information the theoretical curves are allowed to exploit. Guided by the insight that all predictions discussed above depend (at least for uniform covariance functions) on the eigenvalues of the covariance function only (and of course the noise level σ^2), we ask: How tight can bounds and approximations be if they use only this eigenvalue spectrum as input?

To answer this question, it is useful to have a simple scenario with an arbitrary eigenvalue spectrum for which learning curves can be calculated exactly. Consider the case where the input space consists of N discrete points x^α; the input distribution is arbitrary, $P(x^\alpha) = p_\alpha$ with $\sum_\alpha p_\alpha = 1$. Take the covariance function to be degenerate in the sense that there are no correlations between different points: $C(x^\alpha, x^\beta) = c_\alpha\delta_{\alpha\beta}$. The eigenvalue equation (6) then becomes simply

$$\sum_\beta C(x^\alpha, x^\beta)\phi(x^\beta)p_\beta = c_\alpha p_\alpha \phi(x^\alpha) = \lambda\phi(x^\alpha)$$

so that the N different eigenvalues are $\lambda_\alpha = c_\alpha p_\alpha$. The eigenfunctions are $\phi_\alpha(x^\beta) = p_\alpha^{-1/2}\delta_{\alpha\beta}$, where the prefactor follows from the normalization condition $\sum_\gamma p_\gamma\phi_\alpha(x^\gamma)\phi_\beta(x^\gamma) = \delta_{\alpha\beta}$. Note that by choosing the c_α and p_α appropriately, the λ_α can be adjusted to any desired value in this setup. The same still holds even if we require the covariance function to be uniform, *i.e.*, c_α to be independent of α. For this case, a simple connection can also be made to the covariance functions discussed in previous sections: If one imagines the points x^α arranged in some d-dimensional space, then the present covariance function can be viewed as an RBF covariance function $C(x, x') = \mathrm{const}\cdot\exp[-||x-x'||^2/(2l^2)]$ in which the correlation length scale l has been taken to zero, so that different input points have entirely uncorrelated outputs.

A set of n training inputs is, in this scenario, fully characterized by how often it contains each of the possible inputs x^α; we call these numbers n_α. The generalization error is easy to work out from (8), using

$$(\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})_{\alpha\beta} = \sum_\gamma n_\gamma\phi_\alpha(x^\gamma)\phi_\beta(x^\gamma) = (n_\alpha/p_\alpha)\delta_{\alpha\beta}.$$

This shows that $\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}$ is a diagonal matrix, and thus from (8)

$$\epsilon(D) = \mathrm{tr}\,(\boldsymbol{\Lambda}^{-1} + \sigma^{-2}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1} = \sum_\alpha\left(\lambda_\alpha^{-1} + \sigma^{-2}n_\alpha/p_\alpha\right)^{-1} = \sum_\alpha \lambda_\alpha\frac{\sigma^2}{\sigma^2 + n_\alpha\lambda_\alpha/p_\alpha} \tag{29}$$

This has the expected form: The contribution of each eigenvalue is reduced according to the ratio of the noise level σ^2 and the signal $n_\alpha\lambda_\alpha/p_\alpha = n_\alpha c_\alpha$ received at the corresponding input point. To average this over all training sets of size n, one notices that n_α has a binomial distribution, so that

$$\epsilon(n) = \sum_\alpha \lambda_\alpha \sum_{n_\alpha=0}^{n} \binom{n}{n_\alpha} p_\alpha^{n_\alpha}(1-p_\alpha)^{n-n_\alpha} \frac{\sigma^2}{\sigma^2 + n_\alpha\lambda_\alpha/p_\alpha}$$

Writing $(\sigma^2 + n_\alpha\lambda_\alpha/p_\alpha)^{-1} = \sigma^{-2}\int_0^1 dr\ r^{n_\alpha\lambda_\alpha/p_\alpha\sigma^2}$, we can perform the sum over n_α and obtain

$$\epsilon(n) = \int_0^1 dr\ \sum_\alpha \lambda_\alpha \left(1 - p_\alpha + p_\alpha r^{\lambda_\alpha/p_\alpha\sigma^2}\right)^n \tag{30}$$

as the final result; the integral over r can easily be performed numerically for given set of eigenvalues λ_α and input probabilities p_α. Note that, having done the calculation for a finite number N of discrete inputs points (and therefore of eigenvalues), we can now also take N to infinity and therefore analyse scenarios (such as the ones studied in Sec. 3) with an infinite number of nonzero eigenvalues.

A simple limiting case will now tell us about the quality of eigenvalue-dependent upper bounds on learning curves. Assume that one of the p_α is close to 1, whereas all the other ones are close to 0. From (30), one then sees that only the contribution from the eigenvalue λ_α with $p_\alpha \approx 1$ is reduced as n increases[8] while all other ones remain unaffected, so that

$$\epsilon(n) \approx \operatorname{tr}\boldsymbol{\Lambda} - \lambda_\alpha + \frac{\lambda_\alpha\sigma^2}{\sigma^2 + n\lambda_\alpha} = \operatorname{tr}\boldsymbol{\Lambda} - \frac{n\lambda_\alpha^2}{\sigma^2 + n\lambda_\alpha} \geq \operatorname{tr}\boldsymbol{\Lambda} - \lambda_\alpha \tag{31}$$

If $\lambda_\alpha \ll \operatorname{tr}\boldsymbol{\Lambda}$, then we can make the reduction in generalization error arbitrarily small. It thus follows that there is no non-trivial upper bound on learning curves that takes only the eigenvalue spectrum of the covariance function as input. [Accordingly, the two non-asymptotic upper bounds (WV and TWO) discussed in Sec. 3 both contain other information, via the weighted averages of $C_s^2(x) = C^2(0,x)$ in (21) and the average of $C(x,x)\phi_i^2(x)$ in (24).] In particular, this implies that our UC approximation cannot be an upper bound (even though the results for all scenarios investigated above suggested that it might be). Furthermore, our result shows that lower bounds on the generalization error (*e.g.* the OV or Plaskota bounds) can be arbitrarily loose. A similar observation holds for *upper* bounds on the (data set-averaged) *training* error

$$\epsilon_{\rm t} = \langle \epsilon_{\rm t}(D)\rangle_D\,, \qquad \text{where} \quad \epsilon_{\rm t}(D) = \frac{1}{n}\sum_{l=1}^{n}\left(\hat\theta(x_l, D) - y_l\right)^2$$

