Active Learning of Linear Embeddings for Gaussian Processes

Roman Garnett
University of Bonn
Römerstraße 164
53117 Bonn, Germany
rgarnett@uni-bonn.de

Michael A. Osborne
University of Oxford
Parks Road
Oxford OX1 3PJ, UK
mosb@robots.ox.ac.uk

Philipp Hennig
MPI for Intelligent Systems
Spemannstraße
72076 Tübingen, Germany
phennig@tue.mpg.de

Abstract

We propose an active learning method for discovering low-dimensional structure in high-dimensional Gaussian process (GP) tasks. Such problems are increasingly frequent and important, but have hitherto presented severe practical difficulties. We further introduce a novel technique for approximately marginalizing GP hyperparameters, yielding marginal predictions robust to hyperparameter misspecification. Our method offers an efficient means of performing GP regression, quadrature, or Bayesian optimization in high-dimensional spaces.

1 INTRODUCTION

We propose a method to actively learn, simultaneously, about a function and a low-dimensional embedding of its input domain. High dimensionality has stymied the progress of model-based approaches to many common machine learning tasks. In particular, although Bayesian nonparametric modeling with Gaussian processes (GPs) (Rasmussen & Williams, 2006) has become popular for regression, classification, quadrature (O'Hagan, 1991), and global optimization (Brochu et al., 2010), such approaches remain intractable for large numbers of input variables (with the exception of local optimization (Hennig & Kiefel, 2012)). An old idea for the solution to this problem is the exploitation of low-dimensional structure; the most tractable such case is that of a linear embedding. Throughout this text, we consider a function $f(x)\colon \mathbb{R}^D \to \mathbb{R}$ of a high-dimensional variable $x \in \mathbb{R}^D$ (for notational simplicity, x will be assumed to be a row vector). The assumption is that f, in reality, only depends on the variable $u \coloneqq xR^\top$, of much lower dimensionality $d \ll D$, through a linear embedding $R \in \mathbb{R}^{d \times D}$. We are interested in an algorithm that simultaneously learns R and f, and does so in an active way. That is, it iteratively selects informative locations x_* in a box-bounded region $\mathcal{X} \subset \mathbb{R}^D$, and collects associated obser-

vations y_* of $f_* \coloneqq f(x_*)$ corrupted by i.i.d. Gaussian noise: $p(y_* \mid f_*) = \mathcal{N}(y_*; f_*, \sigma^2)$.

The proposed method comprises three distinct steps (Algorithm 1): constructing a probability distribution over possible embeddings (*learning the embedding R*); using this belief to determine a probability distribution over the function itself (*learning the function f*), and then choosing evaluation points to best inform these beliefs (*active selection*). To learn the embedding, we use a Laplace approximation on the posterior over R to quantify the uncertainty in the embedding (Section 2). To learn the function, we develop a novel approximate means of marginalizing over Gaussian process hyperparameters (including those parameterizing embeddings), to provide predictions robust to hyperparameter misspecification (Section 3). This sub-algorithm is more generally applicable to many Gaussian process tasks, and to the marginalization of hyperparameters other than embeddings, and so represents a core contribution of this paper. Finally, for active selection, we extend previous work (Houlsby et al., 2011) to select evaluations that maximize the expected reduction in uncertainty about R (Section 4).

Estimators for R in wide use include LASSO (Tibshirani, 1996) and the Dantzig selector (Candes & Tao, 2007), both of which assume $d = 1$. These are passive methods estimating the linear embedding from a fixed dataset. This paper develops an algorithm that *actively* learns R for the domain of a Gaussian process. The goal is to use few function evaluations to intelligently explore and identify R. Notice that although the embedding is assumed to be linear, the function f itself will be allowed to be nonlinear via the GP prior.

This problem is related to, but distinct from, dimensionality reduction (Lawrence, 2012), for which active learning has recently been proposed (Iwata et al., 2013). Dimensionality reduction is also known as visualization or blind source separation, and is solved using, e.g., principal component analysis (PCA), factor analysis, or latent variable models. As in dimensionality reduction, we consider the problem of finding a low-dimensional representation of an input or feature matrix $X \in \mathbb{R}^{N \times D}$; unlike dimensionality reduction, we do so given an associated vector of training

Algorithm 1 Simultaneous active learning of functions and their linear embeddings (pseudocode)

Require: d, D; kernel κ, mean function μ; prior $p(R)$
  $X \leftarrow \varnothing; y \leftarrow \varnothing$
  repeat
    $q(R) \leftarrow \textsc{LaplaceApprox}\big(p(R \mid X, y, \kappa, \mu)\big)$
        // approximate posterior on embedding R
    $q(f) \leftarrow \textsc{ApproxMarginal}\big(p(f \mid R), q(R)\big)$
        // approximate marginal on function f
    $x_* \quad \leftarrow \textsc{OptimizeUtility}\big(q(f), q(R)\big)$
        // find approximate optimal evaluation point x_*
    $y_* \quad \leftarrow \textsc{Observe}\big(f(x_*)\big)$ // act
    $X \leftarrow [X; x_*]; y \leftarrow [y; y_*]$ // store data
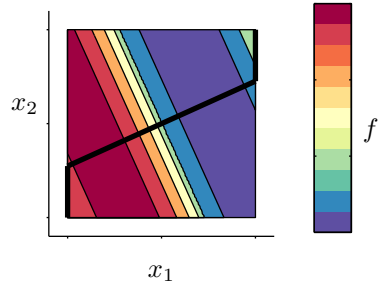 **until** budget depleted
  return $q(R), q(f)$.

Figure 1: A function f with a one-dimensional linear embedding and its box-bounded domain $\mathcal{X}$ Searching over the thick black lines captures all variation of the function. Our aim is to learn this embedding, represented by embedding matrix $R \in \mathbb{R}^{1\times 2}$, by selecting evaluations of f in some search space.

outputs or labels $y \in \mathbb{R}^N$, containing information about which inputs are most relevant to a function. The problem of discovering linear embeddings of GPs was discussed by Snelson & Ghahramani (2006) for the passive case. Active supervised learning has been widely investigated (MacKay, 1992b; Guestrin et al., 2005; Houlsby et al., 2011); our work hierarchically extends this idea to additionally identify the embedding. A special case of our method (the case of a diagonal R) is the hitherto unconsidered problem of *active automatic relevance determination* (MacKay, 1992a; Neal, 1995; Williams & Rasmussen, 1996).

