Entropy Search for Information-Efficient Global Optimization

Philipp Hennig PHENNIG@TUEBINGEN.MPG.DE
Christian J. Schuler CSCHULER@TUEBINGEN.MPG.DE
Department of Empirical Inference
Max Planck Institute for Intelligent Systems
Spemannstraße 38, Tübingen, Germany

Abstract

Contemporary global optimization algorithms are based on local measures of utility, rather than a probability measure over location and value of the optimum. They thus attempt to collect low function values, not to learn about the optimum. The reason for the absence of probabilistic global optimizers is that the corresponding inference problem is intractable in several ways. This paper develops desiderata for probabilistic optimization algorithms, then presents a concrete algorithm which addresses each of the computational intractabilities with a sequence of approximations and explicitly adresses the decision problem of maximizing information gain from each evaluation.

Keywords: Optimization, Probability, Information, Gaussian Processes, Expectation Propagation

1. Introduction

Optimization problems are ubiquitous in science, engineering, and economics. Over time the requirements of many separate fields have led to a heterogenous set of settings and algorithms. Speaking very broadly, however, there are two distinct regimes for optimization. In the first one, relatively cheap function evaluations take place on a numerical machine and the goal is to find a "good" region of low or high function values. Noise tends to be small or negligible, and derivative observations are often available at low additional cost; but the parameter space may be very high-dimensional. This is the regime of *numerical, local* or *convex* optimization, often encountered as a sub-problem of machine learning algorithms. Popular algorithms for such settings include quasi-Newton methods (Broyden et al., 1965; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), the conjugate gradient method (Hestenes and Stiefel, 1952), and stochastic optimization and evolutionary search methods (e.g. Hansen and Ostermeier (2001)), to name only a few. Since these algorithms perform local search, constraints on the solution space are often a crucial part of the problem. Thorough introductions can be found in the textbooks by Nocedal and Wright (1999) and Boyd and Vandenberghe (2004). This paper will utilize algorithms from this domain, but it is not its primary subject.

In the second milieu, which this paper addresses, the function itself is not known and needs to be learned during the search for its *global* minimum within some measurable (usually: bounded) domain. Here, the parameter space is often relatively low-dimensional, but evaluating the function involves a monetarily or morally expensive physical process – building a prototype, drilling a borehole, killing a rodent, treating a patient. Noise is often a nontrivial issue, and derivative observations, while potentially available, cannot be expected in general. While algorithms for such applications need to be tractable, their most important desideratum is efficient use of data, rather than raw computational cost. This domain is often called *global optimization*, but is also closely associated with the field of *experimental design* and related to the concept of *exploration* in reinforcement learning. The learned model of the function is also known as a *response surface* in some communities. The two contributions of this paper are a probabilistic view on this field, and a concrete algorithm for such problems.

1.1 Problem Definition

We define the problem of *probabilistic global optimization*: Let $I \subset \mathbb{R}^D$ be some bounded domain of the real vector space. There is a function $f : I \mathrel{-\!\!\triangleright} \mathbb{R}$, and our knowledge about f is described by a probability measure $p(f)$ over the space of functions $I \mathrel{-\!\!\triangleright} \mathbb{R}$. This induces a measure

$$p_{\min}(x) \equiv p[x = \arg\min f(x)] = \int_{f: I \mathrel{-\!\!\triangleright} \mathbb{R}} p(f) \prod_{\substack{\tilde{x} \in I \\ \tilde{x} \neq x}} \theta[f(\tilde{x}) - f(x)] \, \mathrm{d}f \tag{1}$$

were θ is Heaviside's step function. The exact meaning of the "infinite product" over the entire domain I in this equation should be intuitively clear, but is defined properly in the Appendix. Note that the integral is over the infinite-dimensional space of functions. We assume we can evaluate the function[1] at any point $x \in I$ within some bounded domain I, obtaining function values $y(x)$ corrupted by noise, as described by a likelihood $p(y \,|\, f(x))$. Finally, let $L(x^*, x_{\min})$ be a loss function describing the cost of naming x^* as the result of optimization if the true minimum is at $x_{\min}$. This loss function induces a loss functional $\mathcal{L}(p_{\min})$ assigning utility to the uncertain knowledge about $x_{\min}$, as

$$\mathcal{L}(p_{\min}) = \int_I [\min_{x^*} L(x^*, x_{\min})] p_{\min}(x_{\min}) \, \mathrm{d}x_{\min}. \tag{2}$$

The goal of global optimization is to decrease the expected loss after H function evaluations at locations $\boldsymbol{x} = \{x_1, \dots, x_H\} \subset I$. The expected loss is

$$\langle \mathcal{L} \rangle_H = \int p(\boldsymbol{y} \,|\, \boldsymbol{x}) \mathcal{L}(p_{\min}(x \,|\, \boldsymbol{y}, \boldsymbol{x})) \, \mathrm{d}\boldsymbol{y} = \iint p(\boldsymbol{y} \,|\, \boldsymbol{f}(\boldsymbol{x})) p(\boldsymbol{f}(\boldsymbol{x}) \,|\, \boldsymbol{x}) \mathcal{L}(p_{\min}(x \,|\, \boldsymbol{y}, \boldsymbol{x})) \, \mathrm{d}\boldsymbol{y} \, \mathrm{d}\boldsymbol{f} \tag{3}$$

where $\mathcal{L}(p_{\min}(x \,|\, \boldsymbol{y}, \boldsymbol{x}))$ should be understood as the cost assigned to the measure $p_{\min}(x)$ induced by the posterior belief over f after observations $\boldsymbol{y} = \{y_1, \dots, y_H\} \subset \mathbb{R}$ at the locations $\boldsymbol{x}$.

The remainder of this paper will replace the symbolic objects in this general definition with concrete measures and models to construct an algorithm we call *Entropy Search*. But it is useful to pause at this point to contrast this definition with other concepts of optimization.

[1] We may further consider observations of linear operations on f. This includes derivative and integral observations of any order, if they exist. Section 2.8.1 addresses this point; it is unproblematic under our chosen prior, but clutters the notation, and is thus left out elsewhere in the paper.

Probabilistic Optimization The distinctive aspect of our definition of "optimization" is Equation (1), an explicit role for the function's extremum. Previous work did not consider the extremum so directly. In fact, many frameworks do not even use a measure over the function itself. An example of optimizers that only implicitly encode assumptions about the function are genetic algorithms (Schmitt, 2004) and evolutionary search (Hansen and Ostermeier, 2001). If such formulations feature the global minimum $x_{\min}$ at all, then only in statements about the limit behavior of the algorithm after many evaluations. Not explicitly writing out the prior over the function space can have advantages: Probabilistic analyses tend to involve intractable integrals; a less explicit formulation thus allows to construct algorithms with interesting properties that would be entirely intractable from a probabilistic viewpoint. But non-probabilistic algorithms cannot make explicit statements about the location of the minimum. At best, they may be able to provide bounds.

Fundamentally, reasoning about optimization of functions on continuous domains *after finitely many evaluations*, like any other inference task on spaces without natural measures, is impossible without prior assumptions. For intuition, consider the following thought experiment: Let $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ be a finite, possibly empty, set of previously collected data. For simplicity, and without loss of generality, assume there was no measurement noise, so the true function actually passes through each data point. Say we want to suggest that the minimum of f may be at $x^* \in I$. To make this argument, we propose a number of functions that pass through $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ and are minimized at x^*. We may even suggest an uncountably infinite set of such functions. Whatever our proposal, a critic can always suggest another uncountable set of functions that also pass through the data, and are *not* minimized at x^*. To argue with this person, we need to reason about the relative size of our set versus their set. Assigning size to infinite sets amounts to the aforementioned normalized measure over admissible functions $p(f)$, and the consistent way to reason with such measures is probability theory (Kolmogorov, 1933; Cox, 1946). Of course, this amounts to imposing assumptions on f, but this is a fundamental epistemological limitation of inference, not a special aspect of optimization.

Relationship to the Bandit Setting There is a considerable amount of prior work on continuous bandit problems, also sometimes called "global optimization" (e.g. Kleinberg, 2005; Grünewälder et al., 2010; Srinivas et al., 2010). The bandit concept differs from the setting defined above, and bandit regret bounds do not apply here: Bandit algorithms seek to minimize *regret*, the sum over function values at evaluation points, while probabilistic optimizers seek to infer the minimum, no matter what the function values at evaluation points. An optimizer gets to evaluate H times, then has to make one single decision regarding $\mathcal{L}(p_{\min})$. Bandit players have to make H evaluations, such that the evaluations produce low values. This forces bandits to focus their evaluation policy on function value, rather than the loss at the horizon (see also Section 3.1). In probabilistic optimization, the only quantity that counts is the quality of the belief on $p_{\min}$ under $\mathcal{L}$, after H evaluations, not the sum of the function values returned during those H steps.