Opper and Vivarelli showed (Opper and Vivarelli, 1999) that their $\epsilon_{\rm OV}$ actually also provides an upper bound for this quantity (so that the two errors 'sandwich' the bound, $\epsilon_{\rm t} \leq \epsilon_{\rm OV} \leq \epsilon$). In the present case it is easy to calculate $\epsilon_{\rm t}$ explicitly; we omit details and just quote the result

$$\epsilon_{\rm t} \approx \frac{\lambda_\alpha\sigma^2}{\sigma^2 + n\lambda_\alpha} \leq \lambda_\alpha$$

By taking $\lambda_\alpha \ll \operatorname{tr}\boldsymbol{\Lambda}$, one then sees that the upper bound $\epsilon_{\rm OV}$ can be made arbitrarily loose for any fixed n (and that the ratio of training and generalization error can be made arbitrarily small).

One may object that the above limit of most of the p_α tending to zero is unrealistic because it implies that the corresponding prior variances $c_\alpha = \lambda_\alpha/p_\alpha$ would become very large. Let us therefore now restrict the prior variance to be uniform, $c_\alpha = c$. It then follows that $\lambda_\alpha = c/p_\alpha$ and hence $p_\alpha = \lambda_\alpha/\operatorname{tr}\boldsymbol{\Lambda}$. With this assumption, only the λ_α and σ^2 remain as parameters affecting the learning curve (30). The results for an eigenvalue spectrum from one of the situations covered in Sec. 3 are shown in Figure 9. The main conclusion to be drawn is that the learning curves for the present scenario are quite different from the ones we found earlier, even though the eigenvalue spectra and

[8] This holds if n is not too large, more precisely if $np_\beta \ll 1$ for all the 'small' p_β ($\beta \neq \alpha$).

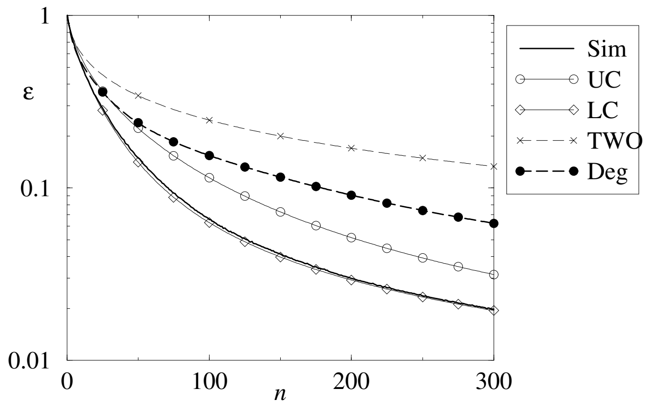

Figure 9: Learning curve for a GP with RBF covariance function and inputs uniformly drawn from $x \in [0,1]^d$. Parameters are as in Figure 5(left): Dimension $d = 4$, length scale $l = 0.3$ and noise level $\sigma^2 = 0.05$. The curves Sim (true learning curve, from numerical simulations), UC, LC and TWO are also as in Figure 5(left). The curve labelled "Deg" shows the exact learning curve for the degenerate scenario (outputs for different inputs are uncorrelated) with *exactly the same spectrum of eigenvalues* λ_i of the covariance function (and uniform prior variance $C(x,x)$). The curves Sim and Deg differ significantly, showing that learning curves cannot be predicted reliably based on eigenvalues alone.

noise levels are, by construction, precisely identical. This demonstrates that theoretical predictions for learning curves which only take into account the eigenvalue spectrum of a covariance function cannot universally match the true learning curves with a high degree of accuracy; the quality of approximation will vary depending on details of the covariance function and input distribution that are not encoded in the spectrum. Note that Figure 9 also provides a concrete example for the fact that the UC approximation is not in general an upper bound on the true learning curve; in fact, it here *underestimates* the true $\epsilon(n)$ quite significantly.

We can also use the present scenario to assess whether, as a bound on the generalization error resulting from a *single* training set, the Plaskota bound (19) could be significantly improved. We focus on the case of uniform covariance functions, where (29) becomes

$$\epsilon(D) = \sum_\alpha \lambda_\alpha \frac{\sigma^2}{\sigma^2 + n_\alpha \mathrm{tr}\,\mathbf{\Lambda}} \tag{32}$$

For any assignment of the n_α there is at least one training set of size $n = \sum_\alpha n_\alpha$ for which the generalization error is given by (32). Minimizing numerically over the n_α for each given n, we find the curves shown in Figure 10, where the Plaskota bounds for the same eigenvalue spectra are also shown. The curves are quite close to each other, implying that the Plaskota bound cannot be significantly tightened as a single data set bound (assuming, as throughout, that the improved bound would again only be based on the covariance function's eigenvalue spectrum). In the limit $\sigma^2 \to 0$, the bound—which then reduces to the MW bound (18)—cannot be tightened at all, as setting $n_\alpha = 1$ for $\alpha = 1 \ldots n$ and $n_\alpha = 0$ for $\alpha \geq n+1$ in (32) shows.