Identifying embeddings is relevant for numerous Gaussian process applications, notably regression, classification, and optimization. Within Bayesian optimization, much recent work has focused on high-dimensional problems (Hutter et al., 2011; Chen et al., 2012; Carpentier & Munos, 2012; Bergstra & Bengio, 2012; Hutter, 2009). Recently, Wang et al. (2013) proposed using randomly generated linear embeddings. In contrast, our active learning strategy can provide an initialization phase that selects objective function evaluations so as to best learn low-dimensional structure. This permits the subsequent optimization of high-dimensional objectives over only the learned low-dimensional embedding.

Alongside this paper, we are releasing open-source software implementing our contributions. A simple MATLAB implementation of Algorithm 1, built on the Gaussian Process for Machine Learning (GPML) Toolbox,[1] is freely available.[2] An independent, GPML-compatible implementation of the MGP (Section 3) is also available.[3]

---

[1] http://www.gaussianprocess.org/gpml/code
[2] https://github.com/rmgarnett/active_gp_hyperlearning
[3] https://github.com/rmgarnett/mgp

2 LINEAR EMBEDDINGS OF GAUSSIAN PROCESSES

In many applications, like image analysis, D can be in the order of thousands or millions. But even $D = 10$ is a high dimensionality for common Gaussian process models, not only from a computational, but also from an informational, perspective. Because standard GP covariance functions stipulate that function values separated by more than a few input length scales are negligibly correlated, for high D, almost all of $\mathcal{X}$ is uncorrelated with observed data. Hence data is effectively ignored during most predictions, and learning is impossible. Practical experience shows, however, that many functions are insensitive to some of their inputs (Wang et al., 2013), thus have low *effective* dimensionality (Figure 1). Our goal is to discover an $R \in \mathbb{R}^{d\times D}$ such that, for low-dimensional $\mathcal{U} \subset \mathbb{R}^d$, $u = xR^\top$, $\forall u \in \mathcal{U}$, $x \in \mathcal{X}$ and $f(x) = \tilde{f}(u)$ for a new function, $\tilde{f}:\mathcal{U} \to \mathbb{R}$. The discussion here will be restricted to predefined d; in reality, this is likely to be defined as the maximum number of dimensions that can be feasibly considered in light of computational constraints. If the actual d is lower than this limit, R can be padded with rows of zeros.

We adopt a GP prior on $\tilde{f}$ with mean and covariance functions $\tilde{\mu}$ and $\tilde{\kappa}$, respectively. The linear embedding induces another GP prior $p(f) = \mathcal{GP}(f; \mu, \kappa)$, where $\mu(x) = \tilde{\mu}(x R^\top)$ and $\kappa(x, x') = \tilde{\kappa}(x R^\top, x' R^\top)$. For example, if $\tilde{\kappa}$ is the well-known isotropic exponentiated-quadratic (squared-exponential, radial basis function, Gaussian) covariance, $\tilde{\kappa}(u, u') := \gamma^2 \exp\left[-\frac{1}{2}(u - u')(u - u')^\top\right]$ with output scale γ, κ on f is the Mahalanobis exponentiated-quadratic covariance

$$\kappa(x, x') = \gamma^2 \exp\left[-\frac{1}{2}(x - x')R^\top R(x - x')^\top\right]. \quad (1)$$

If $d = D = 1$, then $R \in \mathbb{R}$ is an inverse length scale. We will return to this one-dimensional example later to build intuition. A further special case is a diagonal R (assuming

$d = D$), in which case κ is the automatic relevance determination (ARD) covariance (Neal, 1995), widely used to identify the most-important inputs.

Given an appropriate R with acceptably small d, learning about f is possible even for large D, because the regression problem is reduced to the manageable space $\mathbb{R}^d$. This can remain true even in the case of an uncertain R: in particular, assume a prior $p(R) = \mathcal{N}(R; \hat{R}, \Sigma)$. Thus, recalling that $u = xR^\top$, and using standard Gaussian identities, if $d = 1$, $p(u \mid x) = \mathcal{N}(u; x\hat{R}^\top, x\Sigma x^\top)$. If $d > 1$, Σ is $\mathrm{Cov}[\mathrm{vect}R]$, resulting in another Gaussian for $p(u \mid x)$ that is only slightly more involved than in the $d = 1$ case. As such, regression on f reduces to GP regression on $\tilde{f}$, whose domain is the much smaller $\mathcal{U} \subset \mathbb{R}^d$, but with uncertain, Gaussian-distributed, inputs. Unlike the work of McHutchon & Rasmussen (2011), giving an existing approach to GP regression with uncertain inputs, the Gaussian over the inputs here is correlated; the location of a point is correlated with all others via mutual dependence on R. And unlike the setting considered by Girard & Murray-Smith (2005), there is no natural ordering of this domain enabling an iterative procedure. The following section describes a novel means of regression with uncertain embedding R.

2.1 APPROXIMATING THE POSTERIOR ON R

The log-likelihood of R, after N observations forming a dataset $\mathcal{D} := (X, y) \in \mathbb{R}^{N \times D} \times \mathbb{R}^N$, is

$$\log p(y \mid X, R) = \log \mathcal{N}(y; \mu_X, K_{XX} + \sigma^2 \mathbb{I}) \qquad (2)$$
$$= -1/2 \big[(y - \mu_X)^\top (K_{XX} + \sigma^2 \mathbb{I})^{-1} (y - \mu_X)$$
$$+ \log|K_{XX} + \sigma^2 \mathbb{I}| + N \log 2\pi \big].$$

As $\mu_X := \mu(X)$ and $K_{XX} := \kappa(X, X)$ have nonlinear dependence upon R, so does $p(y \mid X, R)$. Even a simplistic prior on the elements of R thus gives a complicated posterior. We will use a Laplace approximation for $p(R \mid \mathcal{D})$ to attain a tractable algorithm. To construct a Gaussian approximation, $\mathcal{N}(R; \hat{R}, \Sigma) \simeq p(R \mid \mathcal{D})$, we find a mode of the posterior of $p(R \mid \mathcal{D})$ and set this mode as the mean $\hat{R}$ of our approximate distribution. The covariance of the Gaussian approximation is taken as the inverse Hessian of the negative logarithm of the posterior evaluated at $\hat{R}$,

$$\Sigma^{-1} = -\nabla\nabla^\top \log p(R \mid \mathcal{D})\big|_{R=\hat{R}}. \qquad (3)$$

2.1.1 Computational Cost

How costly is it to construct the Laplace approximation of Equation (3)? Since D may be a large number, active learning should have low cost in D. This section shows that the required computations can be performed in time linear in D, using standard approximate numerical methods. It is a technical aspect that readers not interested in details may want to skip over.