Relationship to Heuristic Gaussian Process Optimization and Response Surface Optimization There are also a number of works employing Gaussian process measures to construct heuristics for search, also known as "Gaussian process global optimization" (Jones et al., 1998; Lizotte, 2008; Osborne et al., 2009). As in our definition, these methods explicitly infer the function from observations, constructing a Gaussian process posterior. But they then evaluate at the location maximizing a heuristic $u[p(f(x))]$ that turns the *marginal* belief over $f(x)$ at x, which is a univariate Gaussian $p(f(x)) = \mathcal{N}[f(x); \mu(x), \sigma^2(x)]$, into an ad hoc utility for evaluation, designed to have high value at locations close to the function's minimum. Two popular heuristics are the *probability of improvement* (Lizotte, 2008)

$$u_{\mathrm{PI}}(x) = p[f(x) < \eta] = \int_{-\infty}^{\eta} \mathcal{N}(f(x); \mu(x), \sigma(x)^2)\,\mathrm{d}f(x) = \Phi\left(\frac{\eta - \mu(x)}{\sigma(x)}\right) \tag{4}$$

and *expected improvement* (Jones et al., 1998)

$$u_{\mathrm{EI}}(x) = \mathsf{E}[\min\{0, (\eta - f(x))\}] = (\eta - \mu)\Phi\left(\frac{\eta - \mu(x)}{\sigma(x)}\right) + \sigma\phi\left(\frac{\eta - \mu(x)}{\sigma(x)}\right) \tag{5}$$

where $\Phi(z) = 1/2[1 + \mathrm{erf}(z/\sqrt{2})]$ is the standard Gaussian cumulative density function, $\phi(x) = \mathcal{N}(x; 0, 1)$ is the standard Gaussian probability density function, and η is a current "best guess" for a low function value, e.g. the lowest evaluation so far.

These two heuristics have different units of measure: probability of improvement is a probability, expected improvement has the units of f. Both utilities differ markedly from Eq. (1), $p_{\min}$, which is a probability *measure* and as such a *global* quantity. See Figure 2 for a comparison of the three concepts on an example. The advantage of the heuristic approach is that it is computationally lightweight, because the utilities have analytic form. But local measures cannot capture general decision problems of the type described above. For example, these algorithms do not capture the effect of evaluations on knowledge: A small region of high density $p_{\min}(x)$ may be less interesting to explore than a broad region of lower density, because the expected *change* in knowledge from an evaluation in the broader region may be much larger, and may thus have much stronger effect on the loss. If the goal is to infer the location of the minimum (more generally: minimize loss at the horizon), the optimal strategy is to evaluate where we expect to *learn* most about the minimum (reduce loss toward the horizon), rather then where we think the minimum *is* (recall Section 1.1). The former is a nonlocal problem, because evaluations affect the belief, in general, everywhere. The latter is a local problem.

2. Entropy Search

The probable reason for the absence of global optimization algorithms from the literature is a number of intractabilities in any concrete realisation of the setting of Section 1.1. This section makes some choices and constructs a series of approximations, to arrive at a tangible algorithm, which we call *Entropy Search.* The derivations evolve along the following path.

choosing $p(f)$ We commit to a Gaussian process prior on f (Section 2.1). Limitations and implications of this choice are outlined, and possible extensions suggested, in Sections 2.8.1 and 2.8.3.

discretizing $p_{\min}$ We discretize the problem of calculating $p_{\min}$, to a finite set of representer points chosen from a non-uniform measure, which deals gracefully with the curse of dimensionality. Artefacts created by this discretization are studied in the tractable one-dimensional setting (Section 2.2).

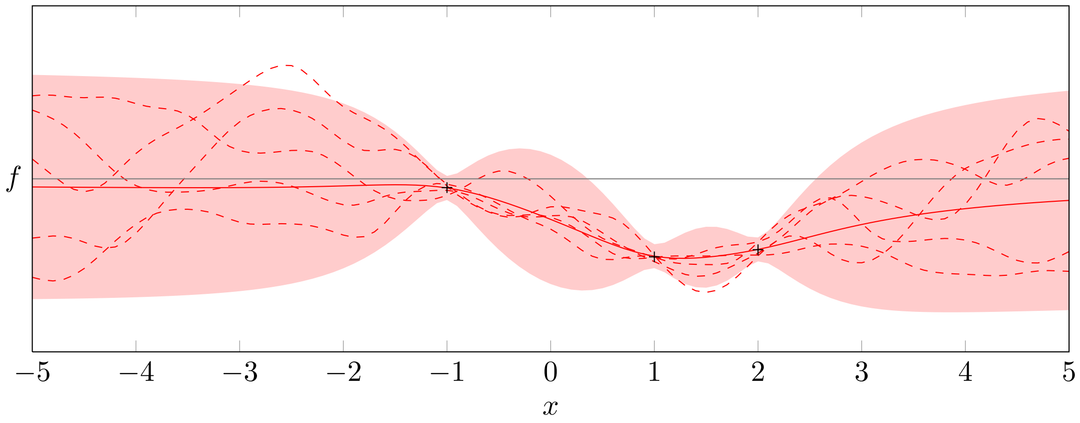

Figure 1: A Gaussian process measure (rational quadratic kernel), conditioned on three previous observations (black crosses). Mean function in solid red, marginal standard deviation at each location (two standard deviations) as light red tube. Five sampled functions from the current belief as dashed red lines. Arbitrary ordinate scale, zero in gray.

approximating $p_{\min}$ We construct an efficient approximation to $p_{\min}$, which is required because Eq. (1), even for finite-dimensional Gaussian measures, is not analytically tractable, (Section 2.3). We compare the approximation to the (asymptotically exact, but more expensive) Monte Carlo solution.

predicting change to $p_{\min}$ The Gaussian process measure affords a straightforward but rarely used analytic probabilistic formulation for the *change* of $p(f)$ as a function of the next evaluation point (Section 2.4).

choosing loss function We commit to relative *entropy* from a uniform distribution as the loss function, as this can be interpreted as a utility on gained *information* about the location of the minimum (Section 2.5).

predicting expected information gain From the predicted change, we construct a first-order expansion on $\langle \mathcal{L} \rangle$ from future evaluations and, again, compare to the asymptotically exact Monte Carlo answer (Section 2.6).

choosing greedily Faced with the exponential cost of the exact dynamic problem to the horizon H, we accept a greedy approach for the reduction of $\langle \mathcal{L} \rangle$ at every step. We illustrate the effect of this shortcut in an example setting (Section 2.7).

2.1 Gaussian Process Measure on f

The remainder of the paper commits to Gaussian process measures for $p(f)$. These are convenient for the task at hand due to their descriptive generality and their convenient analytic properties. Since this paper is aimed at readers from several communities, this section contains a very brief introduction to some relevant aspects of Gaussian processes;

readers familiar with the subject can safely skip ahead. A thorough introduction can be found in a textbook of Rasmussen and Williams (2006). Some readers from other fields may find it helpful to know that more or less special cases of Gaussian process inference are elsewhere known under names like *Kriging* (Krige, 1951) and *Kolmogorov-Wiener prediction* (Wiener and Masani, 1957), but while these frameworks are essentially the same idea, the generality of their definitions varies, so restrictions of those frameworks should not be assumed to carry over to Gaussian process inference as understood in machine learning.

A Gaussian process is an infinite-dimensional probability density, such that each linear finite-dimensional restriction is multivariate Gaussian. The infinite-dimensional space can be thought of as a space of functions, and the finite-dimensional restrictions as *values* of those functions at locations $\{x_i^*\}_{i=1,\dots,N}$. Gaussian process beliefs are parametrized by a *mean function* $m : I \to \mathbb{R}$ and a *covariance function* $k : I \times I \to \mathbb{R}$. For our particular analysis, we restrict the domain I to finite, compact subsets of the real vector spaces $\mathbb{R}^D$. The covariance function, also known as the *kernel*, has to be positive definite, in the sense that any finite-dimensional matrix with elements $K_{ij} = k(x_i, x_j)$ has to be positive definite $\forall x_i, x_j \in I$. A number of such kernel functions are known in the literature, and different kernel functions induce different kinds of Gaussian process measures over the space of functions. Among the most widely used kernels for regression are the *squared exponential* kernel

$$k_{\mathrm{SE}}(x, x'; \boldsymbol{S}, s) = s^2 \exp\left[-\frac{1}{2}(x - x')^\intercal \boldsymbol{S}^{-1}(x - x')\right] \tag{6}$$

which induces a measure that puts nonzero mass on only smooth functions of *characteristic length-scale* $\boldsymbol{S}$ and *signal variance* s^2 (MacKay, 1998b), and the *rational quadratic* kernel (Matérn, 1960; Rasmussen and Williams, 2006)

$$k_{\mathrm{RQ}}(x, x'; \boldsymbol{S}, s, \alpha) = s^2 \left(1 + \frac{1}{2\alpha}(x - x')^\intercal \boldsymbol{S}^{-1}(x - x')\right)^{-\alpha} \tag{7}$$

which induces a belief over smooth functions whose characteristic length scales are a scale mixture over a distribution of width $1/\alpha$ and location $\boldsymbol{S}$. Other kernels can be used to induce beliefs over non-smooth functions (Matérn, 1960), and even over non-continuous functions (Uhlenbeck and Ornstein, 1930). Experiments in this paper use the two kernels defined above, but the results apply to all kernels inducing beliefs over *continuous* functions. While there is a straightforward relationship between kernel continuity and the mean square continuity of the induced *process*, the relationship between the kernel function and the continuity of each *sample* is considerably more involved (Adler, 1981, §3). Regularity of the kernel also plays a nontrivial role in the question wether the distribution of infima of samples from the process is well-defined at all (Adler, 1990). In this work, we side-step this issue by assuming that the chosen kernel is sufficiently regular to induce a well-defined belief $p_{\min}$ as defined by Equation (26).

