ACTIVELY LEARNING HYPERPARAMETERS FOR GPS

Roman Garnett
Washington University in St. Louis

12.10.2016

Joint work with
Michael Osborne (University of Oxford)
Philipp Hennig (MPI Tübingen)

INTRODUCTION

Learning hyperparameters

Problem

- Gaussian processes (GPs) are powerful models able to express a wide range of structure in nonlinear functions.
- This power is sometimes a *curse,* as it can be very difficult to determine appropriate values of *hyperparameters*, especially with small datasets.

Small datasets

- Small datasets are inherent in situations when the function of interest is very *expensive*, as is typical in *Bayesian optimization.*
- Success on these problems hinges on accurate modeling of *uncertainty*, and undetermined hyperparameters can contribute a great deal (*often hidden!*).
- The traditional approach in these scenarios is to spend some portion of the budget on *model-agnostic* initialization (Latin hypercubes, etc.)
- We present a *model-driven* approach here.

Motivating problem: Learning embeddings

- High-dimensionality has *stymied* the progress of model-based approaches to many machine learning tasks.
- In particular, *Gaussian processes* approaches remain *intractable* for large numbers of input variables.
- An old idea for combating this problem is to exploit *low-dimensional* structure in the function, the most simple example of which is a *linear embedding.*

Learning embeddings for GPs

- We want to learn a function

$$f : \mathbb{R}^D \to \mathbb{R},$$

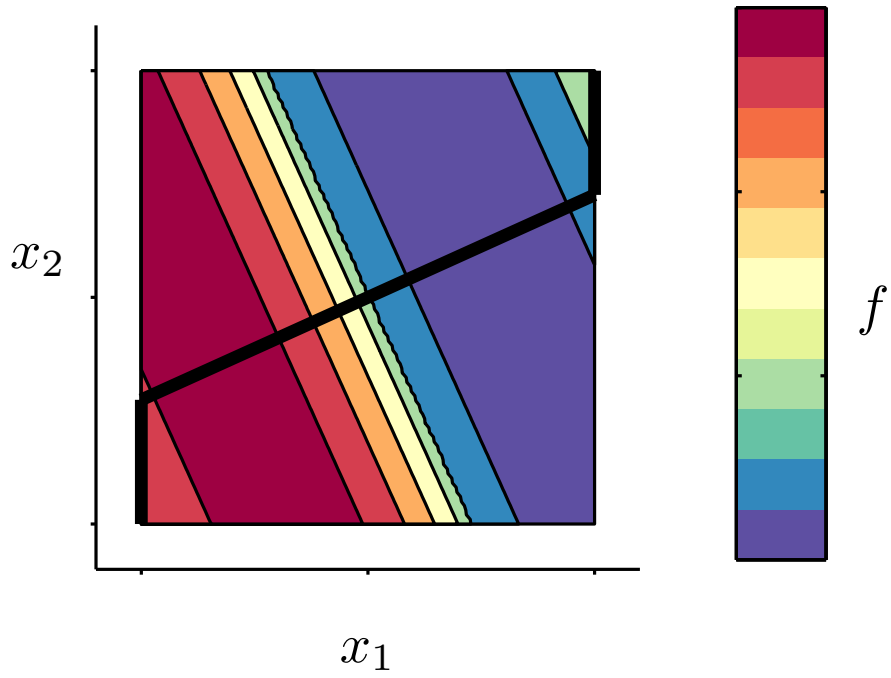
  where D is very large.
- We assume that f has low *intrinsic dimension*, that is, that there is a function $g : \mathbb{R}^d \to \mathbb{R}$ such that

$$f(x) = g(Rx),$$

  where $R \in \mathbb{R}^{d \times D}$ is a matrix defining a *linear embedding.*

Example

- Here $f : \mathbb{R}^2 \to \mathbb{R}$ ($D = 2$), but only depends on a one-dimensional projection of x ($d = 1$).
- All function values are realized along the *black line.*



The GP model

If we knew the embedding R, modeling f would be *straightforward.* Our model for f given the embedding R is a zero-mean Gaussian process:

$$p(f \mid R) = \mathcal{GP}(f; 0, K),$$

with

$$K(x, x'; R) = \kappa(Rx, Rx'),$$

where κ is a covariance on $\mathbb{R}^d \times \mathbb{R}^d$.