Up to normalization, the log posterior is the sum of log prior and log likelihood (2). The former can be chosen very simplistically; the latter has gradient and Hessian given by, defining $G := \kappa_{XX} + \sigma^2 \mathbb{I}$ and $\Gamma := G^{-1}(y - \mu_X)$,

$$-2 \frac{\partial \log p(y \mid X, R)}{\partial \theta} = -\Gamma^\top \frac{\partial \kappa_{XX}}{\partial \theta} \Gamma + \mathrm{Tr}\left[G^{-1} \frac{\partial \kappa_{XX}}{\partial \theta} \right];$$
$$-2 \frac{\partial^2 \log p(y \mid X, R)}{\partial \theta \partial \eta} = 2\Gamma^\top \frac{\partial \kappa_{XX}}{\partial \eta} G^{-1} \frac{\partial \kappa_{XX}}{\partial \theta} \Gamma \qquad (4)$$
$$- \mathrm{Tr}\left[G^{-1} \frac{\partial \kappa_{XX}}{\partial \eta} G^{-1} \frac{\partial \kappa_{XX}}{\partial \theta} \right]$$
$$- \Gamma^\top \frac{\partial^2 \kappa_{XX}}{\partial \theta \partial \eta} \Gamma + \mathrm{Tr}\left[G^{-1} \frac{\partial^2 \kappa_{XX}}{\partial \theta \partial \eta} \right].$$

Together with the analogous expressions for a prior $p(R)$, these expressions can be used to find a maximum of the posterior distribution (e.g., via a quasi-Newton method), and the Hessian matrix required for the Laplace approximation to $p(R \mid \mathcal{D})$. The computational cost of evaluating these expressions depends on the precise algebraic form of the kernel κ. For the exponentiated quadratic kernel of Equation (1), careful analysis shows that the storage cost for the Hessian of (2) is $\mathcal{O}(N^2 dD)$, and its structure allows its multiplication with a vector in $\mathcal{O}(N^2 dD)$. The corresponding derivations are tedious and not particularly enlightening. To give an intuition, consider the most-involved term in (4): Using the short-hand $\Delta_\ell^{ij} := x_{i\ell} - x_{j\ell}$, a straightforward derivation gives the form

$$H^1_{k\ell,ab} := -\sum_{ij} \Gamma_i \frac{\partial^2 \kappa(x_i, x_j)}{\partial R_{k\ell} \partial R_{ab}} \Gamma_j$$
$$= \sum_{ijop} R_{ko} \Delta_o^{ij} \Delta_\ell^{ij} \Gamma_i \kappa(x_i, x_j) \Gamma_j R_{ap} \Delta_p^{ij} \Delta_b^{ij}$$
$$- \sum_{ij} \delta_{ka} \Delta_b^{ij} \Gamma_i \kappa(x_i, x_j) \Gamma_j \Delta_\ell^{ij}.$$

Multiplication of this term with some vector g_{ab} (resulting from stacking the elements of the $D \times d$ matrix g into a vector) requires storage of the $d \times N \times N$ array $R\Delta$ with elements $(R\Delta)_k^{ij}$, the $D \times N \times N$ array Δ with elements Δ_ℓ^{ij}, and the $N \times N$ matrix $\Gamma\Gamma^\top \otimes K$. Multiplication then takes the form

$$[H^1 g]_{k\ell} = \sum_{j=1}^{N} \sum_{i=1}^{N} (R\Delta)_k^{ij} \Delta_\ell^{ij} \Gamma_i \Gamma_j \kappa_{x_i x_j} \qquad (5)$$
$$\times \underbrace{\left[\sum_{a=1}^{d} (R\Delta)_a^{ij} \left[\sum_{b=1}^{D} \Delta_b^{ij} g_{ab} \right] \right]}_{\text{compute once in } \mathcal{O}(N^2 dD), \text{ store in } \mathcal{O}(N^2).}$$

Since the $N \times N$ matrix in the square brackets is independent of $k\ell$, it can be reused in the dD computations required to evaluate the full matrix–vector product, so the overall computation cost of this product is $\mathcal{O}(N^2 dD)$. The other required terms are of similar form. This means that approximate inversion of the Hessian, using an iterative solver like

the Lanczos or conjugate gradient methods, is achievable in time linear in D. The methods described here are computationally feasible even for high-dimensional problems. Our implementation of the active method, released along with this text, does not yet allow this kind of scalability, but the derivations above show that it is feasible in principle.

3 APPROXIMATE MARGINALIZATION OF GAUSSIAN PROCESS HYPERPARAMETERS

To turn the approximate Gaussian belief on R into an approximate Gaussian process belief on f, the active learning algorithm (constructed in Section 4) requires an (approximate) means of integrating over the belief on R. The elements of R form hyperparameters of the GP model. The problem of dealing with uncertainty in Gaussian process hyperparameters is a general one, also faced by other, non-active, Gaussian process regression models. This section presents a novel means of approximately integrating over the hyperparameters of a GP. The most widely used approach to learning GP hyperparameters is type-II maximum likelihood estimation (evidence maximization), or maximum *a posteriori* (MAP) estimation, which both approximate the likelihood as a delta function. However, ignoring the uncertainty in the hyperparameters in this way can lead to pathologies (MacKay, 2003).