Kernels form a semiring: products and sums of kernels are kernels. These operations can be used to generalize the induced beliefs over the function space (Section 2.8.3). Without loss of generality, the mean function is often set to $m \equiv 0$ in theoretical analyses, and this paper will keep with this tradition, except for Section 2.8.3. Where m is nonzero, its effect is a straightforward off-set $p(f(x)) \to p(f(x) - m(x))$.

For the purpose of regression, the most important aspect of Gaussian process priors is that they are conjugate to the likelihood from finitely many observations $(\boldsymbol{X}, \boldsymbol{Y}) = \{\boldsymbol{x}_i, y_i\}_{i=1,\dots,N}$ of the form $y_i(\boldsymbol{x}_i) = f(\boldsymbol{x}_i) + \xi$ with Gaussian noise $\xi \sim \mathcal{N}(0, \sigma^2)$. The posterior is a Gaussian process with mean and covariance functions

$$\mu(\boldsymbol{x}^*) = k_{\boldsymbol{x}^*,\boldsymbol{X}}[K_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I}]^{-1}\boldsymbol{y} \quad ; \quad \Sigma(\boldsymbol{x}^*, \boldsymbol{x}_*) = k_{\boldsymbol{x}^*,\boldsymbol{x}_*} - k_{\boldsymbol{x}^*,\boldsymbol{X}}[K_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I}]^{-1} k_{\boldsymbol{X},\boldsymbol{x}_*} \tag{8}$$

where $K_{\boldsymbol{X},\boldsymbol{X}}$ is the kernel Gram matrix $K^{(i,j)}_{\boldsymbol{X},\boldsymbol{X}} = k(x_i, x_j)$, and other objects of the form $k_{\boldsymbol{a},\boldsymbol{b}}$ are also matrices with elements $k^{(i,j)}_{\boldsymbol{a},\boldsymbol{b}} = k(\boldsymbol{a}_i, \boldsymbol{b}_j)$. Finally, for what follows it is important to know that it is straightforward to sample "functions" (point-sets of arbitrary size from I) from a Gaussian process. To sample the value of a particular sample at the M locations $\boldsymbol{X}^*$, evaluate mean and variance function as a function of any previously collected datapoints, using Eq. (8), draw a vector $\boldsymbol{\zeta} \sim \prod^M \mathcal{N}(0, 1)$ of M random numbers i.i.d. from a standard one-dimensional Gaussian distribution, then evaluate

$$\tilde{f}(\boldsymbol{X}^*) = \mu(\boldsymbol{X}^*) + \mathsf{C}[\Sigma(\boldsymbol{X}^*, \boldsymbol{X}^*)]^\intercal \boldsymbol{\zeta} \tag{9}$$

where the operator C denotes the Cholesky decomposition (Benoit, 1924).

2.2 Discrete Representations for Continuous Distributions

Having established a probability measure $p(f)$ on the function, we turn to constructing the belief $p_{\min}(x)$ over its minimum. Inspecting Equation (1), it becomes apparent that it is challenging in two ways: First, because it is an integral over an infinite-dimensional space, and second, because even on a finite-dimensional space it may be a hard integral for a particular $p(f)$. This section deals with the former issue, the following Section 2.3 with the latter.

It may seem daunting that $p_{\min}$ involves an infinite-dimensional integral. The crucial observation for a meaningful approximation in finite time is that regular functions can be represented meaningfully on finitely many points. If the stochastic process representing the belief over f is sufficiently regular, then Equation (1) can be approximated arbitrarily well with finitely many representer points. The discretization grid need not be regular – it may be sampled from any distribution which puts non-zero measure on every open neighborhood of I. This latter point is central to a graceful handling of the curse of dimensionality: The naïve approach of approximately solving Equation (1) on a regular grid, in a D-dimensional domain, would require $\mathcal{O}(\exp(D))$ points to achieve any given resolution. This is obviously not efficient: Just like in other numerical quadrature problems, any given resolution can be achieved with fewer representer points if they are chosen irregularly, with higher resolution in regions of greater influence on the result of integration. We thus choose to *sample* representer points from a proposal measure u, using a Markov chain Monte Carlo sampler (our implementation uses shrinking rank slice sampling (Thompson and Neal, 2010)).

What is the effect of this stochastic discretization? A non-uniform quadrature measure $u(\tilde{x})$ for N representer locations $\{\tilde{x}_i\}_{i=1,\dots,N}$ leads to varying widths in the "steps" of the representing staircase function. As $N \rightharpoonup \infty$, the width of each step is approximately proportional to $(u(\tilde{x}_i)N)^{-1}$. Section 2.3 will construct a discretized $\hat{q}_{\min}(\tilde{x}_i)$ that is an approximation to the probability that $f_{\min}$ occurs within the step at $\tilde{x}_i$. So the approximate

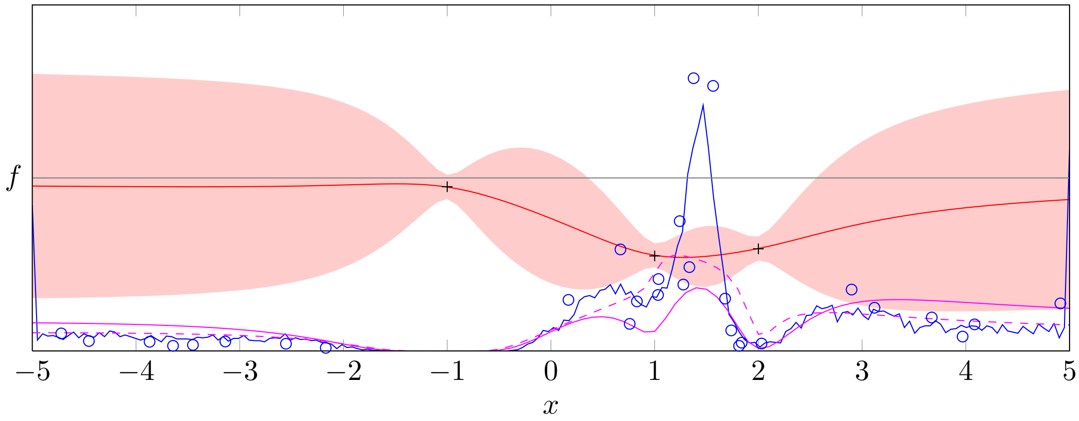

Figure 2: $p_{\min}$ induced by $p(f)$ from Figure 1. $p(f)$ repeated for reference. Blue solid line: Asymptotically exact representation gained from exact sampling of functions on a regular grid. For comparison, the plot also shows the local utilities *probability of improvement* (dashed magenta) and *expected improvement* (solid magenta) often used for Gaussian process global optimization. Blue circles: Approximate representation on representer points, sampled from probability of improvement measure. Stochastic error on sampled values, due to only asymptotically correct assignment of mass to samples, and varying density of points, focusing on relevant areas of $p_{\min}$. This plot uses arbitrary scales for each object: The two heuristics have different units of measure, differing from that of $p_{\min}$. Notice the interesting features of $p_{\min}$ at the boundaries of the domain: The prior belief encodes that f is smooth, and puts finite probability mass on the hypothesis that f has negative (positive) derivative at the right (left) boundary of the domain. With nonzero probability, the minimum thus lies exactly on the boundary of the domain, rather than within a Taylor radius of it.

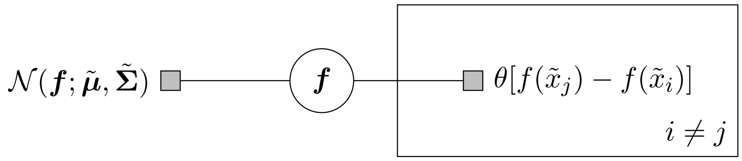
$\mathcal{N}(\boldsymbol{f}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$
$\boldsymbol{f}$
$\theta[f(\tilde{x}_j) - f(\tilde{x}_i)]$
$i \neq j$

Figure 3: Graphical model providing motivation for EP approximation on $p_{\min}$. See text for details.

$\hat{p}_{\min}$ on this step is proportional to $\hat{q}_{\min}(\tilde{x}_i)u(\tilde{x}_i)$, and can be easily normalized numerically, to become an approximation to $p_{\min}$.

How should the measure u be chosen? Unfortunately, the result of the integration, being a density rather than a function, is itself a function of u, and the loss-function is also part of the problem. So it is nontrivial to construct an optimal quadrature measure. Intuitively, a good proposal measure for discretization points should put high resolution on regions of I where the shape of $p_{\min}$ has strong influence on the loss, and on its change. For our choice of loss function (Section 2.5), it is a good idea to choose u such that it puts high mass on regions of high value for $p_{\min}$. But for other functions, this need not always be the case.