Within the simple degenerate scenario introduced in this section, we finally comment briefly on a universal relation recently proposed by Malzahn and Opper (Malzahn and Opper, 2001); Manfred Opper, private communication). They suggest considering an empirical estimate of the (Bayes) generalization error, which is obtained by replacing the average over all inputs x by one over the n training inputs x_i:

$$\epsilon_{\mathrm{emp}}(D) = \frac{1}{n}\sum_{i=1}^{n} \epsilon(x_i, D)$$

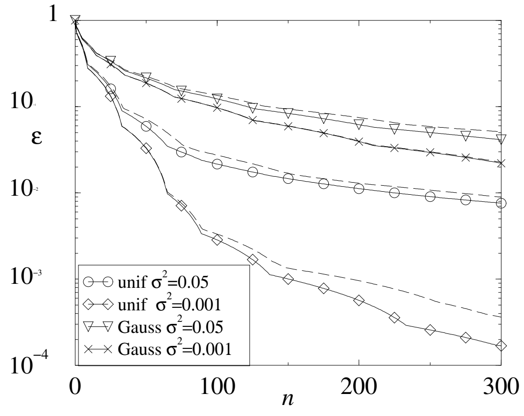

Figure 10: Comparison of the Plaskota bound (solid lines) and the lowest generalization error achievable for single data sets of size n within the degenerate scenario (dashed lines). The eigenvalue spectra used to construct the curves are those for an RBF covariance function with length scale $l = 0.3$, in $d = 4$ dimensions, and for the input distributions (uniform over $[0, 1]^d$, or Gaussian) shown in the legend; the noise level σ^2 is also given there. Note that for a given n, the curves become closer for lower σ^2; this is as expected since for $\sigma^2 \to 0$ the Plaskota bound can be saturated for a specific data set (see text).

Within the approximations of their calculation, the data set average of this quantity is then universally linked to a modified version of the true generalization error:

$$\langle \epsilon_{\text{emp}}(D) \rangle_D = \left\langle \frac{\epsilon(x)}{\sigma^2 + \epsilon(x)} \right\rangle_x \tag{33}$$

Note that the average over data sets is on the 'inside' of the fraction on the right hand side, through the definition of $\epsilon(x) = \langle \epsilon(x, D) \rangle_D$. Within our degenerate scenario, we can calculate both sides of (33) explicitly, but find no obvious relation between the two sides. However, if we move the data set average on the right hand side to the outside, we do (after a brief calculation, the details of which we omit) find a simple result:

$$\langle \epsilon_{\text{emp}}(D_{n+1}) \rangle_{D_{n+1}} = \left\langle \left\langle \frac{\epsilon(x, D_n)}{\sigma^2 + \epsilon(x, D_n)} \right\rangle_x \right\rangle_{D_n} \tag{34}$$

As indicated by the subscripts, the left hand side of this relation is to be evaluated for data sets of size $n + 1$ rather than n. The result (34) is remarkable in that it holds for any eigenvalue spectrum and any input distribution (within the degenerate scenario considered here). We take this as a hopeful sign that some universal link between true and empirical generalization errors, along the lines derived by (Malzahn and Opper, 2001) within their approximation, may indeed exist.

6 Conclusion

In summary, we have derived an exact representation of the average generalization error ϵ of Gaussian processes used for regression, in terms of the eigenvalue decomposition of the covariance function. Starting from this, we obtained two different approximations (LC and UC) to the learning curve $\epsilon(n)$. Both become exact in the large noise limit; in practice, one generically expects the opposite case ($\sigma^2 / C(x, x) \ll 1$), but comparison with simulation results shows that even in this regime the new approximations perform well.

The LC approximation in particular represents the overall shape of the learning curves very well, both for 'rough' (OU) and 'smooth' (RBF) Gaussian priors, and for small as well as for large numbers of training examples n. It is not perfect, but does generally get substantially closer to the true learning curves than existing bounds (two of which, due to Plaskota and to Williams and Vivarelli, we generalized to a wider range of scenarios). For situations with non-uniform input distributions the predictions of the LC approximation tend to be less accurate, and we linked this observation to recent work by Malzahn and Opper (Malzahn and Opper, 2001) on the effects of non-uniformity across input space. Their result, which reduces to the LC approximation for sufficiently uniform scenarios, may in other cases provide better approximations, but this has to be traded off against the higher computational cost that would be involved in actually evaluating the predictions.

We then discussed how the LC approximation could be improved. The effects of discrete n can be incorporated to leading order, but were seen to be relatively minor; on the other hand, the second approximation involved in the derivation (decoupling of averages) appears difficult to improve on within our framework.

Finally, we investigated a simple "degenerate" Gaussian process learning scenario, where the outputs corresponding to different inputs are uncorrelated. This provided us with a means of assessing whether there are limits on the quality of approximations and bounds that take into account only the eigenvalue spectrum of the covariance function. We found indeed that such limits exist: There can be no nontrivial upper bound on the learning curve of this form, and approximations are necessarily of limited quality because different covariance functions with the same eigenvalue spectrum can produce rather different learning curves. We also found that as a bound on the generalization error for *single* data sets (rather than its average over data sets) the Plaskota bound is close to being tight. Whether a tight lower bound on the *average* learning curve exists remains an open question; one plausible candidate worth investigating would be the average generalization error of our degenerate scenario, minimized over all possible input distributions for a fixed eigenvalue spectrum.

There are a number of open problems. One is whether a sub-class of GP learning scenarios can be defined for which the covariance function's eigenvalue spectrum is sufficient to predict the learning curves accurately. Alternatively, one could ask what (minimal) extra information beyond the eigenvalue spectrum needs to be taken into account to arrive at accurate learning curves for all possible GP regression problems. Finally, one may wonder whether the eigenvalue decomposition we have chosen, which explicitly depends on the input distribution, is really the optimal one. On the one hand, recent work (see *e.g.* (Williams and Seeger, 2000)) appears to answer this question in the affirmative. On the other hand, the variability of learning curves among GP covariance functions with the same eigenvalue spectrum suggests that the eigenvalues alone do not provide sufficient information for accurate predictions. One may therefore speculate that eigendecompositions with respect to other input distributions (*e.g.*, maximum entropy ones) might not suffer from this problem. We leave these challenges for future work.

Acknowledgements: We would like to thank Chris Williams, Manfred Opper and Dörte Malzahn for stimulating discussions, and the Royal Society for financial support through a Dorothy Hodgkin Research Fellowship.