For compact notation, all hyperparameters to be marginalized will be subsumed into a vector θ. We will denote as $m_{f|\mathcal{D},\theta}(x)$ the GP posterior mean prediction for $f(x)$ conditioned on data $\mathcal{D}$ and θ, and similarly as $V_{f|\mathcal{D},\theta}(x)$ the posterior variance V of $f(x)$ conditioned on $\mathcal{D}$ and θ.

We seek an approximation to the intractable posterior for $f_* = f(x_*)$, which requires marginalization over θ:

$$p(f_* \mid \mathcal{D}) = \int p(f_* \mid \mathcal{D}, \theta)\, p(\theta \mid \mathcal{D})\, \mathrm{d}\theta. \qquad (6)$$

Assume a Gaussian conditional, $p(\theta|\mathcal{D}) = \mathcal{N}(\theta; \hat{\theta}, \Sigma)$, on the hyperparameters, such as the approximate distribution over R constructed in the preceding section. To make the integral in (6) tractable, we seek a linear approximation

$$p(f_*|\mathcal{D}, \theta) = \mathcal{N}\big(f_*; m_{f|\mathcal{D},\theta}(x_*), V_{f|\mathcal{D},\theta}(x_*)\big) \qquad (7)$$
$$\simeq q(f_*; \theta) := \mathcal{N}(f_*; a^\top \theta + b, \nu^2), \qquad (8)$$

using free parameters a, b, ν^2 to optimize the fit. The motivation for this approximation is that it yields a tractable marginal, $p(f_*|\mathcal{D}) \simeq \mathcal{N}(f_*; a^\top \hat{\theta} + b, \nu^2 + a^\top \Sigma a)$. Further, the posterior for θ typically has quite narrow width, over which $p(f_*|\mathcal{D}, \theta)$'s dependence on θ can be reasonably approximated. We choose the variables a, b, ν^2 by matching a local expansion of $q(f_* \mid \theta)$ to $p(f_*|\mathcal{D}, \theta)$. The expansion will be performed at $\theta = \hat{\theta}$, and at a $f_* = \hat{f}_*$ to be determined.

Specifically, we match as

$$\frac{\partial}{\partial f_*} q(f_*; \theta)\bigg|_{\hat{\theta}, \hat{f}_\star} = \frac{\partial}{\partial f_*} p(f_*|\mathcal{D}, \theta)\bigg|_{\hat{\theta}, \hat{f}_\star}; \qquad (9)$$

$$\frac{\partial}{\partial \theta_i} q(f_*; \theta)\bigg|_{\hat{\theta}, \hat{f}_\star} = \frac{\partial}{\partial \theta_i} p(f_*|\mathcal{D}, \theta)\bigg|_{\hat{\theta}, \hat{f}_\star}; \qquad (10)$$

$$\frac{\partial^2}{\partial f_*^2} q(f_*; \theta)\bigg|_{\hat{\theta}, \hat{f}_\star} = \frac{\partial^2}{\partial f_*^2} p(f_*|\mathcal{D}, \theta)\bigg|_{\hat{\theta}, \hat{f}_\star}; \qquad (11)$$

$$\frac{\partial^2}{\partial f_* \partial \theta_i} q(f_*; \theta)\bigg|_{\hat{\theta}, \hat{f}_\star} = \frac{\partial^2}{\partial f_* \partial \theta_i} p(f_*|\mathcal{D}, \theta)\bigg|_{\hat{\theta}, \hat{f}_\star}. \qquad (12)$$

An alternative set of constraints could be constructed by including second derivatives with respect to θ. But this would require computation scaling as $\mathcal{O}\big((\#\theta)^2\big)$, prohibitive for large numbers of hyperparameters, such as the $D \times d$ required to parameterize R for large D. We define

$$\hat{m} := m_{f|\mathcal{D},\hat{\theta}} \quad \text{and} \quad \frac{\partial \hat{m}}{\partial \theta_i} := \frac{\partial m_{f|\mathcal{D},\theta}}{\partial \theta_i}\big|_{\theta=\hat{\theta}}, \qquad (13)$$

along with analogous expressions for $\hat{V}$ and $\frac{\partial \hat{V}}{\partial \theta_i}$. Turning to solving for a, b, ν^2 and f_*, note that, firstly, (9) implies that $a^\top \hat{\theta} + b = \hat{m}$, and that (11) implies that $\nu^2 = \hat{V}$. Rearranging (10) and (12), respectively, we have

$$2a_i = \frac{\partial \hat{V}}{\partial \theta_i}\left(\frac{1}{\hat{f}_\star - \hat{m}} - \frac{\hat{f}_\star - \hat{m}}{\hat{V}}\right) + 2\frac{\partial \hat{m}}{\partial \theta_i}; \qquad (14)$$

$$2a_i = 2\frac{\partial \hat{V}}{\partial \theta_i}\frac{\hat{f}_\star - \hat{m}}{\hat{V}} + 2\frac{\partial \hat{m}}{\partial \theta_i}. \qquad (15)$$

(14) and (15) can be solved only for

$$a_i = a_{i\pm} := \pm\frac{1}{\sqrt{3\hat{V}}}\frac{\partial \hat{V}}{\partial \theta_i} + \frac{\partial \hat{m}}{\partial \theta_i}; \qquad (16)$$

$$f_* = \hat{f}_{*\pm} := \hat{m}(x_*) \pm \sqrt{\frac{\hat{V}(x_*)}{3}}. \qquad (17)$$

In particular, note that the intuitive choice $f_* = \hat{m}(x_*)$, for which $\frac{\partial}{\partial f_*} p(f_*|\mathcal{D}, \theta) = 0$, gives q inconsistent constraints related to its variation with θ. Introducing the separation of $(\hat{V}(x_*)/3)^{1/2}$ provides optimal information about the curvature of $p(f_*|\mathcal{D}, \theta)$ with θ. Hence there are two possible values, $\hat{f}_{*\pm}$, to expand around, giving a separate Gaussian approximation for each. We average over the two solutions, giving an approximation that is a mixture of two Gaussians. We then further approximate this as a single moment-matched Gaussian.