The GP model

If κ is the familiar squared-exponential, then

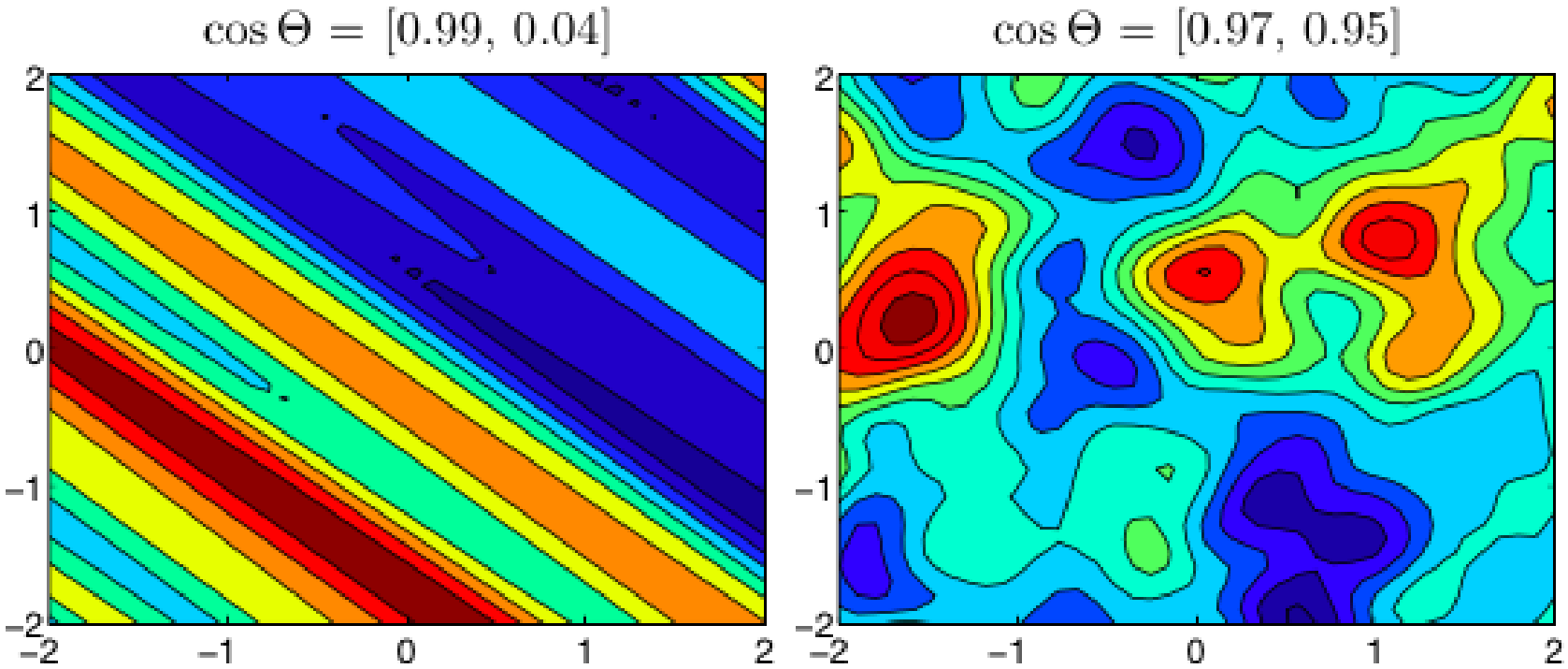
$$K(x, x'; R, \gamma) = \gamma^2 \exp\left[-\frac{1}{2}(x - x')^\top R^\top R(x - x')\right].$$

This is a low-rank *Mahalanobis* covariance, also known as a *factor analysis* covariance.

Our approach

- Our goal is to learn R (in general, any θ) as quickly as possible!
- Unlike previous approaches, which focus on *random embeddings* (Wang, et al. 2013), we focus on *learning the embedding directly.*

What can happen with random choices



Djolonga, et al. NIPS 2013

LEARNING THE HYPERPARAMETERS

Learning the hyperparameters

We maintain a *probabilistic belief* on θ. We start with a prior

$$p(\theta),$$

and given data $\mathcal{D}$ we find the (approximate) posterior

$$p(\theta \mid \mathcal{D}).$$

The uncertainty in θ (in particular, its *entropy*) measures our progress!

The prior

The prior is *arbitrary*, but here we took diffuse independent prior distribution on each entry:

$$p(\theta_i) = \mathcal{N}(\theta_i; 0, \sigma_i^2).$$

Could also use something more sophisticated.

The posterior

Now, given observations $\mathcal{D}$, we *approximate* the posterior distribution on θ:

$$p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\theta; \hat{\theta}, \Sigma).$$

The method of approximation is also *arbitrary*, but we took a Laplace approximation.

SELECTING INFORMATIVE POINTS

Active learning

Selecting informative points

- We wish to sequentially *sample the most informative point* about θ.
- We suggest maximizing the *mutual information* between the observed function value and the hyperparameters, particularly in the form known as *Bayesian active learning by disagreement* (BALD).[1]

$$x^* = \arg\max_x H[y \mid x, \mathcal{D}] - \mathbb{E}_\theta\Big[H[y \mid x, \mathcal{D}, \theta]\Big].$$

[1] Houlsby, et al. BAYESOPT 2011

BALD

Breaking this down, we want to find points with high *marginal uncertainty* (à la uncertainty sampling)...

$$x^* = \arg\max_x H[y \mid x, \mathcal{D}] - \mathbb{E}_\theta\big[H[y \mid x, \mathcal{D}, \theta]\big].$$

BALD

. . . but would have *low* uncertainty if we *knew the hyperparameters* θ:

$$x^* = \arg\max_x H[y \mid x, \mathcal{D}] - \mathbb{E}_\theta\big[H[y \mid x, \mathcal{D}, \theta]\big].$$

BALD

- That is, we want to find points where the competing models (one for each value of θ) are all certain, but *disagree highly* with each other.
- These points are the most informative points about the hyperparameters! (We can discard hyperparameters that were confident about the *wrong answer*).

Computation of BALD

How can we compute or approximate the BALD objective for our model?

$$x^* = \arg\max_x H[y \mid x, \mathcal{D}] - \mathbb{E}_\theta\big[H[y \mid x, \mathcal{D}, \theta]\big].$$

The first term (marginal uncertainty in y) is especially *troubling*...

LEARNING THE FUNCTION

Approximate marginalization of GP hyperparameters

Learning the function

Given data $\mathcal{D}$, and an input x^*, we wish to capture our belief about the associated latent value f^*, *accounting for uncertainty in* θ:

$$p(f^* \mid x^*, \mathcal{D}) = \int p(f^* \mid x^*, \mathcal{D}, \theta) p(\theta \mid \mathcal{D}) \, \mathrm{d}\theta.$$

We provide an *approximation* called the "marginal GP" (MGP).

The MGP

The result is this:

$$p(f^* \mid x^*, \mathcal{D}) \approx \mathcal{N}(f^*; m^*_{\mathcal{D}}, C^*_{\mathcal{D}}),$$

where

$$m^*_{\mathcal{D}} = \mu^*_{\mathcal{D},\hat{\theta}}.$$

The approximate mean is the *MAP posterior mean,* and...

The MGP

$$C^*_{\mathcal{D}} = \frac{4}{3} V^*_{\mathcal{D},\hat{\theta}} + \frac{\partial \mu^*}{\partial \theta}^\top \Sigma \frac{\partial \mu^*}{\partial \theta} + (3V^*_{\mathcal{D},\hat{\theta}})^{-1} \frac{\partial V^*}{\partial \theta}^\top \Sigma \frac{\partial V^*}{\partial \theta}.$$

The variance is *inflated* according to how the posterior mean and posterior variance change with the hyperparameters.