A Solving for the LC approximation

In this appendix we describe how to solve eqns. (15,27) for the LC approximation and its first order correction, using the method of characteristic curves. The method applies to partial differential equations of the form $a\,\partial f/\partial x + b\,\partial f/\partial y = c$, where $f = f(x, y)$ and a, b, c can be arbitrary functions of x, y, f. Viewing the solution as a surface in x, y, f-space, one can then show (John, 1978) that if the point (x_0, y_0, f_0) belongs to the solution surface then so does the entire characteristic curve $(x(t), y(t), f(t))$ defined by

$$\frac{dx}{dt} = a, \qquad \frac{dy}{dt} = b, \qquad \frac{df}{dt} = c, \qquad (x(0), y(0), f(0)) = (x_0, y_0, f_0)$$

The solution surface can then be recovered by combining an appropriate one-dimensional family of characteristic curves.

Denote the generalization error predicted by the LC approximation as $\epsilon_0(n, v)$, with v the auxiliary parameter introduced in (13,14). It is the solution of the equation (15)

$$\frac{\partial \epsilon_0}{\partial n} - \frac{1}{\sigma^2 + \epsilon_0} \frac{\partial \epsilon_0}{\partial v} = 0$$

subject to the initial conditions $\epsilon_0(n = 0, v) = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + v\mathbf{I})^{-1}$. These give us a family of solution points which the characteristic curves have to pass through, namely $(n(0) = 0, v(0) = v_0, \epsilon_0(0) = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + v_0\mathbf{I})^{-1})$. The equations for the characteristic curves are

$$\frac{dn}{dt} = 1, \qquad \frac{dv}{dt} = -\frac{1}{\sigma^2 + \epsilon_0}, \qquad \frac{d\epsilon_0}{dt} = 0$$

and can be integrated to give

$$n = n(0) + t = t, \qquad v = v(0) - \frac{t}{\sigma^2 + \epsilon_0} = v_0 - \frac{t}{\sigma^2 + \epsilon_0}, \qquad \epsilon_0 = \epsilon_0(0) = \mathrm{tr}\,(\mathbf{\Lambda}^{-1} + v_0\mathbf{I})^{-1} \quad (35)$$

Eliminating t (the curve parameter) and v_0 (which parameterizes the family of initial points) gives the required solution $\epsilon_0 = \mathrm{tr}\,\{\mathbf{\Lambda}^{-1} + [v + n/(\sigma^2 + \epsilon_0)]\mathbf{I}\}^{-1}$. The LC approximation (16) is obtained by setting $v = 0$.

For the first order correction ϵ_1, we have to solve the equation (27)

$$\frac{\partial \epsilon_1}{\partial n} - \frac{1}{\sigma^2 + \epsilon_0} \frac{\partial \epsilon_1}{\partial v} = -\frac{1}{2}\frac{\partial^2 \epsilon_0}{\partial n^2} - \frac{\epsilon_1}{(\sigma^2 + \epsilon_0)^2} \frac{\partial \epsilon_0}{\partial v}$$

with the initial condition (explained in the main text) $\epsilon_1(n = 0, v) = 0$. Hence a suitable family of initial solution points is $(n(0) = 0, v(0) = v_0, \epsilon_1(0) = 0)$. The characteristic curves must obey

$$\frac{dn}{dt} = 0, \qquad \frac{dv}{dt} = -\frac{1}{\sigma^2 + \epsilon_0}, \qquad \frac{d\epsilon_1}{dt} = -\frac{1}{2}\frac{\partial^2 \epsilon_0}{\partial n^2} - \frac{\epsilon_1}{(\sigma^2 + \epsilon_0)^2} \frac{\partial \epsilon_0}{\partial v}$$

The solutions for the $n(t)$ and $v(t)$-dependence are therefore the same as before, and as a result ϵ_0 is again constant along the characteristic curves. For the derivatives of ϵ_0 that appear, one finds after some algebra

$$\frac{\partial \epsilon_0}{\partial v} = -(\sigma^2 + \epsilon_0)^2 \frac{a_2}{1 - na_2}, \qquad \frac{\partial^2 \epsilon_0}{\partial n^2} = -2(\sigma^2 + \epsilon_0)\frac{a_2^2 - a_3}{(1 - na_2)^3}$$

Here we have used the definitions from (28); because both a_2 and a_3 only depend on n and v in the combination $v + n/(\sigma^2 + \epsilon_0)$, they are also constant along the characteristic curves. Using also that $n = t$ from (35), the equation for ϵ_1 becomes

$$\frac{d\epsilon_1}{dt} = (\sigma^2 + \epsilon_0)\frac{a_2^2 - a_3}{(1 - ta_2)^3} + \epsilon_1 \frac{a_2}{1 - ta_2}$$

This linear differential equation is easily integrated; using the initial condition $\epsilon_1(0) = 0$ one finds

$$\epsilon_1 = (\sigma^2 + \epsilon_0)\frac{t(a_2^2 - a_3)}{(1 - ta_2)^2}$$

Eliminating t again via $t = n$ finally gives the solution (28).

B The Plaskota bound and its extension

To summarize the proof of Plaskota's lower bound on learning curves (Plaskota, 1990) and generalize it to non-uniform covariance functions, we find it most convenient to work with a discrete space of inputs x^α, $\alpha = 1 \ldots N$, with input probabilities $P(x^\alpha) = p_\alpha$. (The case where inputs can vary continuously can be approximated arbitrarily well by such a scenario, either by discretizing the input space on a sufficiently fine grid or by taking the x^α as a large number N of samples from the input distribution and setting $p_\alpha = 1/N$.) A function θ on these inputs can then be thought of as a vector $\boldsymbol{\theta}$ with elements $\theta(x^\alpha)$; similarly, the covariance function $C(x^\alpha, x^\beta) \equiv C_{\alpha\beta}$ becomes an $N \times N$ matrix $\mathbf{C} = \left\langle \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}} \right\rangle$. The eigenvalue equation (6) and the corresponding normalization condition then read

$$\sum_\beta C_{\alpha\beta} p_\beta \phi_i(x^\beta) = \lambda_i \phi_i(x^\alpha), \qquad \sum_\alpha \phi_i(x^\alpha)\phi_j(x^\alpha) p_\alpha = \delta_{ij}$$