The consequence of this approximation is that

$$p(f_* \mid \mathcal{D}) \simeq \mathcal{N}\big(f_*; \tilde{m}_{f|\mathcal{D}}(x_*), \tilde{V}_{f|\mathcal{D}}(x_*)\big), \qquad (18)$$

where the marginal mean for f_* is $\tilde{m}_{f|\mathcal{D}}(x_*) := \hat{m}(x_*)$,

and the marginal variance is

$$\tilde{V}_{f|\mathcal{D}}(x_*) \coloneqq \frac{4}{3}\hat{V}(x_*) + \frac{\partial \hat{m}(x_*)}{\partial \theta}^\top \Sigma \frac{\partial \hat{m}(x_*)}{\partial \theta}$$
$$+ \frac{1}{3\hat{V}(x_*)} \frac{\partial \hat{V}(x_*)}{\partial \theta}^\top \Sigma \frac{\partial \hat{V}(x_*)}{\partial \theta} . \quad (19)$$

Figure 2 provides an illustration of our approximate marginal GP (henceforth abbreviated as MGP).

Our approach is similar to that of Osborne et al. (2012) (BBQ), for which $\tilde{V}_{f|\mathcal{D}} = V_{f|\mathcal{D},\hat{\theta}} + \frac{\partial \hat{m}}{\partial \theta}^\top \Sigma \frac{\partial \hat{m}}{\partial \theta}$. However, BBQ ignores the variation of the predictive variance with changes in hyperparameters.

To compare the two methods, we generated (from a GP) $10 \times D$ random function values, $\mathcal{D}$, where D is the problem dimension. We then trained a GP with zero prior mean and ARD covariance on that data, and performed prediction for $10 \times D$ test data. Test points, (x_*, y_*), were generated a small number (drawn from $\mathcal{U}(1,3)$) of input scales away from a training point in a uniformly random direction. The MGP and BBQ were used to approximately marginalize over all GP hyperparameters (the output scale and D input scales), computing posteriors for the test points. We considered $D \in \{5, 10, 20\}$ and calculated the mean symmetrized Kullback–Leibler divergence (SKLD) over fifty random repetitions of each experiment. We additionally tested on two real datasets:[4] yacht hydrodynamics (Gerritsma et al., 1981) and (centered) concrete compressive strength (Yeh, 1998). In these two, a random selection of 50 and 100 points, respectively, was used for training and the remainder for testing. All else was as above, with the exception that ten random partitions of each dataset were considered.

We evaluate performance using the SKLD between approximate posteriors and the "true" posterior (obtained using a run of slice sampling (Neal, 2003) with 10^5 samples and 10^4 burn-in); the better the approximate marginalization, the smaller this divergence. We additionally measured the average negative predictive log-likelihood, $-\mathbb{E}\big[\log p(y_* \mid x_*, \mathcal{D})\big]$, on the test points (x_*, y_*). Results are displayed in Table 1; it can be seen that the MGP provides both superior predictive likelihoods and posteriors closer to the "true" distributions. The only exception is found on the yacht dataset, where the MGP's SKLD score was penalized for having predictive variances that were consistently slightly larger than the "true" variances. However, these conservative variances, in better accommodating test points that were unexpectedly large or small, led to better likelihoods than the consistently over-confident MAP and BBQ predictions.

4 ACTIVE LEARNING OF GAUSSIAN PROCESS HYPERPARAMETERS

Now we turn to the question of actively selecting observation locations to hasten our learning of R. We employ an active learning strategy due to Houlsby et al. (2011), known as *Bayesian active learning by disagreement* (BALD). The idea is that, in selecting the location x of a function evaluation f to learn parameters θ, a sensible utility function is the expected reduction in the entropy of θ,

$$\upsilon(x) \coloneqq H(\Theta) - H(\Theta \mid F) = H(F) - H(F \mid \Theta), \quad (20)$$

also equal to the mutual information $I(\Theta; F)$ between f and θ. Mutual information, unlike differential entropies, is well-defined: the BALD objective is insensitive to changes in the representation of f and θ. The right-hand side of (20), the expected reduction in the entropy of f given the provision of θ, is particularly interesting. For our purposes, θ will parameterize $R \in \mathbb{R}^{d \times D}$; that is, θ is very high-dimensional, making the computation of $H(\Theta)$ computationally demanding. In contrast, the calculation of the entropy of $f \in \mathbb{R}$ is usually easy or even trivial. The right-hand side of (20) is particularly straightforward to evaluate under the approximation of Section 3, for which $p(f \mid \mathcal{D}, \theta)$ and the marginal $p(f \mid \mathcal{D})$ are both Gaussian. Further, under this approximation, $p(f \mid \mathcal{D}, \theta)$ has variance $\nu^2 = \hat{V}$ that is independent of θ, hence, $H(F \mid \Theta) = H(F \mid \Theta = \hat{\theta})$. We henceforth consider the equivalent but transformed utility function

$$\upsilon'(x) = \tilde{V}_{f|\mathcal{D}}(x) \left(V_{f|\mathcal{D},\hat{\theta}}(x) \right)^{-1}. \quad (21)$$

The MGP approximation has only a slight influence on this objective – Figure 2 compares it to a full MCMC-derived marginal. With reference to (19), (21) encourages evaluations where the posterior mean and covariance functions are most sensitive to changes in θ (Figure 3), normalized by the variance in f: such points are most informative about the hyperparameters. An alternative to BALD is found in *uncertainty sampling*. Uncertainty sampling selects the location with highest variance, that is, its objective is simply $H(F)$, the first term in the BALD objective. This considers only the variance of a single point, whereas the BALD objective rewards points that assist in the learning of embeddings, thereby reducing the variance associated with all points. An empirical comparison of our method against uncertainty sampling follows below.