We have experimented with a number of ad-hoc choices for u, and found the aforementioned "expected improvement" and "probability of improvement" (Section 1.1) to lead to reasonably good performance. We use these functions for a similar reason as their original authors: Because they *tend* to have high value in regions where $p_{\min}$ is also large. To avoid confusion, however, note that we use these functions as unnormalized measures to *sample discretization points* for our *calculation* of $p_{\min}$, not as an approximation for $p_{\min}$ itself, as was done in previous work by other authors. Defects in these heuristics have weaker effect on our algorithm than in the cited works: In our case, if u is not a good proposal measure, we simply need more samples to construct a good representation of $p_{\min}$. In the limit of $N \rightarrow \infty$, all choices of u perform equally well, as long as they put nonzero mass on all open neighborhoods of the domain.

2.3 Approximating $p_{\min}$ with Expectation Propagation

The previous Section 2.2 provided a way to construct a non-uniform grid of N discrete locations $\tilde{x}_i$, $i = 1, \dots, N$. The restriction of the Gaussian process belief to these locations is a multivariate Gaussian density with mean $\tilde{\boldsymbol{\mu}} \in \mathbb{R}^N$ and covariance $\tilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{N \times N}$. So Equation (1) reduces to a discrete probability *distribution* (as opposed to a density)

$$\hat{p}_{\min}(x_i) = \int_{\boldsymbol{f} \in \mathbb{R}^N} \mathcal{N}(\boldsymbol{f}; \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}) \prod_{i \neq j}^{N} \theta(f(x_j) - f(x_i)) \, \mathrm{d}\boldsymbol{f}. \tag{10}$$

This is a multivariate Gaussian integral over a half-open, convex, piecewise linearly constrained integration region – a polyhedral cone. Unfortunately, such integrals are known to be intractable (Plackett, 1954; Lazard-Holly and Holly, 2003). However, it is possible to construct an effective approximation $\hat{q}_{\min}$ based on Expectation Propagation (EP) (Minka, 2001): Consider the belief $p(f(\tilde{x}))$ as a "prior message" on $f(\tilde{x})$, and each of the terms in the product as one factor providing another message. This gives the graphical model shown in Figure 3. Running EP on this graph provides an approximate Gaussian marginal, whose normalisation constant $\hat{q}_{\min}(x_i)$, which EP also provides, approximates $p(f \,|\, x_{\min} = x_i)$. The EP algorithm itself is somewhat involved, and there are a number of algorithmic technicalities to take into account for this particular setting. We refer interested readers to recent work by Cunningham et al. (2011), which gives a detailed description of these aspects. The cited work also establishes that, while EP's approximations to Gaussian integrals are not always reliable, in this particular case, where there are as many constraints as dimensions to the problem, the approximation is generally of high quality (see Figure 4 for an example). An important advantage of the EP approximation over both numerical integration and Monte Carlo integration (see next Section) is that it allows analytic differentiation of $\hat{q}_{\min}$ with respect to the parameters $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\Sigma}}$ (Cunningham et al., 2011; Seeger, 2008). This fact will become important in Section 2.6.

The computational cost of this approximation is considerable: Each computation of $\hat{q}_{\min}(\tilde{x}_i)$, for a given i, involves N factor updates, which each have rank 1 and thus cost $\mathcal{O}(N^2)$. So, overall, the cost of calculating $\hat{q}_{\min}(\tilde{\boldsymbol{x}})$ is $\mathcal{O}(N^4)$. This means N is effectively limited to well below $N = 1000$. Our implementation uses a default of $N = 50$, and can calculate next evaluation points in ~ 10 seconds. Once again, it is clear that this algorithm is not suitable for simple numerical optimization problems; but a few seconds are arguably an acceptable waiting time for physical optimization problems.

2.3.1 An alternative: Sampling

An alternative to EP is Monte Carlo integration: sample S functions exactly from the Gaussian belief on $p(f)$, at cost $O(N^2)$ per sample, then find the minimum for each sample in $\mathcal{O}(N)$ time. This technique was used to generate the asymptotically exact plots in Figures 2 and following. It has overall cost $\mathcal{O}(SN^3)$, and can be implemented efficiently using Matrix-Matrix multiplications, so each evaluation of this algorithm is considerably faster than EP. It also has the advantage of asymptotic exactness. But, unfortunately, it provides no analytic derivatives, because of strong discontinuity in the step functions of Eq. (1). So the choice is between a first-order expansion using EP (see Section 2.6) which is expensive, but provides a re-usable, differentiable function, and repeated calls to a cheaper, asymptotically exact sampler. In our experiments, the former option appeared to be considerably faster, and of acceptable approximative quality. But for relatively high-dimensional optimization problems, where one would expect to require relatively large N for acceptable discretization, the sampling approach can be expected to scale better. The code we plan to publish upon acceptance offers a choice between these two approaches.

2.4 Predicting Innovation from Future Observations

As detailed in Equation (3), the optimal choice of the next H evaluations is such that the *expected* change in the loss $\langle \mathcal{L} \rangle_{\boldsymbol{x}}$ is minimal, i.e. effects the biggest possible expected drop in loss. The loss is a function of $p_{\min}$, which in turn is a function of $p(f)$. So predicting change in loss requires predicting change in $p(f)$ as a function of the next evaluation points. It is another convenient aspect of Gaussian processes that they allow such predictions in analytic form (Hennig, 2011): Let previous observations at $\boldsymbol{X}_0$ have yielded observations

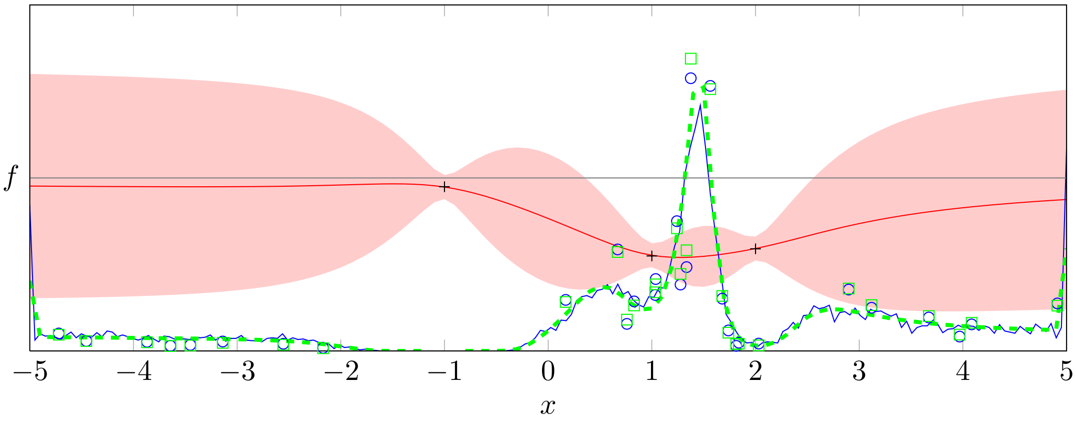

Figure 4: EP-approximation to $p_{\min}$ (green). Other plots as in previous figures. EP achieves good agreement with the asymptotically exact Monte Carlo approximation to $p_{\min}$, including the point masses at the boundaries of the domain.

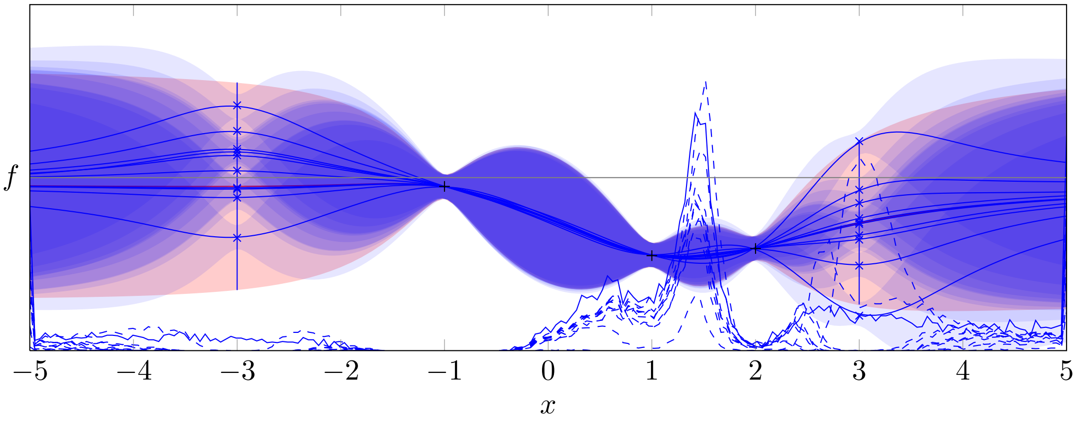

Figure 5: Innovation from two observations at $x = -3$ and $x = 3$. Current belief in red, as in Figure 1. Samples from the belief over possible beliefs after observations at $\boldsymbol{x}$ in blue. For each sampled innovation, the plot also shows the induced innovated $p_{\min}$ (lower sampling resolution as previous plots). Innovations from several (here: two) observations can be sampled jointly.