We can write these in a more compact form: If we set $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1 \ldots \lambda_N)$ and $\mathbf{P} = \mathrm{diag}(p_1 \ldots p_N)$, denote $\boldsymbol{\phi}_i$ the vector with entries $\phi_i(x^\alpha)$ and $\mathbf{\Phi}$ the $N \times N$ matrix whose columns are the $\boldsymbol{\phi}_i$, we have

$$\mathbf{CP\Phi} = \mathbf{\Phi\Lambda}, \qquad \mathbf{\Phi}^{\mathrm{T}}\mathbf{P\Phi} = \mathbf{I}$$

and hence

$$\mathbf{C} = \mathbf{\Phi\Lambda\Phi}^{\mathrm{T}}$$

One can also write this in the form of a more conventional matrix diagonalization:

$$\mathbf{P}^{1/2}\mathbf{C}\mathbf{P}^{1/2} = (\mathbf{P}^{1/2}\mathbf{\Phi})\mathbf{\Lambda}(\mathbf{P}^{1/2}\mathbf{\Phi})^{\mathrm{T}}, \qquad (\mathbf{P}^{1/2}\mathbf{\Phi})^{\mathrm{T}}(\mathbf{P}^{1/2}\mathbf{\Phi}) = \mathbf{I}$$

It follows that the λ_i are the eigenvalues of the matrix $\mathbf{P}^{1/2}\mathbf{C}\mathbf{P}^{1/2}$ and that

$$\langle C(x,x)\rangle_x \equiv \mathrm{tr}\,\mathbf{PC} = \mathrm{tr}\,\mathbf{P}^{1/2}\mathbf{C}\mathbf{P}^{1/2} = \mathrm{tr}\,\mathbf{\Lambda} \tag{36}$$

in agreement with (7).

Plaskota now considers the case of *generalized* training sets consisting of n generalized observations $y_l = \mathbf{L}_l^{\mathrm{T}}\boldsymbol{\theta} + \eta_l$. Each y_l is a linear combinations of the values of the function $\theta(x)$, with coefficients specified by the vector $\mathbf{L}_l$, corrupted by additive Gaussian noise η_l which as before we take to be of variance σ^2. The conventional scenario, where each training point contains a (corrupted) observation of $\theta(x)$ at a single point $x = x^{\alpha(l)}$, corresponds to $\mathbf{L}_l = \mathbf{e}_{\alpha(l)}$. Here we denote $\mathbf{e}_\alpha$ the α-th unit vector (with the α-th component equal to one and all others zero) and write the l-th training input as $x^{\alpha(l)}$, with $\alpha(l) \in \{1 \ldots N\}$.

After the observation of the y_l, we will have some posterior covariances $\langle \Delta\theta(x^\alpha)\Delta\theta(x^\beta)\rangle$. Collecting these into an $N \times N$ matrix $\mathbf{V}$ one can write them explicitly as

$$\mathbf{V} = \mathbf{C} - \mathbf{CL}(\mathbf{L}^{\mathrm{T}}\mathbf{CL} + \sigma^2\mathbf{I})^{-1}\mathbf{L}^{\mathrm{T}}\mathbf{C} = (\mathbf{C}^{-1} + \sigma^{-2}\mathbf{LL}^{\mathrm{T}})^{-1} \tag{37}$$

where $\mathbf{L}$ is an $N \times n$ matrix whose columns are the $\mathbf{L}_l$. The first version of this result can be seen as a generalization of (2); the second version is the corresponding generalization[9] of (8) in a different guise. It then follows that the generalization error is given by

$$\epsilon(D) = \sum_\alpha p_\alpha \left\langle (\Delta\theta(x^\alpha))^2 \right\rangle = \mathrm{tr}\,\mathbf{PV} = \mathrm{tr}\,\mathbf{PC} - \mathrm{tr}\,[\mathbf{L}^{\mathrm{T}}\mathbf{CPCL}(\mathbf{M} + \sigma^2\mathbf{I})^{-1}] \tag{38}$$

where we have defined the $n \times n$ matrix $\mathbf{M} = \mathbf{L}^{\mathrm{T}}\mathbf{CL}$. Its trace (which will become important shortly) is $\mathrm{tr}\,\mathbf{M} = \sum_l \mathbf{L}_l\mathbf{CL}_l$; for a conventional training set with training inputs $x^{\alpha(l)}$ this becomes $\mathrm{tr}\,\mathbf{M} = \sum_l \mathbf{e}_{\alpha(l)}^{\mathrm{T}}\mathbf{Ce}_{\alpha(l)} = \sum_l C(x^{\alpha(l)}, x^{\alpha(l)})$.

So far everything is exact. The idea behind Plaskota's bound is now as follows: For any given *conventional* training set with n training inputs, minimize $\epsilon(D)$ over all *generalized* training sets of size n which have the same value of $\mathrm{tr}\,\mathbf{M}$. The result then clearly gives a lower bound on $\epsilon(D)$ for the given training set.

To carry out this minimization, one can proceed as follows. The matrix $\mathbf{M}$ is symmetric and can be diagonalized, so that $\mathbf{M} = \mathbf{O}\boldsymbol{\eta}\mathbf{O}^{\mathrm{T}}$ where $\mathbf{O}$ is an orthogonal $n \times n$ matrix and $\boldsymbol{\eta}$ is a diagonal matrix whose entries η_i are the eigenvalues of $\mathbf{M}$. Since $\mathbf{M}$ is positive semi-definite, the η_i are

[9] Note that the matrices $\mathbf{V}$ and $\mathcal{G}$ are related in the same way as $\mathbf{C}$ and $\mathbf{\Lambda}$; explicitly, one has $\mathbf{V} = \mathbf{\Phi}\mathcal{G}\mathbf{\Phi}^{\mathrm{T}}$.