4.1 ACTIVE LEARNING OF LINEAR EMBEDDINGS FOR GAUSSIAN PROCESSES

To apply BALD to learning the linear embedding of a Gaussian process, we consider the case $R \subset \theta$; the GP hyperparameters define the embedding described in Section 2. Figure 4 demonstrates an example of active learning for the embedding of a two-dimensional function.
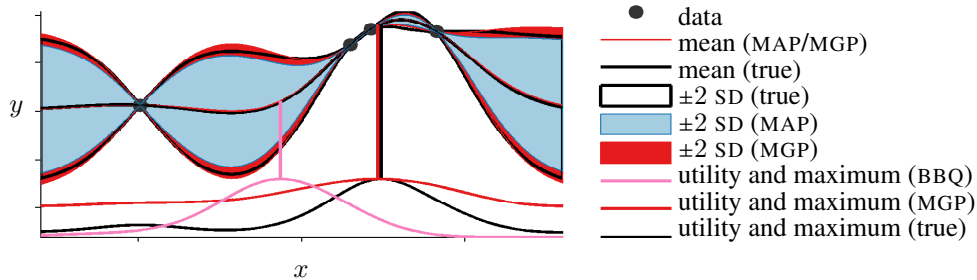
Figure 2: Approximate marginalization (MGP) of covariance hyperparameters θ increases the predictive variance to closely match the "true" posterior (obtained using slice sampling with 10^5 samples). BBQ (Osborne et al., 2012) provides a standard deviation differing from the MAP standard deviation by less than 3.1% everywhere, and would hence be largely invisible on this plot. The bottom of the figure displays the (normalized) mutual information $I\big(\Theta; F(x)\big)$ (equal to the BALD utility function $\upsilon(x)$) for the various methods, and their maxima, giving the optimal positions for the next function evaluations. The MGP position is very close to the true position.

Table 1: Mean negative log-likelihood for test points and mean SKLD (nats) between approximate and true posteriors. Both metrics were averaged over test points, as well as over fifty and ten random repeats for synthetic and real experiments, respectively.

		$-\mathbb{E}\big[\log p(y_* \mid x_*, \mathcal{D})\big]$			SKLD		
problem	dim	MAP	BBQ	MGP	MAP	BBQ	MGP
synthetic	5	3.58	2.67	**1.73**	0.216	0.144	**0.0835**
synthetic	10	3.57	3.10	**1.86**	0.872	0.758	**0.465**
synthetic	20	1.46	1.41	**0.782**	1.01	0.947	**0.500**
yacht	6	123.0	97.8	**56.8**	0.0322	**0.0133**	0.0323
concrete	8	$2.96 \cdot 10^9$	$2.96 \cdot 10^9$	**$1.67 \cdot 10^9$**	0.413	0.347	**0.337**

The latent model of lower dimension renders optimizing an objective with domain $\mathcal{X}$ (e.g., $f(x)$, or the BALD objective) feasible even for high-dimensional $\mathcal{X}$. Instead of direct search over $\mathcal{X}$, one can choose a $u \in \mathcal{U}$, requiring search over only the low-dimensional $\mathcal{U}$, and then evaluate the objective at an $x \in \mathcal{X}$ for which $u = x R^\top$. A natural choice is the x which is most likely to actually map to u under R, that is, the x for which $p(u \mid x)$ is as tight as possible. For example, we could minimize $\log \det \operatorname{cov}[u \mid x]$, subject to $\mathbb{E}[u \mid x] = x \hat{R}^\top$, by solving the appropriate program. For $d = 1$, this is a quadratic program that minimizes the variance $x \Sigma x^\top$ under the equality constraint. Finally, we evaluate the objective at the solution.

For simplicity, we will henceforth assume $\mathcal{X} = [-1, 1]^D$. For any box-bounded problem, there is an invertible affine transformation mapping the box to this $\mathcal{X}$; this then requires only that R is composed with this transformation. Further, define the signature of the ith row of R to be $[\operatorname{sign}(R_{i1}), \operatorname{sign}(R_{i2}), \ldots]$. Then, for the ith coordinate, the maximum and minimum value obtained by mapping the corners of $\mathcal{X}$ through R are achieved by the corner matching this signature and its negative. This procedure defines the extreme corners of the search volume $\mathcal{U}$.

Consider the typical case in which we take $\tilde{\mu}$ as constant and $\tilde{\kappa}$ as isotropic (e.g., the exponentiated quadratic (1)). Since $p(f \mid X, R)$ is then invariant to orthogonal transformations of R in $\mathbb{R}^d$, there is no unique embedding. In the special case $d = 1$, R and $-R$ are equivalent. For most means and covariances there will be similar symmetries, and likely ever more of them as d increases.[5] We therefore evaluate the performance of our algorithms not by comparing estimated to true Rs, which is difficult due to these symmetries, but rather in the direct predictive performance for f.

4.2   ACTIVE LEARNING OF LINEAR EMBEDDINGS EXPERIMENTS

We now present the results of applying our proposed method for learning linear embeddings on both real and synthetic data with dimension up to $D = 318$. Given a function $f: \mathcal{X} \to \mathbb{R}$ with a known or suspected low-dimensional embedding, we compare the following methods for sequentially

---

[5]An alternative would be to place a prior on the *Stiefel manifold* rather than directly on R; the Stiefel manifold accounts for most of these symmetries Minka (2000). This would require the rows of R to be orthogonal.
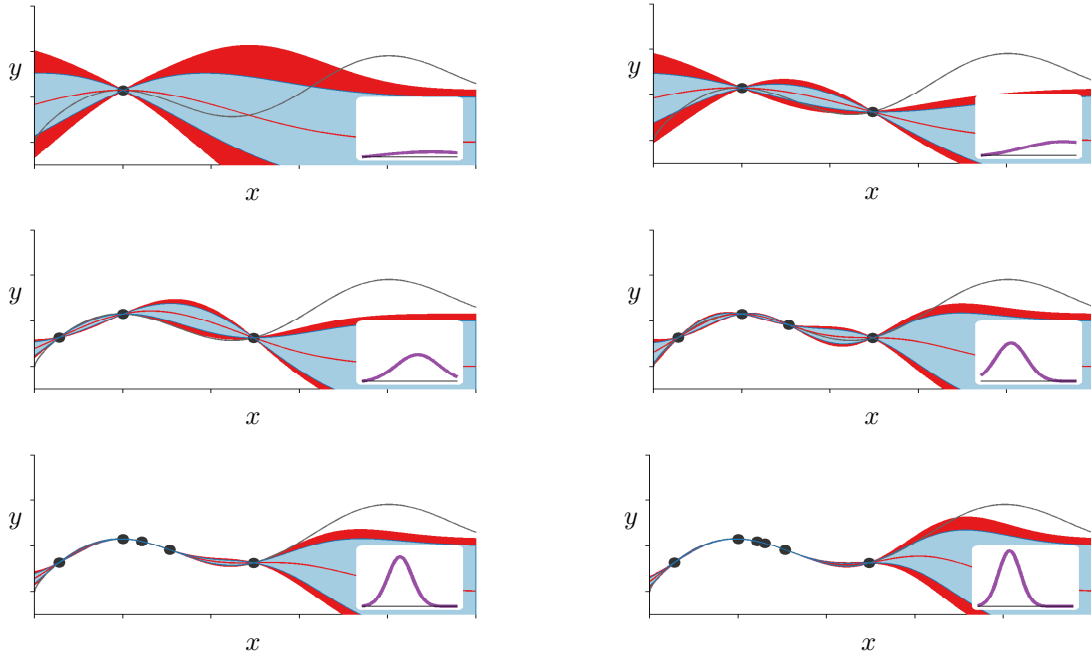
Figure 3: Active learning of the length scale of a one-dimensional GP (beginning at the top left and continuing across and then down): the next sample is taken where the MAP and approximate variances maximally disagree, normalized by the MAP variance. Samples are taken at a variety of separations to refine belief about the length scale. The inset plots (all of which share axes) display the approximate posteriors over log-length scales, which tighten with increasing numbers of samples. The legend is identical to that of Figure 2.