$\boldsymbol{Y}_0$. Evaluating at locations $\boldsymbol{X}$ will give new observations $\boldsymbol{Y}$, and the mean will be given by

$$\begin{aligned}\mu(x^*) &= [k_{x^*,\boldsymbol{X}_0}, k_{x^*,\boldsymbol{X}}] \begin{pmatrix} K_{\boldsymbol{X}_0,\boldsymbol{X}_0} & k_{\boldsymbol{X}_0,\boldsymbol{X}} \\ k_{\boldsymbol{X},\boldsymbol{X}_0} & K_{\boldsymbol{X},\boldsymbol{X}} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{Y}_0 \\ \boldsymbol{Y} \end{pmatrix} \\ &= k_{x^*,\boldsymbol{X}_0} K^{-1}_{\boldsymbol{X}_0,\boldsymbol{X}_0} \boldsymbol{Y}_0 + (k_{x^*,\boldsymbol{X}} - k_{x^*,\boldsymbol{X}_0} K^{-1}_{\boldsymbol{X}_0,\boldsymbol{X}_0} k_{\boldsymbol{X}_0,\boldsymbol{X}}) \times \\ &\qquad (k_{\boldsymbol{X},\boldsymbol{X}} - k_{\boldsymbol{X},\boldsymbol{X}_0} K^{-1}_{\boldsymbol{X}_0,\boldsymbol{X}_0} k_{\boldsymbol{X}_0,\boldsymbol{X}})^{-1} (\boldsymbol{Y} - k_{\boldsymbol{X},\boldsymbol{X}_0} K^{-1}_{\boldsymbol{X}_0,\boldsymbol{X}_0} \boldsymbol{Y}_0) \\ &= \mu_0(x^*) + \Sigma_0(x^*, \boldsymbol{X}) \Sigma_0^{-1}(\boldsymbol{X}, \boldsymbol{X})(\boldsymbol{Y} - \mu_0(\boldsymbol{X}))\end{aligned} \tag{11}$$

where $K^{(i,j)}_{a,b} = k(a_i, b_j) + \delta_{ij}\sigma^2$. The step from the first to the second line involves an application of the matrix inversion lemma, the last line uses the mean and covariance functions conditioned on the dataset $(\boldsymbol{X}_0, \boldsymbol{Y}_0)$ so far. Since $\boldsymbol{Y}$ is presumed to come from this very Gaussian process belief, we can write

$$\boldsymbol{Y} = \mu(\boldsymbol{X}) + \mathsf{C}[\Sigma(\boldsymbol{X},\boldsymbol{X})]^\intercal \boldsymbol{\Omega}' + \sigma\boldsymbol{\omega} = \mu(\boldsymbol{X}) + \mathsf{C}[\Sigma(\boldsymbol{X},\boldsymbol{X}) + \sigma^2 \boldsymbol{I}_H]^\intercal \boldsymbol{\Omega} \qquad \boldsymbol{\Omega}, \boldsymbol{\Omega}', \boldsymbol{\omega} \sim \mathcal{N}(0, \boldsymbol{I}_H), \tag{12}$$

and Equation (11) simplifies. An even simpler construction can be made for the covariance function. We find that mean and covariance function of the posterior after observations $(\boldsymbol{X}, \boldsymbol{Y})$ are mean and covariance function of the prior, incremented by the *innovations*

$$\begin{aligned}\Delta\mu_{\boldsymbol{X},\Omega}(x^*) &= \Sigma(x^*, \boldsymbol{X})\Sigma^{-1}(\boldsymbol{X},\boldsymbol{X})\,\mathsf{C}[\Sigma(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\boldsymbol{I}_H]\boldsymbol{\Omega} \\ \Delta\Sigma_{\boldsymbol{X}}(x^*, x_*) &= \Sigma(x^*, \boldsymbol{X})\Sigma^{-1}(\boldsymbol{X},\boldsymbol{X})\Sigma(\boldsymbol{X}, x_*).\end{aligned} \tag{13}$$

The change to the mean function is stochastic, while the change to the covariance function is deterministic. Both innovations are functions both of $\boldsymbol{X}$ and of the evaluation points x^*. One use of this result is to sample $\langle\mathcal{L}\rangle_{\boldsymbol{X}}$ by sampling innovations, then evaluating the innovated $p_{\min}$ for each innovation in an inner loop, as described in Section 2.3.1. An alternative, described in the next section, is to construct an analytic first order approximation to $\langle\mathcal{L}\rangle_{\boldsymbol{X}}$ from the EP prediction constructed in Section 2.3. As mentioned above, the advantage of this latter option is that it provides an analytic function, with derivatives, which allows efficient numerical local optimization.

2.5 Information Gain – the Log Loss

To solve the decision problem of where to evaluate the function next in order to learn most about the location of the minimum, we need to say what it means to "learn". Thus, we require a loss functional that evaluates the information content of innovated beliefs $p_{\min}$. This is, of course, a core idea in information theory. The seminal paper by Shannon (1948) showed that the negative expectation of probability logarithms,

$$\mathsf{H}[\boldsymbol{p}] = -\langle \log p \rangle_{\boldsymbol{p}} = -\sum_i p_i \log p_i \tag{14}$$

known as entropy, has a number of properties that allow its interpretation as a measure of uncertainty represented by a probability distribution p. It's value can be be interpreted as the number of natural information units an optimal compression algorithm requires to encode a sample from the distribution, given knowledge of the distribution. However,

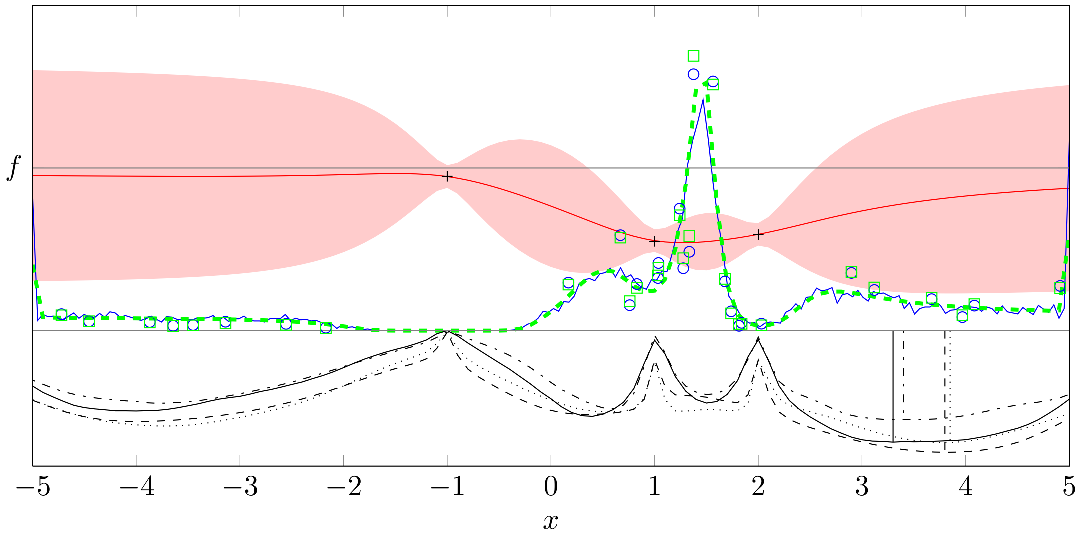

Figure 6: 1-step predicted loss improvement for the log loss (relative entropy). Upper part of plot as before, for reference. Monte Carlo prediction on regular grid as solid black line. Monte Carlo prediction from sampled irregular grid as dot-dashed black line. EP prediction on regular grid as black dashed line. EP prediction from samples as black dashed line. The minima of these functions, where the algorithm will evaluate next, are marked by vertical lines. While the predictions from the various approximations are not identical, they lead to similar next evaluation points. Note that these next evaluation points differ qualitatively from the choice of the GP optimization heuristics of Figure 2. Since each approximation is only tractable up a multiplicative constant, the scales of these plots are arbitrary, and only chosen to overlap for convenience.

it has since been pointed out repeatedly that this concept does not trivially generalize to probability densities. A density $p(x)$ has a unit of measure $[x]^{-1}$, so its logarithm is not well-defined, and one cannot simply replace summation with integration in Equation (14). A functional that *is* well-defined on probability densities and preserves many of the information-content interpretations of entropy (Jaynes and Bretthorst, 2003) is *relative entropy*, also known as Kullback-Leibler divergence (Kullback and Leibler, 1951). We use its negative value as a loss function emphasizing information gain.

$$\mathcal{L}_{\text{KL}}(p; b) = -\int p(x) \log \frac{p(x)}{b(x)} \, \mathrm{d}x \tag{15}$$

As base measure b we choose the uniform measure $\mathbb{U}_I(x) = |I|^{-1}$ over I, which is well-defined because I is presumed to be bounded[2]. With this choice, the loss is maximized (at $\mathcal{L} = 0$) for a uniform belief over the minimum, and diverges toward negative infinity if p approaches a Dirac point distribution. The resulting algorithm, Entropy Search, will thus choose evaluation points such that it expects to move away from the uniform base measure toward a Dirac distribution as quickly as possible.