non-negative[10]; we also assume them to be ordered in descending order, $\eta_1 \geq \ldots \geq \eta_N$. It then follows that $(\mathbf{M} + \sigma^2\mathbf{I})^{-1} = \mathbf{O}(\boldsymbol{\eta} + \sigma^2\mathbf{I})^{-1}\mathbf{O}^{\mathrm{T}}$, so that

$$\epsilon(D) = \operatorname{tr}\boldsymbol{\Lambda} - \operatorname{tr}[\mathbf{O}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{CPCLO}(\boldsymbol{\eta} + \sigma^2\mathbf{I})^{-1}]$$

where we have also used that $\operatorname{tr}\mathbf{PC} = \operatorname{tr}\boldsymbol{\Lambda}$ from (36). Now, if we define the $N \times n$ matrix $\mathbf{Q} = \mathbf{C}^{1/2}\mathbf{LO}\boldsymbol{\eta}^{-1/2}$ and write $\tilde{\mathbf{C}} = \mathbf{C}^{1/2}\mathbf{PC}^{1/2}$, then $\boldsymbol{\eta}^{1/2}\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q}\boldsymbol{\eta}^{1/2} = \mathbf{O}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{CPCLO}$ and so

$$\epsilon(D) = \operatorname{tr}\boldsymbol{\Lambda} - \operatorname{tr}[\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q}\boldsymbol{\eta}(\boldsymbol{\eta} + \sigma^2\mathbf{I})^{-1}] = \operatorname{tr}\boldsymbol{\Lambda} - \sum_{i=1}^{n}(\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q})_{ii}\frac{\eta_i}{\eta_i + \sigma^2} \tag{39}$$

To minimize $\epsilon(D)$, we need to make the sum over i maximal. If we write the n columns of $\mathbf{Q}$ as vectors $\mathbf{Q}_i$ then $(\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q})_{ii} = \mathbf{Q}_i^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q}_i$; also, from the definition of $\mathbf{Q}$ we have $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$, implying that the $\mathbf{Q}_i$ are orthonormal. Now it is easy to see that $\tilde{\mathbf{C}} = \mathbf{C}^{1/2}\mathbf{PC}^{1/2}$ has the same eigenvalues as $\mathbf{P}^{1/2}\mathbf{CP}^{1/2}$, namely, the λ_i. Assuming these to be ordered in a descending sequence, the largest value that $\mathbf{Q}_i^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q}_i$ can take is therefore λ_1. Because of the orthonormality of the $\mathbf{Q}_i$, one has similarly that $\sum_{i\leq k}\mathbf{Q}_i^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q}_i \leq \sum_{i\leq k}\lambda_i$ for all $k \leq n$. Since the terms $\eta_i/(\eta_i + \sigma^2)$ form a descending sequence (due to our assumed ordering of the η_i), it is then clear that the optimal situation is the one where $\mathbf{Q}_1^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q}_1$, multiplying the largest value $\eta_1/(\eta_1 + \sigma^2)$, has the largest possible value λ_1, $\mathbf{Q}_2^{\mathrm{T}}\tilde{\mathbf{C}}\mathbf{Q}_2$ has the largest possible value subject to this, which is λ_2, and so on. Plaskota indeed proved formally that

$$\epsilon(D) \geq \operatorname{tr}\boldsymbol{\Lambda} - \sum_{i=1}^{n}\frac{\lambda_i\eta_i}{\eta_i + \sigma^2}$$

Finally, one can now optimize over the η_i, subject to the constraint that we imposed initially, *i.e.*, that $\operatorname{tr}\mathbf{M} = \sum_i \eta_i$ equals the sum $S = \sum_l C(x^{\alpha(l)}, x^{\alpha(l)})$. This gives the final Plaskota bound

$$\epsilon(D) \geq \operatorname{tr}\boldsymbol{\Lambda} - \max_{\{\eta_i\}}\sum_{i=1}^{n}\frac{\lambda_i\eta_i}{\eta_i + \sigma^2} = \min_{\{\eta_i\}}\left(\sum_{i=1}^{n}\frac{\lambda_i\sigma^2}{\eta_i + \sigma^2}\right) + \sum_{i\geq n+1}\lambda_i \tag{40}$$

as given in (19). Plaskota also proved this bound to be realizable by an appropriate choice of the generalized observations $\mathbf{L}_l$. His original derivation only applied to uniform covariance functions, but the above proof sketch shows that this restriction is not in fact necessary.

To evaluate the Plaskota bound in practice it is desirable to have a more explicit expression for the minimum over the η_i. Introducing a Lagrange multiplier α for the constraint $\sum_i \eta_i = S$, one finds by differentiation of (40) that each positive η_i must satisfy

$$-\frac{\lambda_i\sigma^2}{(\eta_i + \sigma^2)^2} + \alpha = 0 \tag{41}$$

while for the vanishing η_i the quantity on the left hand side has to be positive, *i.e.*, $\lambda_i \leq \alpha\sigma^2$. Due to the ordering of the λ_i, one sees from this that at the minimum the first k of the η_i will be nonzero and the rest will be zero, with $k \leq n$ a number to be determined. Assume we know k. Then from (41) the nonzero η_i are given by

$$\eta_i = \left(\frac{\lambda_i\sigma^2}{\alpha}\right)^{1/2} - \sigma^2$$

Using that $\sum_{i\leq k}\eta_i = S$, one then finds $\alpha^{-1/2} = (S + k\sigma^2)/(\sigma\sum_{j\leq k}\lambda_j^{1/2})$ and hence

$$\eta_i = \lambda_i^{1/2}\frac{S + k\sigma^2}{\sum_{j\leq k}\lambda_j^{1/2}} - \sigma^2 \tag{42}$$

[10] We actually assume, for simplicity of presentation, that all the η_i are nonzero. Equation (39) still holds if some η_i vanish, but the derivation is then slightly more complicated: One has to replace $\boldsymbol{\eta}$ by a smaller diagonal matrix which contains only the positive eigenvalues and $\mathbf{O}$ by the matrix whose columns are the corresponding eigenvectors of $\mathbf{M}$.