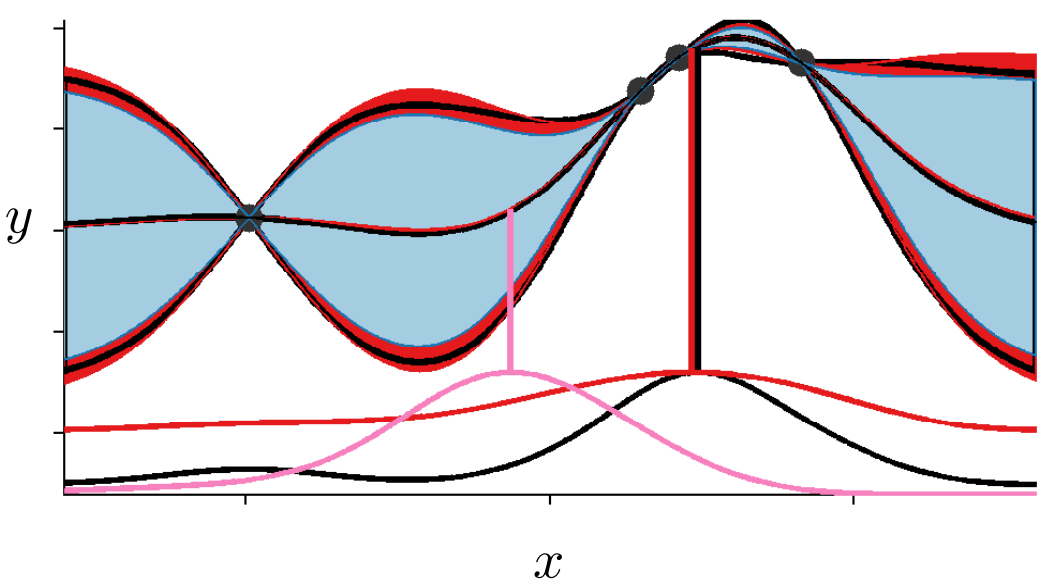
Return to BALD

The MGP gives us a *simple approximation* to the BALD objective; we maximize the following simple objective:

$$\frac{C^*_{\mathcal{D}}}{V^*_{\mathcal{D},\hat{\theta}}}.$$

So we sample the point with maximal *variance inflation*. This is the point where the plausible hyperparameters *maximally disagree* under our approximation!

BALD and the MGP



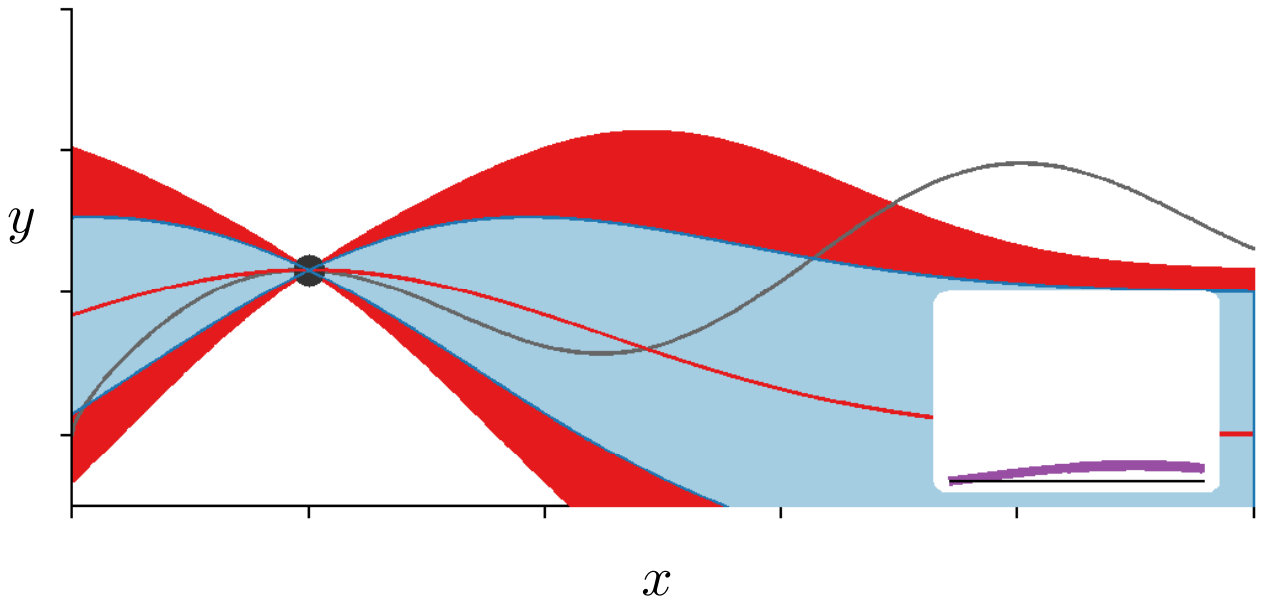
data
mean (map/MGP)
mean (true)
± 2 sd (true)
± 2 sd (map)
± 2 sd (MGP)
utility and maximum (BBQ)
utility and maximum (MGP)
utility and maximum (true)