The reader may wonder: What about the alternative idea of maximizing, at each evaluation, entropy relative to the *current* $p_{\min}$? This would only encourage the algorithm to attempt to change the current belief, but not necessarily in the right direction. For example, if the current belief puts very low mass on a certain region, an evaluation that has even a small chance of increasing $p_{\min}$ in this region could appear more favorable than an alternative evaluation predicted to have a large effect on regions where the current $p_{\min}$ has larger values. The point is not to just change $p_{\min}$, but to change it such that it moves away from the base measure.

Recall that we approximate the *density* $p(x)$ using a *distribution* $\hat{p}(x_i)$ on a finite set $\{x_i\}$ of representer points, which define steps of width proportional, up to stochastic error, to an unnormalized measure $\tilde{u}(x_i)$. In other words, we can approximate $p_{\min}(x)$ as

$$p_{\min}(x) \approx \frac{\hat{p}(x_i) N \tilde{u}(x_i)}{Z_u} \qquad Z_u = \int \tilde{u}(x) \, \mathrm{d}x \qquad x_i = \underset{\{x_j\}}{\arg\min} \|x - x_j\|. \tag{16}$$

We also note that after N samples, the unit element of measure has size, up to stochastic error, of $\Delta x_i \approx \frac{Z_u}{\tilde{u}(x_i) N}$. So we can approximately represent the loss

$$\begin{aligned}
\mathcal{L}_{\text{KL}}(p_{\min}; b) &\approx -\sum_i p_{\min}(x_i) \Delta x_i \log \frac{p_{\min}(x_i)}{b(x_i)} \\
&= -\sum_i \hat{p}_{\min}(x_i) \log \frac{\hat{p}_{\min}(x_i) N \tilde{u}(x_i)}{Z_u b(x_i)} \\
&= -\sum_i \hat{p}_{\min}(x_i) \log \frac{\hat{p}_{\min}(x_i) \tilde{u}(x_i)}{b(x_i)} + \log\left(\frac{Z_u}{N}\right) \sum_i \hat{p}_{\min}(x_i) \\
&= \mathsf{H}[\hat{p}_{\min}] - \langle \log \tilde{u} \rangle_{\hat{p}_{\min}} + \langle \log b \rangle_{\hat{p}_{\min}} + \log Z_u - \log N
\end{aligned} \tag{17}$$

[2] Although uniform measures appeal as a natural representation of ignorance, they do encode an assumption about I being represented in a "natural" way. Under a nonlinear transformation of I, the distribution would not remain uniform. For example, uniform measures on the [0,1] simplex appear bell-shaped in the softmax basis (MacKay, 1998a). So, while b here does not represent prior knowledge on $x_{\min}$ per se, it does provide a unit of measure to information and as such is nontrivial.

which means we do not require the normalization constant Z_u for optimization of $\mathcal{L}_{\mathrm{KL}}$. For our uniform base measure, the third term in the last line is a constant, too; but other base measures would contribute nontrivially.

2.6 First-Order Approximation to $\langle\mathcal{L}\rangle$

Since EP provides analytic derivatives of $p_{\min}$ with respect to mean and covariance of the Gaussian measure over f, we can construct a first order expansion of the expected change in loss from evaluations. To do so, we consider, in turn, the effect of evaluations at $\boldsymbol{X}$ on the measure on f, the induced change in $p_{\min}$, and finally the change in $\mathcal{L}$. Since the change to the mean is Gaussian stochastic, Itō's Lemma (Itō, 1951) applies. The following Equation uses the summation convention: double indices in products are summed over.

$$\begin{aligned}\langle\Delta\mathcal{L}\rangle_{\boldsymbol{X}} = \int \mathcal{L}\Bigg[p^0_{\min} + \frac{\partial p_{\min}}{\partial\Sigma(\tilde{x}_i,\tilde{x}_j)}\Delta\Sigma_{\boldsymbol{X}}(\tilde{x}_i,\tilde{x}_j) + \frac{\partial^2 p_{\min}}{\partial\mu_i\partial\mu_j}\Delta\mu_{\boldsymbol{X},\mathbf{1}}(\tilde{x}_i)\Delta\mu_{\boldsymbol{X},\mathbf{1}}(\tilde{x}_j)\\ + \frac{\partial p_{\min}}{\partial\mu(\tilde{\boldsymbol{x}}_i)}\Delta_{\boldsymbol{X},\Omega}\mu(\tilde{x}_i) + \mathcal{O}((\Delta\mu)^2,(\Delta\Sigma)^2)\Bigg]\mathcal{N}(\Omega;0,1)\,\mathrm{d}\Omega - \mathcal{L}[p^0_{\min}]. \quad (18)\end{aligned}$$

The first line contains deterministic effects, the first term in the second line covers the stochastic aspect. Monte Carlo integration over the stochastic effects can be performed approximately using a small number of samples $\boldsymbol{\Omega}$. These samples should be drawn only once, at first calculation, to get a differentiable function $\langle\Delta\mathcal{L}\rangle_{\boldsymbol{X}}$ that can be re-used in subsequent optimization steps.

The above formulation is agnostic with respect to the loss function. Hence, in principle, Entropy Search should be easy to generalize to different loss functions. But recall that the fidelity of the calculation of Equation (18) depends on the intermediate approximate steps, in particular the choice of discretization measure $\tilde{u}$. We have experimented with other loss functions and found it difficult to find a good measure $\tilde{u}$ providing good performance for many such loss functions. So this paper is limited to the specific choice of the relative entropy loss function. Generalization to other losses is future work.

2.7 Greedy Planning, and its Defects

The previous sections constructed a means to predict, approximately, the expected drop in loss from H new evaluations at locations $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1,\dots,N}$. The remaining task is to optimize these locations. It may seem pointless to construct an optimization algorithm which itself contains an optimization problem, but note that this new optimization problem is quite different from the initial one. It is a numerical optimization problem, of the form described in Section 1: We can evaluate the utility function numerically, without noise, with derivatives, and at hopefully relatively low cost compared to the physical process we are ultimately trying to optimize.

Nevertheless, one issue remains: Optimizing evaluations over the entire horizon H is a dynamical programming problem, which, in general, has cost exponential in H. However, this problem has a particular structure: Apart from the fact that evaluations drawn from Gaussian process measures are exchangeable, there is also other evidence that optimization problems are benign from the point of view of planning. For example, Srinivas et al. (2010)

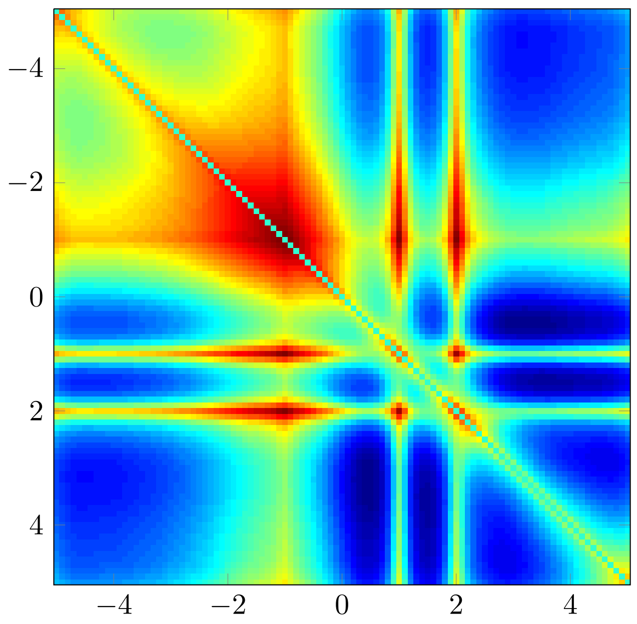

Figure 7: Expected drop in relative entropy (see Section 2.5) from two additional evaluations to the three old evaluations shown in previous plots. First new evaluation on abscissa, second new evaluation on ordinate, but due to the exchangeability of Gaussian process measures, the plot is symmetric. Diagonal elements excluded for numerical reasons. Blue regions are more beneficial than red ones. The relatively complicated structure of this plot illustrates the complexity of finding the optimal H-step evaluation locations.

show that the information gain over the function values is submodular, so that greedy learning of the function comes close to optimal learning of the function. While is is not immediately clear whether this statement extends to our issue of learning about the function's minimum, it is obvious that the greedy choice of whatever evaluation location most reduces expected loss in the immediate next step is guaranteed to never be catastrophically wrong. In contrast to general planning, there are no "dead ends" in inference problems. At worst, a greedy algorithm may choose an evaluation point revealed as redundant by a later step. But thanks to the consistency of Bayesian inference in general, and Gaussian process priors in particular (van der Vaart and van Zanten, 2011), no decision can lead to an evaluation that somehow makes it impossible to learn the true function afterward. In our approximate algorithm, we thus adopt this greedy approach. It remains an open question for future research whether approximate planning techniques can be applied efficiently to improve performance in this planning problem.