selecting $N = 100$ observations from the domain $[-1, 1]^D$: random sampling (RAND), a Latin hypercube design (LH), uncertainty sampling (UNC), and BALD. UNC and BALD use identical models (Laplace approximation on R followed by MGP) and hyperparameter priors. We also compare with LASSO, choosing the regularization parameter by minimizing squared loss on the training data. The functions that these methods are compared on are:

- Synthetic in-model data drawn from a GP matching our model with an embedding drawn from our prior, for $d \in \{2, 3\}$ and $D \in \{10, 20\}$.

- The Branin function, a popular test function for global optimization ($d = 2$), embedded in $D \in \{10, 20\}$ via an embedding drawn from our prior.

- The temperature data[6] described in Snelson & Ghahramani (2006) ($D = 106$), with $d = 2$. The associated prediction problem concerns future temperature at a weather station, given the output of a circulation model. The training and validation points were combined to form the dataset, comprising 10 675 points.

- The normalized "communities and crime" (C&C)

dataset from the UCI Machine Learning Repository[7] ($D = 96$), with $d = 2$. The task here is to predict the number of violent crimes per capita in a set of US communities given historical data from the US Census and FBI. The LEMAS survey features were discarded due to missing values, as was a single record missing the "AsianPerCap" attribute, leaving 1 993 points.

- The "relative location of CT slices on axial axis" dataset from the UCI Machine Learning Repository[8] ($D = 318$), with $d = 2$. The task is to use features extracted from slices of a CT scan to predict its vertical location in the human body. Missing features were replaced with zeros. Only axial locations in the range $[50, 60]$ were used. Features that did not vary over these points were discarded, leaving 3 071 points.

The CT slices and communities and crime datasets are, respectively, the highest- and third-highest-dimensional regression datasets available in the UCI Machine Learning Repository with real attributes; in second place is an unnormalized version of the C&C dataset.

---

[6]http://theoval.cmp.uea.ac.uk/~gcc/
competition.

[7]http://archive.ics.uci.edu/ml/datasets/
Communities+and+Crime.

[8]http://archive.ics.uci.edu/ml/datasets/
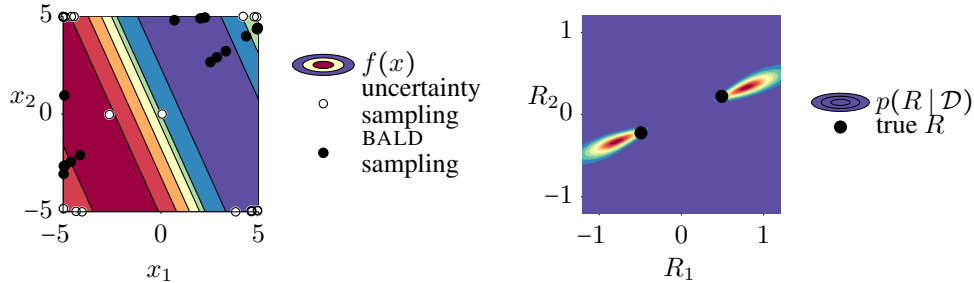Relative+location+of+CT+slices+on+axial+
axis.

Figure 4: Left: Twenty samples selected by uncertainty sampling and BALD for a function f with a one-dimensional linear embedding. Note that uncertainty sampling prefers corner locations, to maximize the variance in u, $x\Sigma x^\top$, and hence the variance in f; these points are less useful for learning f and its embedding than those selected by BALD. Right: the posterior over embeddings returned by the BALD samples, concentrated near the true embedding (equivalent under negation).

Table 2: The combination of MGP and BALD actively learns embeddings whose predictive performance improves on alternatives. Average negative log predictive probability and average RMSE on test functions for various D and d.

dataset	D/d	$-\mathbb{E}\big[\log p(y_* \mid x_*, \hat{R})\big]$				RMSE				
		RAND	LH	UNC	BALD	RAND	LH	UNC	BALD	LASSO
synthetic	10/2	0.272	0.224	−0.564	**−0.649**	0.412	0.371	0.146	**0.138**	0.842
synthetic	10/3	0.711	0.999	0.662	**0.465**	0.553	0.687	0.557	**0.523**	0.864
synthetic	20/2	0.804	0.745	0.749	**0.470**	0.578	0.549	0.551	**0.464**	0.853
synthetic	20/3	1.07	1.10	1.04	**0.888**	0.714	0.740	0.700	**0.617**	0.883
Branin	10/2	3.87	3.90	1.58	**0.0165**	18.2	17.8	3.63	**2.29**	40.0
Branin	20/2	4.00	3.70	**3.55**	3.63	18.3	14.8	**13.4**	15.0	39.1
communities & crime	96/2	1.09	—	1.17	**1.01**	0.720	—	0.782	**0.661**	1.16
temperature	106/2	0.566	—	0.583	**0.318**	0.423	—	0.427	**0.328**	0.430
CT slices	318/2	1.30	—	1.26	**1.16**	0.878	—	0.845	**0.767**	0.900

For the synthetic and Branin problems, where the true embedding R was chosen explicitly, we report averages over five separate experiments differing only in the choice of R. On these datasets, the UNC and BALD methods selected points by successively maximizing their respective objectives on a set of 20 000 fixed points in the input domain, 10 000 selected uniformly in $[−1, 1]^D$ and 10 000 selected uniformly in the unit D-sphere. For a given D, these points were fixed across methods and experimental runs. This choice allows us to compare methods based only on their objectives and not the means of optimizing them.