EXAMPLE

Example

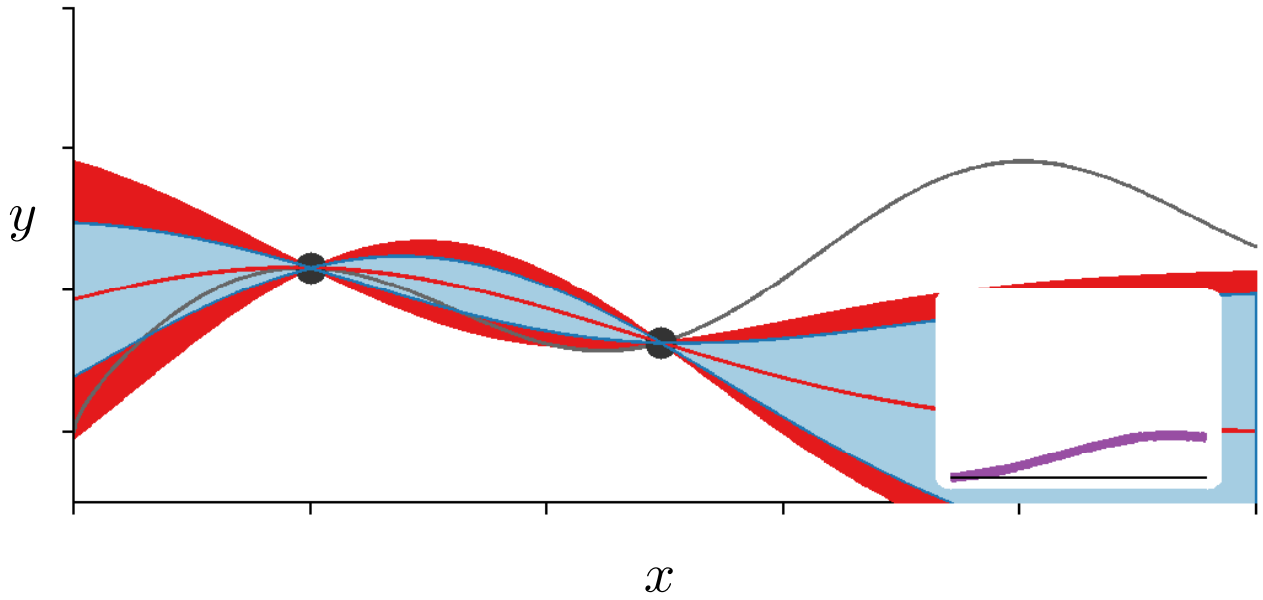
Consider a simple *one-dimensional* example (here R is simply an inverse length scale).

- The blue envelope shows the uncertainty given by the *MAP embedding*.
- The red envelope shows the additional uncertainty due to *not knowing the embedding.*
- We sample where the ratio of these is *maximized.*