2.8 Further Issues

This section digresses from the main line of thought to briefly touch upon some extensions and issues arising from the choices made in previous sections. For the most part, we point out well-known analytic properties and approximations that can be used to generalize the algorithm. Since they apply to Gaussian process regression rather than the optimizer itself, they will not play a role in the empirical evaluation of Section 3.

2.8.1 Derivative Observations

Gaussian process inference remains analytically tractable if instead of, or in addition to direct observations of f, we observe the result of any *linear* operator acting on f. This includes observations of the function's derivatives (Rasmussen and Williams, 2006, §9.4) and, with some caveats, to integral observations (Minka, 2000). The extension is pleasingly straightforward: The kernel defines covariances between function values. Covariances between the function and its derivatives are simply given by

$$\operatorname{cov}\left(\frac{\partial^n f(\boldsymbol{x})}{\prod_i \partial x_i}, \frac{\partial^m f(\boldsymbol{x}')}{\prod_j \partial x'_j}\right) = \frac{\partial^{n+m} k(\boldsymbol{x}, \boldsymbol{x}')}{\prod_i \partial x_i \prod_j \partial x'_j} \tag{19}$$

so kernel evaluations simply have to be replaced with derivatives (or integrals) of the kernel where required. Obviously, this operation is only valid as long as the derivatives and integrals in question exist for the kernel in question. Hence, all results derived in previous sections for optimization from function evaluations can trivially be extended to optimization from function and derivative observations, or from only derivative observations.

2.8.2 Learning Hyperparameters

Throughout this paper, we have assumed kernel and likelihood function to be given. In real applications, this will not usually be the case. In such situations, the hyperparameters defining these two functions, and if necessary a mean function, can be learned from the data, either by setting them to maximum likelihood values, or by full-scale Bayesian inference using Markov chain Monte Carlo methods. See Rasmussen and Williams (2006, §5) and

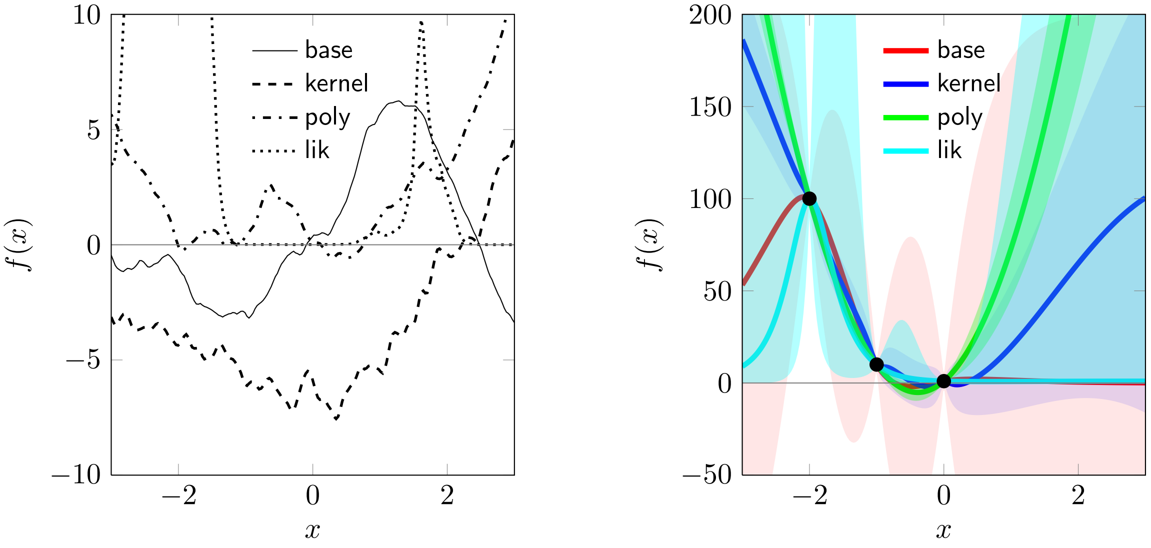

Figure 8: Generalizing GP regression. **Left:** Samples from different priors. **Right:** Posteriors (mean, two standard deviations) after observing three datapoints with negligible noise (kernel parameters differ between the two plots). base: standard GP regression with Matérn kernel. kernel: sum of two kernels (square exponential and rational quadratic) of different length scales and strengths. poly: polynomial (here: quadratic) mean function. lik: Non-Gaussian likelihood (here: logarithmic link function). The scales of both x and $f(x)$ are functions of kernel parameters, so the numerical values in this plot have relevance only relative to each other. Note the strong differences in both mean and covariance functions of the posteriors.

Murray and Adams (2010) for details. In the latter case, the belief $p(f)$ over the function is a mixture of Gaussian processes. To still be able to use the algorithm derived so far, we approximate this belief with a single Gaussian process by calculating expected values of mean and covariance function.

Ideally, one would want to take account of this hierarchical learning process in the decision problem addressed by the optimizer. This adds another layer of computation complexity to the problem, and is outside of the scope of this paper. Here, we contend ourselves with considering the uncertainty of the Gaussian process conditioned on a particular set of hyperparameters.

2.8.3 Limitations and Extensions of Gaussian Processes for Optimization

Like any probability measure over functions, Gaussian process measures are not arbitrarily general. In particular, the most widely used kernels, including the two mentioned above, are *stationary*, meaning they only depend on the difference between locations, not their absolute values. Loosely speaking, the prior "looks the same everywhere". One may argue that many real optimization problems do not have this structure. For example, it may be known that the function tends to have larger functions values toward the boundaries of I or, more vaguely, that it is roughly "bowl-shaped". Fortunately, a number of extensions readily suggest themselves to address such issues (Figure 8).

Parametric Means As pointed out in Section 2.1, we are free to add any parametric general linear model as the mean function of the Gaussian process.

$$m(x) = \sum_i \phi_i(x) w_i \tag{20}$$

Using Gaussian beliefs on the weights w_i of this model, this model may be learned at the same time as the Gaussian process itself (Rasmussen and Williams, 2006, §2.7). Polynomials such as the quadratic $\boldsymbol{\phi}(\boldsymbol{x}) = [\boldsymbol{x}; \boldsymbol{x}\boldsymbol{x}^\intercal]$ are beguiling in this regard, but they create an explicit "origin" at the center of I, and induce strong long-range correlations between opposite ends of I. This seems pathological: In most settings, observing the function on one end of I should not tell us much about the value at the opposite end of I. But we may more generally choose any feature set for the linear model. For example, a set of radial basis functions $\phi_i(\boldsymbol{x}) = \exp(\|\boldsymbol{x} - \boldsymbol{c}_i\|^2/\ell_i^2)$ around locations $\boldsymbol{c}_i$ at the rims of I can explain large function values in a region of width ℓ_i around such a feature, without having to predict large values at the center of I. This idea can be extended to a nonparametric version, described in the next point.

Composite Kernels Since kernels form a semiring, we may sum a kernel of large length scale and large signal variance and a kernel of short length scale and low signal variance. For example

$$k(x, x') = k_{\text{SE}}(x, x'; s_1, \boldsymbol{S}_1) + k_{\text{RQ}}(x, x', s_2, \boldsymbol{S}_2, \alpha_2) \qquad s_1 \gg s_2; \boldsymbol{S}_1^{ij} \gg \boldsymbol{S}_2^{ij}\ \forall i, j \tag{21}$$

yields a kernel over functions that, within the bounded domain I, look like "rough troughs": global curvature paired with local stationary variations. A disadvantage of this prior is that it thinks "domes" just as likely as "bowls". An advantage is that it is a very flexible framework, and does not induce unwanted global correlations.