In order for the value of k that we assumed to be the correct one, this expression needs to be positive for $i = 1 \ldots k$, while for $i = k+1 \ldots n$ it must be zero or negative (as follows from $\lambda_i \leq \alpha\sigma^2$). Due to the ordering of the λ_i, it is sufficient to check whether (42) gives $\eta_k > 0$ and $\eta_{k+1} \leq 0$. The k that satisfies these conditions[11] gives the minimum in (40); using (42), the bound can then be simplified to

$$\epsilon(D) \geq \frac{\sigma^2}{S + k\sigma^2} \left(\sum_{i \leq k} \lambda_i^{1/2} \right)^2 + \sum_{i \geq k+1} \lambda_i$$

In the example scenarios for which we evaluated the bound, we found that $k = n$ in the vast majority of cases. A sensible numerical procedure is therefore to check first whether for $k = n$ equation (42) gives $\eta_n > 0$. If yes, then $k = n$ gives the required optimum; if no, then k should be decreased until the conditions $\eta_k > 0$ and $\eta_{k+1} \leq 0$ are met.

C Eigenvalue spectra for the example scenarios

We explain first how the covariance functions with periodic boundary conditions for $x \in [0,1]^d$ are constructed. Consider first the case $d = 1$. The periodic RBF covariance function is defined as

$$C(x, x') = \sum_{r=-\infty}^{\infty} c(x - x' - r) \tag{43}$$

where $c(x - x') = \exp[-|x - x'|^2/(2l^2)]$ is the original covariance function; for the periodic OU case we use instead $c(x - x') = \exp(-|x - x'|/l)$. One sees that for sufficiently small l ($l \ll 1$), only the $r = 0$ term makes a significant contribution, except when x and x' are within $\approx l$ of opposite ends of the input space (so that either $x - x' + 1$ or $x - x' - 1$ are of order l). We therefore expect the periodic covariance functions and the conventional non-periodic ones to yield very similar learning curves, as long as the length scale of the covariance function is smaller than the size of the input domain.

The advantage of having a periodic covariance function is that its eigenfunctions are simple Fourier waves and the eigenvalues can be calculated by Fourier transformation. This can be seen as follows. For the assumed uniform input distribution on $[0,1]$, the defining equation for an eigenfunction $\phi(x)$ with eigenvalue λ is, from (6)

$$\langle C(x, x')\phi(x') \rangle_{x'} = \int_0^1 dx' \; C(x, x')\phi(x') = \lambda\phi(x).$$

Inserting (43) and assuming that $\phi(x)$ is continued periodically outside the interval $[0,1]$, this becomes

$$\sum_{r=-\infty}^{\infty} \int_0^1 dx' \; c(x - x' - r)\phi(x') = \sum_{r=-\infty}^{\infty} \int_r^{r+1} dx' \; c(x - x')\phi(x' - r) = \int_{-\infty}^{\infty} dx' \; c(x - x')\phi(x') = \lambda\phi(x)$$

It is well known that the solutions of this eigenfunction equation are Fourier waves $\phi_q(x) = e^{2\pi i q x}$ for integer (positive or negative) q, with corresponding eigenvalues

$$\lambda_q = \int_{-\infty}^{\infty} dx \; c(x) e^{-2\pi i q x}$$

The eigenvalues λ_q are real since $c(x) = c(-x)$ (this follows from the requirement that the covariance function $C(x, x')$ must be symmetric). The eigenfunctions for $q \neq 0$ are complex in the form given but can be made explicitly real by linearly transforming the pair $\phi_q(x)$ and $\phi_{-q}(x)$ into $(1/\sqrt{2})\cos(2\pi q x)$ and $(1/\sqrt{2})\sin(2\pi q x)$.

[11]There is a unique k with this property; this follows because the minimum in (40) is over a convex function of the η_i and therefore unique.

All of the above generalizes immediately to higher input dimension d. One defines

$$C(x, x') = \sum_r c(x - x' - r)$$

where r now runs over all d-dimensional vectors with integer components; the argument $x - x' - r$ of $c(\cdot)$ is now a d-dimensional real vector. The eigenvalues of this periodic covariance function are then given by

$$\lambda_q = \int dx\ c(x) e^{-2\pi i q \cdot x}$$

They are indexed by d-dimensional integer vectors q; the integration is over all real-valued d-dimensional vectors x, and $q \cdot x$ is the conventional dot product between vectors. Explicitly, one derives that for the periodic RBF covariance function

$$\lambda_q = (2\pi)^{d/2} l^d e^{-(2\pi l)^2 ||q||^2/2}$$

For the periodic OU covariance function, on the other hand, one has

$$\lambda_q = \kappa_d l^d [1 + (2\pi l)^2 ||q||^2]^{-(d+1)/2} \tag{44}$$

with $\kappa_d = 2,\ 2\pi,\ 8\pi$ for $d = 1, 2, 3$, respectively; for general d, $\kappa_d = \pi^{(d-1)/2} 2^d \Gamma((d+1)/2)$ where $\Gamma(z) = (z-1)!$ is the Gamma function.

All the bounds and approximations in principle require traces over the whole eigenvalue spectrum, corresponding to sums over an infinite number of terms. Numerically, we perform the sums over all eigenvalues up to some suitably large maximal value $q_{\max}$ of $||q||$. The remaining small eigenvalue tail of the spectrum is then treated by approximating $||q||$ as a continuous variable $\tilde{q}$ and integrating over it from $q_{\max}$ to infinity, with the appropriate weighting for the number of vectors q in a shell $\tilde{q} \le ||q|| \le \tilde{q} + d\tilde{q}$. To check that this procedure gave accurate results, we always verified that the numerically calculated $\operatorname{tr} \mathbf{\Lambda}$ agreed with the known value of $C(x, x)$.