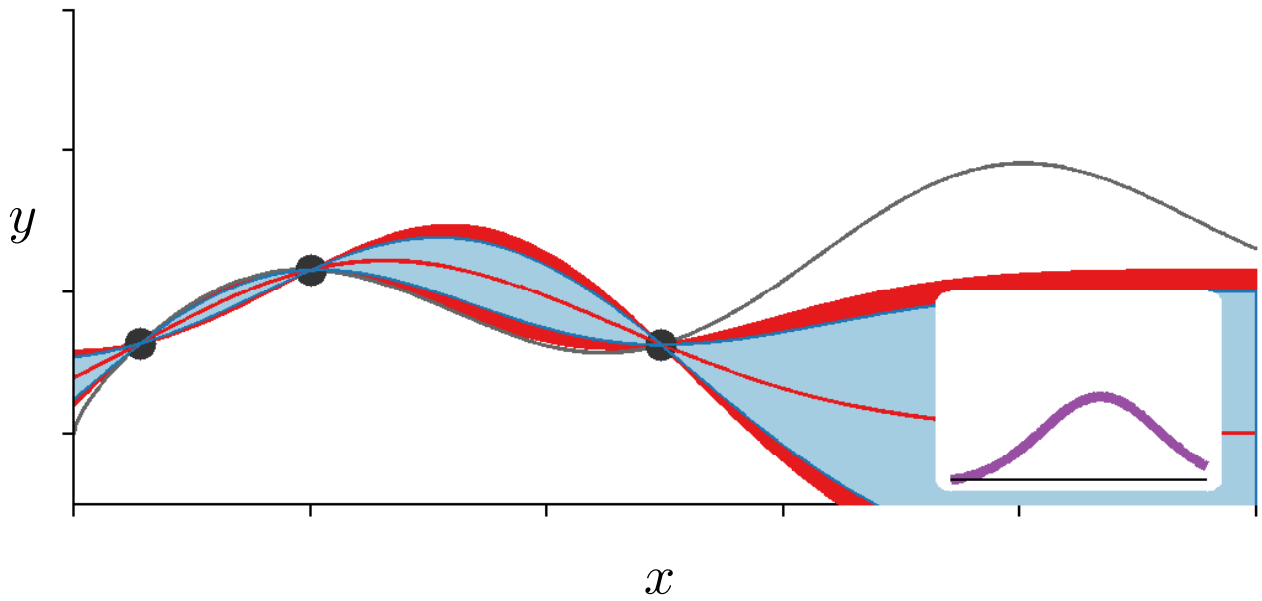


Example

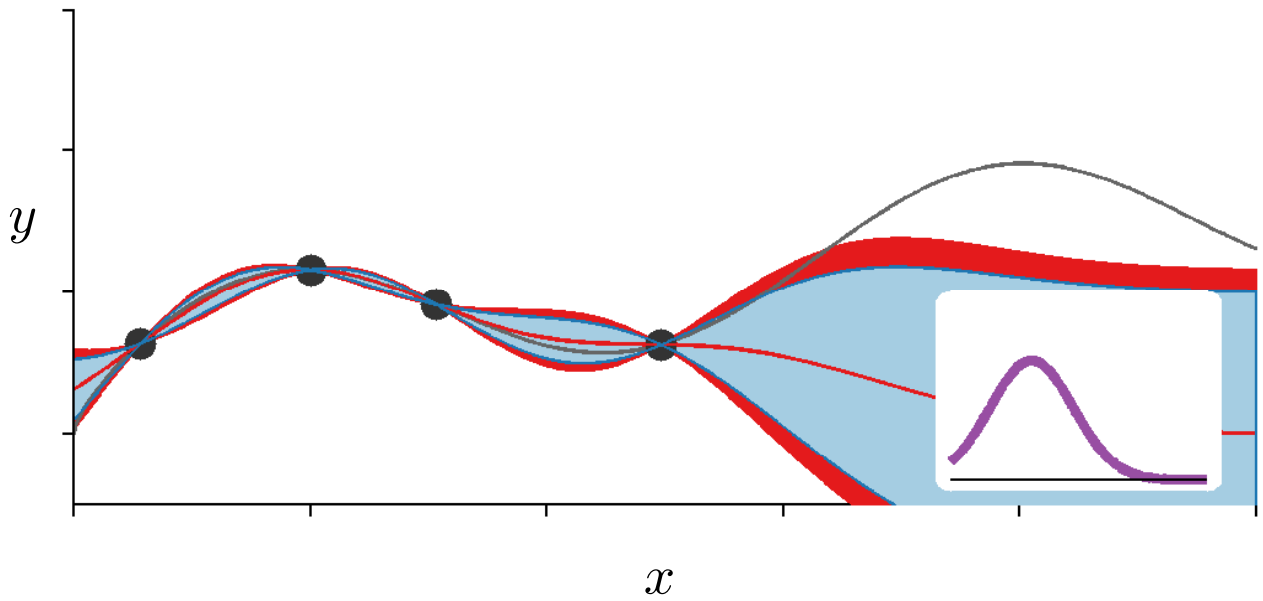
The inset shows our belief over $\log R$, it *tightens* as we continue to sample.



Example

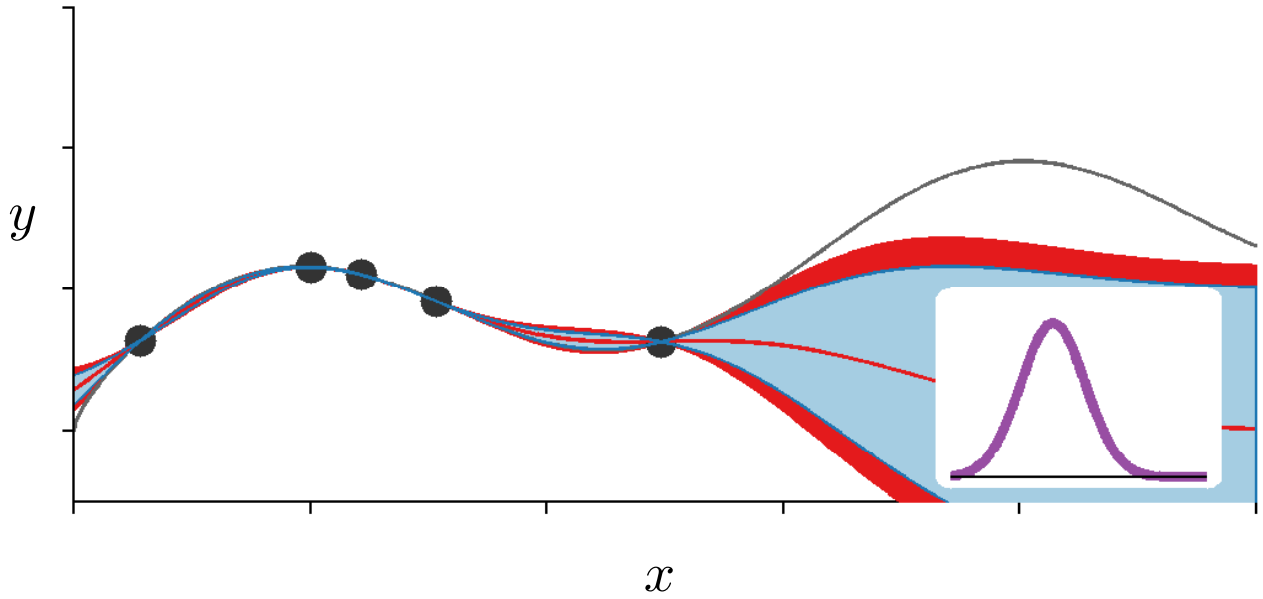


Example



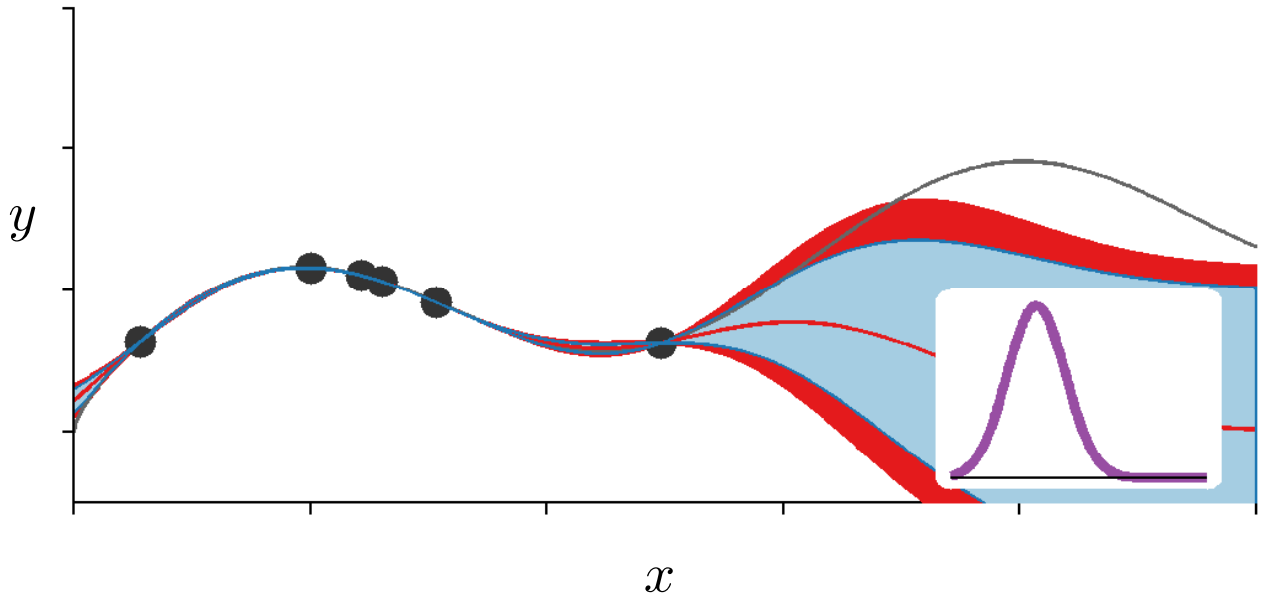
Example

We sample at a variety of separations to *further refine our belief* about R.