For the real datasets (temperature, communities and crimes, and CT slices), each method selected from the available points; LH is incapable of doing so and so is not considered on these datasets. The real datasets were further processed by transforming all features to the box $[−1, 1]^D$ via the "subtract min, divide by max" map and normalizing the outputs to have zero mean and unit variance. For the synthetic problems, we added i.i.d. Gaussian observation noise with variance $\sigma^2 = (0.1)^2$. For the remaining problems, the datapoints were used directly (assuming that these real measurements already reflect noise).

After each method selected 100 observations, we compare the quality of the learned embeddings by fixing the hyper-parameters of a GP to the MAP embedding at termination and measuring predictive performance. This is intended to emulate a fixed-budget embedding learning phase followed by an experiment using only the most likely R. We chose $N = 100$ training points and 1 000 test points uniformly at random from those available; these points are common to all methods. We report root-mean-square error (RMSE) and the average negative predictive log-likelihood on the test points. The RMSE measures predictive accuracy, whereas the log-likelihood additionally captures the accuracy of variance estimates. This procedure was repeated 10 times for each experiment; the reported numbers are averages.

The embedding prior $p(R)$ was set to be i.i.d. zero-mean Gaussian with standard deviation $5/4 D^{-1}$. This choice roughly implies that we expect $[−1, 1]^D$ to map approximately within $[−2.5, 2.5]^d$, a box five length scales on each side, under the unknown embedding. This prior is extremely diffuse and does not encode any structure of R beyond preferring low-magnitude values. At each step, the mode of the log posterior over R was found using using L-BFGS, starting from both the previous best point and one random restart drawn from $p(R)$.

The results are displayed in Table 2. The active algorithm achieves the most accurate predictions on all but one prob-

lem, including each of the real datasets, according to both metrics. These results strongly suggest an advantage for actively learning linear embeddings.

5 CONCLUSIONS

Active learning in regression tasks should include hyperparameters, in addition to the function model itself. Here we studied simultaneous active learning of the function and a low-dimensional linear embedding of its input domain. We also developed a novel means of approximately integrating over the hyperparameters of a GP model. The resulting algorithm addresses needs in a number of domains, including Bayesian optimization, Bayesian quadrature, and also the underlying idea of nonparametric Gaussian regression itself. Empirical evaluation demonstrates the efficacy of the resulting algorithm on both synthetic and real problems in up to 318 dimensions, and an analysis of computational cost shows that the algorithm can, at least in principle, be scaled to problems of much larger dimensionality as well.

ACKNOWLEDGMENTS

References

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

Brochu, E., Cora, V. M., and de Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Candes, E. and Tao, T. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

Carpentier, A. and Munos, R. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. *arXiv preprint arXiv:1205.4094*, 2012.

Chen, B., Castro, R., and Krause, A. Joint optimization and variable selection of high-dimensional Gaussian processes. In *Proceedings of the 29th Annual International Conference on Machine Learning*, 2012.

Gerritsma, J., Onnink, R., and Versluis, A. Geometry, resistance and stability of the delft systematic yacht hull series. *International Shipbuilding Progress*, 28(328):276–297, 1981.

Girard, A. and Murray-Smith, R. Gaussian processes: Prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-series. In *Switching and Learning in Feedback Systems*, number 3355 in Lecture Notes in Computer Science. Springer, 2005.

Guestrin, C., Krause, A., and Singh, A. P. Near-optimal sensor placements in Gaussian processes. In *Proceedings of the 22nd Annual International Conference on Machine Learning*, pp. 265–272. ACM, 2005.

Hennig, P. and Kiefel, M. Quasi-Newton methods – A new direction. In *Proceedings of the 29th Annual International Conference on Machine Learning*, 2012.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Hutter, F. *Automated configuration of algorithms for solving hard computational problems*. PhD thesis, University of British Columbia, 2009.

Hutter, F., Hoos, H., and Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pp. 507–523. Springer, 2011.

Iwata, T., Houlsby, N., and Ghahramani, Z. Active learning for interactive visualization. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS 2013)*, (31), 2013.

Lawrence, N. D. A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models. *Journal of Machine Learning Research*, 13:1609–1638, 2012.

MacKay, D. J. C. Bayesian interpolation. *Neural Computation*, 4 (3):415–447, 1992a.

MacKay, D. J. C. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992b.

MacKay, David J. C. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

McHutchon, A. and Rasmussen, C. E. Gaussian process training with input noise. In *Advances in Neural Information Processing Systems 24*, 2011.

Minka, T. Automatic choice of dimensionality for PCA. Technical Report 514, MIT Media Laboratory, 2000.

Neal, R. M. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.

Neal, R. M. Slice sampling. *Annals of Statistics*, 31(3):705–767, 2003.

O'Hagan, A. Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260, 1991.

Osborne, M. A., Duvenaud, D., Garnett, R., Rasmussen, C. E., Roberts, S. J., and Ghahramani, Z. Active learning of model evidence using Bayesian quadrature. In *Advances in Neural Information Processing Systems 25*, pp. 46–54, 2012.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Snelson, E. and Ghahramani, Z. Variable noise and dimensionality reduction for sparse Gaussian processes. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI 2006)*, pp. 461–468, 2006.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., and de Freitas, N. Bayesian optimization in a billion dimensions via random embeddings. *arXiv preprint arXiv:1301.1942*, 2013.

Williams, C. K. I. and Rasmussen, C. E. Gaussian processes for regression. In *Advances in Neural Information Processing Systems 8*, 1996.

Yeh, I-C. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28 (12):1797–1808, 1998.