The third scenario we consider is that of a conventional RBF kernel $C(x, x') = \exp[-||x - x'||^2/(2l^2)]$ with a nonuniform input distribution which we assume to be an isotropic zero mean Gaussian, $P(x) \propto \exp[-||x||^2/(2\sigma_x^2)]$. The eigenfunctions and eigenvalues are worked out in (Zhu et al., 1998) for the case $d = 1$; the eigenvalues are labelled by a non-negative integer q and given by

$$\lambda_q = (1 - \beta)\beta^q$$

where

$$\beta^{-1} = 1 + r/2 + \sqrt{r^2/4 + r}, \qquad r = l^2/\sigma_x^2.$$

As expected, only the ratio r of the length scales l and σ_x enters since the overall scale of the inputs is immaterial. (To avoid this trivial invariance we fixed $\sigma_x^2 = 1/12$; this specific value gives the same variance for each component of x as for the uniform distribution over $[0, 1]^d$ used in the other scenarios.) For $d > 1$, this result generalizes immediately because both the covariance function and the input distribution factorize over the different input components (Opper and Vivarelli, 1999; Williams and Seeger, 2001). The eigenvalues are therefore just products of the eigenvalues for each component, and indexed by a d-dimensional vector q of non-negative integers:

$$\lambda_q = (1 - \beta)^d \beta^s, \qquad s = \sum_{i=1}^{d} q_i \tag{45}$$

One sees that the eigenvalues will come in blocks: All vectors q with the same $s = \sum_i q_i$ give the same λ_q. Numerically, we therefore only have to store the different eigenvalues and their multiplicities, which can be shown to be[12] $(d+s-1)!/[s!(d-1)!]$. With this trick so many eigenvalues

[12] There is a nice combinatorial way of seeing this. Imagine a row of $d + s + 1$ holes. Each hole is either empty or contains one ball, except the holes at the two ends which have one ball placed in them permanently. Now consider $d-1$ identical balls distributed across the $d+s-1$ free holes, and let q_i ($i = 1 \ldots d$) be the number of free holes between balls $i-1$ and i (where balls 0 and d are the fixed balls in the holes at the left and right end, respectively). The q_i are non-negative integers, and their sum equals the number of free holes, giving $\sum_i q_i = (d + s - 1) - (d - 1) = s$. The number of different assignments of the q_i is thus identical to the number of different arrangements of $d - 1$ identical balls across $d + s - 1$ holes, hence the result.

can be treated by direct summation that a separate treatment of the neglected eigenvalues (via an integral, as above) is unnecessary.

References

D Barber and C K I Williams. Gaussian processes for Bayesian classification via hybrid Monte Carlo. In M C Mozer, M I Jordan, and T Petsche, editors, *Advances in Neural Information Processing Systems NIPS 9*, pages 340–346, Cambridge, MA, 1997. MIT Press.

P W Goldberg, C K I Williams, and C M Bishop. Regression with input-dependent noise: A Gaussian process treatment. In M I Jordan, M J Kearns, and S A Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 493–499, Cambridge, MA, 1998. MIT Press.

F John. *Partial Differential Equations.* Springer, New York, 3rd edition, 1978.

D J C MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

D Malzahn and M Opper. Learning curves for Gaussian processes regression: A framework for good approximations. In T K Leen, T G Dietterich, and V Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 273–279. MIT Press, 2001.

C A Michelli and G Wahba. Design problems for optimal surface interpolation. In Z Ziegler, editor, *Approximation theory and applications*, pages 329–348. Academic Press, 1981.

R M Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

M Opper. Regression with Gaussian processes: Average case performance. In I K Kwok-Yee, M Wong, I King, and Dit-Yun Yeung, editors, *Theoretical Aspects of Neural Computation: A Multidisciplinary Perspective*, pages 17–23. Springer, 1997.

M Opper and F Vivarelli. General bounds on bayes errors for regression with Gaussian processes. In M Kearns, S A Solla, and D Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 302–308, Cambridge, MA, 1999. MIT Press.

L Plaskota. On the average case complexity of linear problems with noisy information. *Journal of Complexity*, 6:199–230, 1990.

W H Press, S A Teukolsky, W T Vetterling, and B P Flannery. *Numerical Recipes in C (2nd ed.).* Cambridge University Press, Cambridge, 1992.

K Ritter. Almost optimal differentiation using noisy data. *Journal of Approximation Theory*, 86 (3):293–309, 1996.

B W Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society Series B*, 47(1):1–52, 1985.

P Sollich. Finite-size effects in learning and generalization in linear perceptrons. *Journal of Physics A*, 27:7771–7784, 1994.

P Sollich. Learning curves for Gaussian processes. In M S Kearns, S A Solla, and D A Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 344–350, Cambridge, MA, 1999b. MIT Press.

P Sollich. Approximate learning curves for Gaussian processes. In *ICANN99 – Ninth International Conference on Artificial Neural Networks*, pages 437–442, London, 1999a. The Institution of Electrical Engineers.

M L Stein. Comment on the paper by Sacks, j et al: Design and analysis of computer experiments. *Statistical Science*, 4:432–433, 1989.

G F Trecate, C K I Williams, and M Opper. Finite-dimensional approximation of Gaussian processes. In M Kearns, S A Solla, and D Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 218–224, Cambridge, MA, 1999. MIT Press.

C K I Williams. Computing with infinite networks. In M C Mozer, M I Jordan, and T Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 295–301, Cambridge, MA, 1997. MIT Press.

C K I Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M I Jordan, editor, *Learning and Inference in Graphical Models*, pages 599–621. Kluwer Academic, 1998.

C K I Williams and C E Rasmussen. Gaussian processes for regression. In D S Touretzky, M C Mozer, and M E Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 514–520, Cambridge, MA, 1996. MIT Press.

C K I Williams and M Seeger. The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann, 2000.

C K I Williams and M Seeger. Using the Nyström method to speed up kernel machines. In T K Leen, T G Dietterich, and V Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

C K I Williams and F Vivarelli. Upper and lower bounds on the learning curve for Gaussian processes. *Mach. Learn.*, 40(1):77–102, 2000.

E Wong. *Stochastic Processes in Information and Dynamical Systems*. McGraw-Hill, New York, 1971.

H Zhu, C K I Williams, R J Rohwer, and M Morciniec. Gaussian regression and optimal finite dimensional linear models. In C M Bishop, editor, *Neural Networks and Machine Learning*. Springer, 1998.