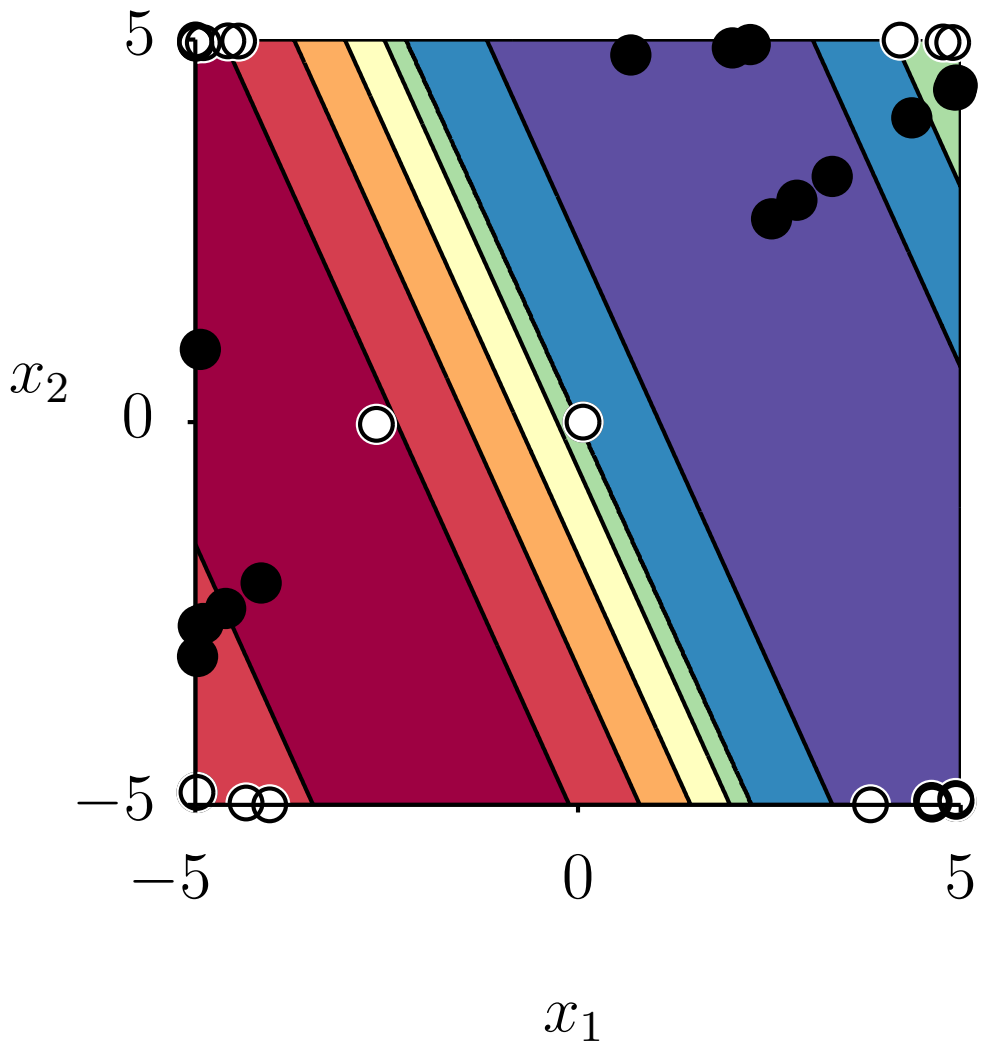


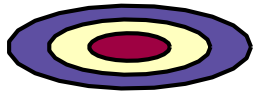
Example

Notice that we are *relatively uncertain* about many function values! Nonetheless, we are effectively learning R.



$2d$ example



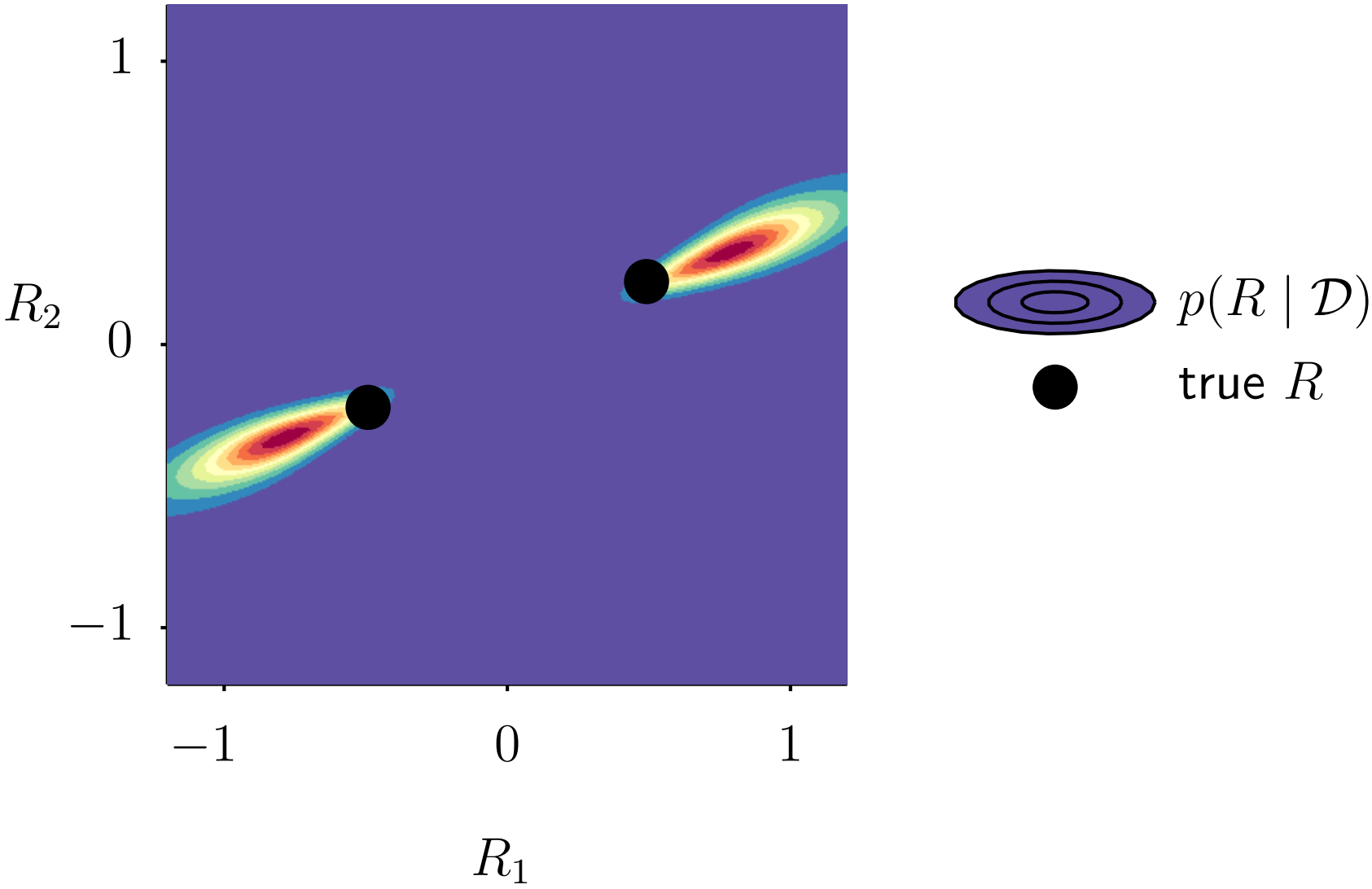


$f(x)$

uncertainty sampling

BALD sampling

$2d$ example



Results

- We have tested this approach on numerous synthetic and real-world regression problems up to dimension $D = 318$, and our performance was significantly superior to:
  - random sampling,
  - Latin-hypercube designs, and
  - uncertainty sampling.

Test setup

For each method/dataset, we:

- Began with a *single observation* of the function at the center of the (box-bounded) domain,
- Allowed each method to select a sequence of $n = 100$ observations,
- Given the resulting training data, found the *MAP* hyperparameters, and
- Used these hyperparameters to test on a *held-out* set of 1000 points, measuring RMSE and negative log likelihood.

Results: RMSE

Choosing 100 observations, predicting on 1000 more.

dataset	D/d	RAND	LH	UNC	BALD
synthetic	10/2	0.412	0.371	0.146	0.138
synthetic	10/3	0.553	0.687	0.557	0.523
synthetic	20/2	0.578	0.549	0.551	0.464
synthetic	20/3	0.714	0.740	0.700	0.617
Branin	10/2	18.2	17.8	3.63	2.29
Branin	20/2	18.3	14.8	13.4	15.0
communities & crime	96/2	0.720	—	0.782	0.661
temperature	106/2	0.423	—	0.427	0.328
CT slices	318/2	0.878	—	0.845	0.767

Reminder

The framework we have presented for actively learning linear embeddings is *completely general;* we can use it for actively learning hyperparameters in *any GP model!*

Question

Both these approaches suggest a *two-stage approach* for optimization. Is this necessary? Can we use BALD to learn the embedding while simultaneously optimizing the function?

Code

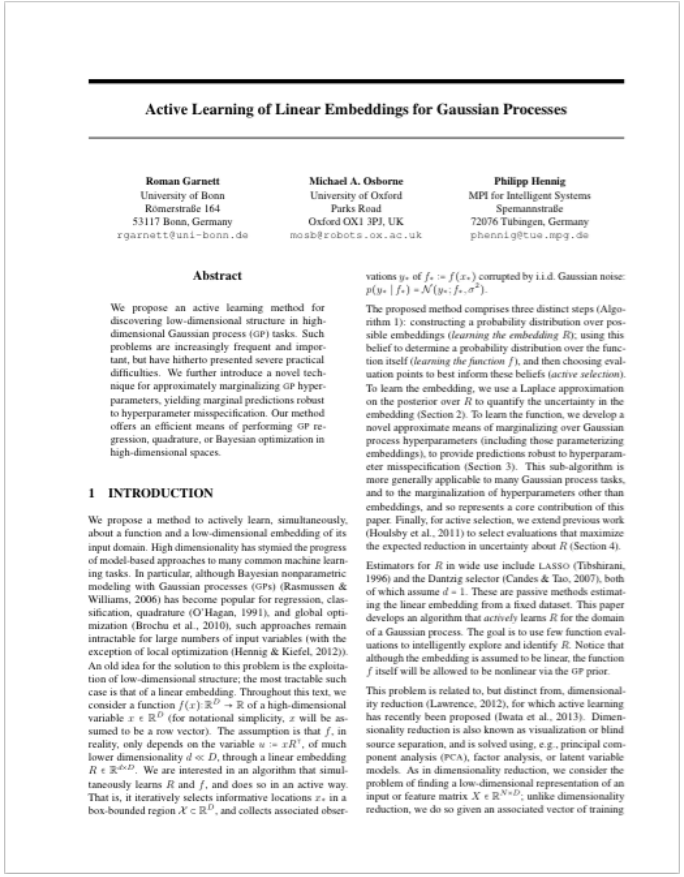
github.com/rmgarnett/
active_gp_hyperlearning

PAPER

For more details

UAI 2014

Actively Learning Linear Embeddings for Gaussian Processes, UAI 2014.



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 $f(x) : \mathbb{R}^D \to \mathbb{R}$ 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 $u := xR^\intercal$, 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 observations y_* of $f_* := f(x_*)$ 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

Extension: NIPS 2015

Extension to *model selection*, one step closer to fully automated Bayesian optimization!

Bayesian Active Model Selection with an Application to Automated Audiometry, NIPS 2015.

Bayesian Active Model Selection with an Application to Automated Audiometry

Jacob R. Gardner
CS, Cornell University
Ithaca, NY 14850
jrg365@cornell.edu

Gustavo Malkomes
CSE, WUSTL
St. Louis, MO 63130
luizgustavo@wustl.edu

Roman Garnett
CSE, WUSTL
St. Louis, MO 63130
garnett@wustl.edu

Kilian Q. Weinberger
CS, Cornell University
Ithaca, NY 14850
kqw4@cornell.edu

Dennis Barbour
BME, WUSTL
St. Louis, MO 63130
dbarbour@wustl.edu

John P. Cunningham
Statistics, Columbia University
New York, NY 10027
jpc2181@columbia.edu

Abstract

We introduce a novel information-theoretic approach for active model selection and demonstrate its effectiveness in a real-world application. Although our method can work with arbitrary models, we focus on actively learning the appropriate structure for Gaussian process (GP) models with arbitrary observation likelihoods. We then apply this framework to rapid screening for noise-induced hearing loss (NIHL), a widespread and preventible disability, if diagnosed early. We construct a GP model for pure-tone audiometric responses of patients with NIHL. Using this and a previously published model for healthy responses, the proposed method is shown to be capable of diagnosing the presence or absence of NIHL with drastically fewer samples than existing approaches. Further, the method is extremely fast and enables the diagnosis to be performed in real time.

1 Introduction

Personalized medicine has long been a critical application area for machine learning [1–3], in which automated decision making and diagnosis are key components. Beyond improving quality of life, machine learning in diagnostic settings is particularly important because collecting additional medical data often incurs significant financial burden, time cost, and patient discomfort. In machine learning one often considers this problem to be one of active feature selection: acquiring each new feature (e.g., a blood test) incurs some cost, but will, with hope, better inform diagnosis, treatment, and prognosis. By careful analysis, we may optimize this trade off.

However, many diagnostic settings in medicine do not involve feature selection, but rather involve querying a sample space to discriminate different models describing patient attributes. A particular, clarifying example that motivates this work is *noise-induced hearing loss* (NIHL), a prevalent disorder affecting 26 million working-age adults in the United States alone [4] and affecting over half of workers in particular occupations such as mining and construction. Most tragically, NIHL is entirely preventable with simple, low-cost solutions (e.g., earplugs). The critical requirement for prevention is effective early diagnosis.

To be tested for NIHL, patients must complete a time-consuming audiometric exam that presents a series of tones at various frequencies and intensities; at each tone the patient indicates whether he/she hears the tone [5–7]. From the responses, the clinician infers the patient's audible threshold on a set of discrete frequencies (the *audiogram*); this process requires the delivery of up to hundreds of tones. Audiologists scan the audiogram for a hearing deficit with a characteristic *notch* shape—a

Extension: NIPS 2016

Another extension to *model selection* with *fixed datasets,* one step closer to fully automated Bayesian optimization!

Bayesian optimization for automated model selection, NIPS 2016.

Bayesian optimization for automated model selection

Gustavo Malkomes [†], **Chip Schaff** [†], **Roman Garnett**
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO 63130
{luizgustavo, cbschaff, garnett}@wustl.edu

Abstract

Despite the success of kernel-based nonparametric methods, kernel selection still requires considerable expertise, and is often described as a "black art." We present a sophisticated method for automatically searching for an appropriate kernel from an infinite space of potential choices. Previous efforts in this direction have focused on traversing a kernel grammar, only examining the data via computation of marginal likelihood. Our proposed search method is based on Bayesian optimization in model space, where we reason about model evidence as a function to be maximized. We explicitly reason about the data distribution and how it induces similarity between potential model choices in terms of the explanations they can offer for observed data. In this light, we construct a novel kernel between models to explain a given dataset. Our method is capable of finding a model that explains a given dataset well without any human assistance, often with fewer computations of model evidence than previous approaches, a claim we demonstrate empirically.

1 Introduction

Over the past decades, enormous human effort has been devoted to machine learning; preprocessing data, model selection, and hyperparameter optimization are some examples of critical and often expert-dependent tasks. The complexity of these tasks has in some cases relegated them to the realm of "black art." In kernel methods in particular, the selection of an appropriate kernel to explain a given dataset is critical to success in terms of the fidelity of predictions, but the vast space of potential kernels renders the problem nontrivial. We consider the problem of automatically finding an appropriate probabilistic model to explain a given dataset. Although our proposed algorithm is general, we will focus on the case where a model can be completely specified by a kernel, as is the case for example for centered Gaussian processes (GPs).

Recent work has begun to tackle the kernel-selection problem in a systematic way. Duvenaud et al. [1] and Grosse et al. [2] described generative grammars for enumerating a countably infinite space of arbitrarily complex kernels via exploiting the closure of kernels under additive and multiplicative composition. We adopt this kernel grammar in this work as well. Given a dataset, Duvenaud et al. [1] proposed searching this infinite space of models using a greedy search mechanism. Beginning at the root of the grammar, we traverse the tree greedily attempting to maximize the (approximate) evidence for the data given by a GP model incorporating the kernel.

In this work, we develop a more sophisticated mechanism for searching through this space. The greedy search described above only considers a given dataset by querying a model's evidence. Our search performs a *metalearning* procedure, which, conditional on a dataset, establishes similarities among the models in terms of the space of explanations they can offer for the data. With this viewpoint, we construct a novel kernel between models (a "kernel kernel"). We then approach

[†]These authors contributed equally to this work

THANK YOU!

Questions?