Nonlinear Likelihoods An altogether different effect can be achieved by a non-Gaussian, non-linear likelihood function. For example, if f is known to be strictly positive, one may assume the noise model

$$p(y \,|\, g) = \mathcal{N}(y; \exp(g), \sigma^2); \quad f = \exp(g) \tag{22}$$

and learn g instead of f. Since the logarithm is a convex function, the minimum of the latent g is also a minium of f. Of course, this likelihood leads to a non-Gaussian posterior. To retain a tractable algorithm, approximate inference methods can be used to construct approximate Gaussian posteriors. In our example (labeled lik in Figure 8), we used a Laplace approximation: It is straightforward to show that Equation (22) implies

$$\left.\frac{\partial \log p(y \,|\, g)}{\partial g}\right|_{g=\hat{g}} \overset{!}{=} 0 \quad \Rightarrow \hat{g} = \log y \qquad \left.\frac{\partial^2 \log p(y \,|\, g)}{\partial^2 g}\right|_{g=\hat{g}} = \frac{y^2}{\sigma^2} \tag{23}$$

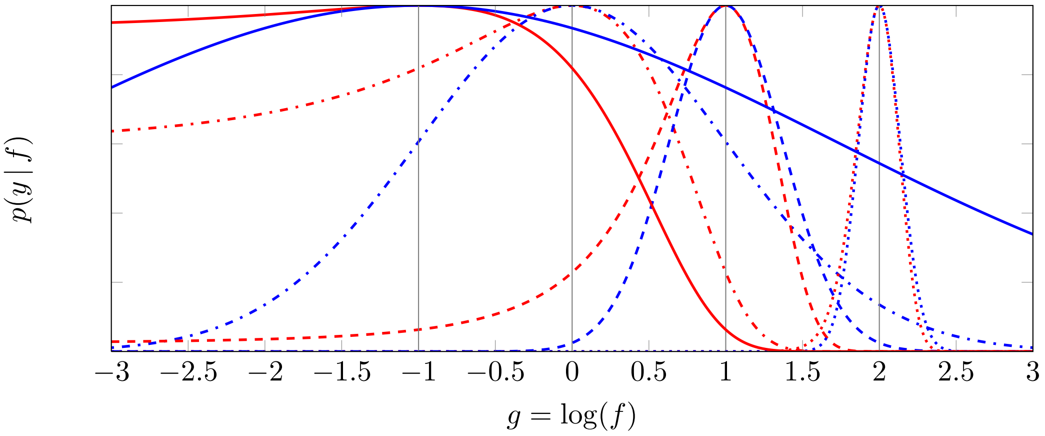

Figure 9: Laplace approximation for a logarithmic Gaussian likelihood. True likelihood in red, Gaussian approximation in blue, maximum likelihood solution marked in grey. Four log relative values $a = \log(y/\sigma)$ of sample y and noise σ (scaled in height for readability). $a = -1$ (solid); $a = 0$ (dash-dotted); $a = 1$ (dashed); $a = 2$ (dotted). The approximation is good for $a \gg 0$.

so a Laplace approximation amounts to a heteroscedastic noise model, in which an observation (y, σ^2) is incorporated into the Gaussian process as $(\log(y), (\sigma/y)^2)$. This approximation is valid if $\sigma \ll y$ (see Figure 3). For functions on logarithmic scales, however, finding minima smaller than the noise level, at logarithmic resolution, is a considerably harder problem anyway.

The right part of Figure 8 shows posteriors produced using the three approaches detailed above, and the base case of a single kernel with strong signal variance, when presented with the same three data points, with very low noise. The strong difference between the posteriors may be disappointing, but it is a fundamental aspect of inference: Different prior assumptions lead to different posteriors, and function space inference is impossible without priors. Each of the four beliefs shown in the Figure may be preferable over the others in particular situations. The polynomial mean describes functions that are almost parabolic. The exponential likelihood approximation is appropriate for functions with an intrinsic logarithmic scale. The sum kernel approach is pertinent for the search for local minima of globally stationary functions. Classic methods based on polynomial approximations are a lot more restrictive than any of the models described above.

Perhaps the most general option is to use additional prior information $\mathcal{I}$ giving $p(x_{\min} \mid \mathcal{I})$, independent of $p(f)$, to encode outside information about the location of the minimum. Unfortunately, this is intractable in general. But it may be approached through approximations. This option is outside of the scope of this paper, but will be the subject of future work.

Algorithm 1 Entropy Search

```
procedure EntropySearch($k, l = p(y \mid f(x)), u, H, (\boldsymbol{x}, \boldsymbol{y})$)
    $\tilde{\boldsymbol{x}} \sim u(\boldsymbol{x}, \boldsymbol{y})$    ▷ discretize using measure $u$ (Section 2.2)
    $[\boldsymbol{\mu}, \boldsymbol{\Sigma}, \Delta\boldsymbol{\mu}_x, \Delta\boldsymbol{\Sigma}_x] \leftarrow \mathrm{GP}(k, l, \boldsymbol{x}, \boldsymbol{y})$    ▷ infer function, innovation, from GP prior (2.1)
    $[\hat{q}_{\min}(\tilde{x}), \frac{\partial \hat{q}_{\min}}{\partial \mu}, \frac{\partial^2 \hat{q}_{\min}}{\partial \mu \partial \mu}, \frac{\partial \hat{q}_{\min} x}{\partial \Sigma}] \leftarrow \mathrm{EP}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$    ▷ approximate $\hat{p}_{\min}$ (2.3)
    if H=0 then
        return $q_{\min}$    ▷ At horizon, return belief for final decision
    else
        $x' \leftarrow \arg\min \langle \mathcal{L} \rangle_x$    ▷ predict information gain; Eq. (18)
        $y' \leftarrow$ Evaluate($f(x')$)    ▷ take measurement
        EntropySearch($k, l, u, H - 1, (\boldsymbol{x}, \boldsymbol{y}) \cup (x', y')$)    ▷ move to next evaluation
    end if
end procedure
```

2.9 Summary – the Entire Algorithm

Algorithm 1 shows pseudocode for Entropy Search. It takes as input the prior, described by the kernel k, and the likelihood $l = p(y \mid f(x))$, as well as the discretization measure u (which may itself be a function of previous data, the Horizon H, and any previously collected observations $(\boldsymbol{x}, \boldsymbol{y})$. To choose where to evaluate next, we first sample discretization points from u, then calculate the current Gaussian belief over f on the discretized domain, along with its derivatives. We construct an approximation to the belief over the minimum using Expectation Propagation, again with derivatives. Finally, we construct a first order approximation on the expected information gain from an evaluation at x' and optimize numerically. We evaluate f at this location, then the cycle repeats. Upon publication of this work, MATLAB source code for the algorithm and its sub-routines will be made available online.

3. Experiments

Figures in previous sections provided some intuition and anecdotal evidence for the efficacy of the various approximations used by Entropy Search. In this section, we compare the resulting algorithm to two Gaussian process global optimization heuristics: Expected Improvement, Probability of Improvement (Section 1.1), as well as to a continuous armed bandit algorithm: GP-UCB (Srinivas et al., 2010). For reference, we also compare to a number of numerical optimization algorithms: Trust-Region-Reflective (Coleman and Li, 1996, 1994), Active-Set (Powell, 1978b,a), interior point (Byrd et al., 1999, 2000; Waltz et al., 2006), and a naïvely projected version of the BFGS algorithm (Broyden et al., 1965; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). We avoid implementation bias by using a uniform code framework for the three Gaussian process-based algorithms, i.e. the algorithms share code for the Gaussian process inference and only differ in the way they calculate their utility. For the local numerical algorithms, we used third party code: The projected BFGS method is based on code by Carl Rasmussen[3], the other methods come from version 6.0 of the optimization toolbox of MATLAB[4].

In some communities, optimization algorithms are tested on hand-crafted test functions. This runs the risk of introducing bias. Instead, we compare our algorithms on a number of functions sampled from a generative model. In the first experiment, the function is sampled from the model used by the GP algorithms themselves. This eliminates all model-mismatch issues and allows a direct comparison of other GP optimizers to the probabilistic optimizer. In a second experiment, the functions were sampled from a model strictly more general than the model used by the algorithms, to show the effect of model mismatch.

3.1 Within-Model Comparison

The first experiment was carried out over the 2-dimensional unit domain $I = [0, 1]^2$. To generate test functions, 1000 function values were jointly sampled from a Gaussian process with a squared-exponential covariance function of length scale $\ell = 0.1$ in each direction and unit signal variance. The resulting posterior mean was used as the test function. All algorithms had access to noisy evaluations of the test functions. For the benefit of the numerical optimizers, noise was kept relatively low: Gaussian with standard deviation $\sigma = 10^{-3}$. All algorithms were tested on the same set of 40 test functions, all Figures in this section are averages over those sets of functions. It is nontrivial to provide error bars on these average estimates, because the data sets have no parametric distribution. But the regular structure of the plots, given that individual experiments were drawn i.i.d., indicates that there is little remaining stochastic error.

After each function evaluation, the algorithms were asked to return a best guess for the minimum $x_{\min}$. For the local algorithms, this is simply the point of their next evaluation. The Gaussian process based methods returned the global minimum of the mean belief over the function (found by local optimization with random restarts). Figure 10 shows the difference between the global optimum of the function and the function value at the reported best guesses. Since the best guesses do not in general lie at a datapoint, their quality can actually decrease during optimization. The most obvious feature of this plot is that local optimization algorithms are not adept at finding global minima, which is not surprising, but gives an intuition for the difficulty of problems sampled from this generative model. The plot shows a clear advantage for Entropy Search over its competitors, even though the algorithm does not directly aim to optimize this particular loss function. The flattening out of the error of all three global optimizers toward the right is due to evaluation noise (recall that evaluations include Gaussian noise of standard deviation 10^{-3}). Interestingly, Entropy Search flattens out at an error almost an order of magnitude lower than that of the nearest competitor, Expected Improvement. One possible explanation for this behavior is a pathology in the classic heuristics: Both Expected Improvement and Probability of Improvement require a “current best guess” η, which has to be a point estimate, because proper marginalization over an uncertain belief is not tractable. Due to noise, it can

[3] `http://www.gaussianprocess.org/gpml/code/matlab/util/minimize.m`, version using BFGS: personal communication

[4] `http://www.mathworks.de/help/toolbox/optim/rn/bsqj_zi.html`

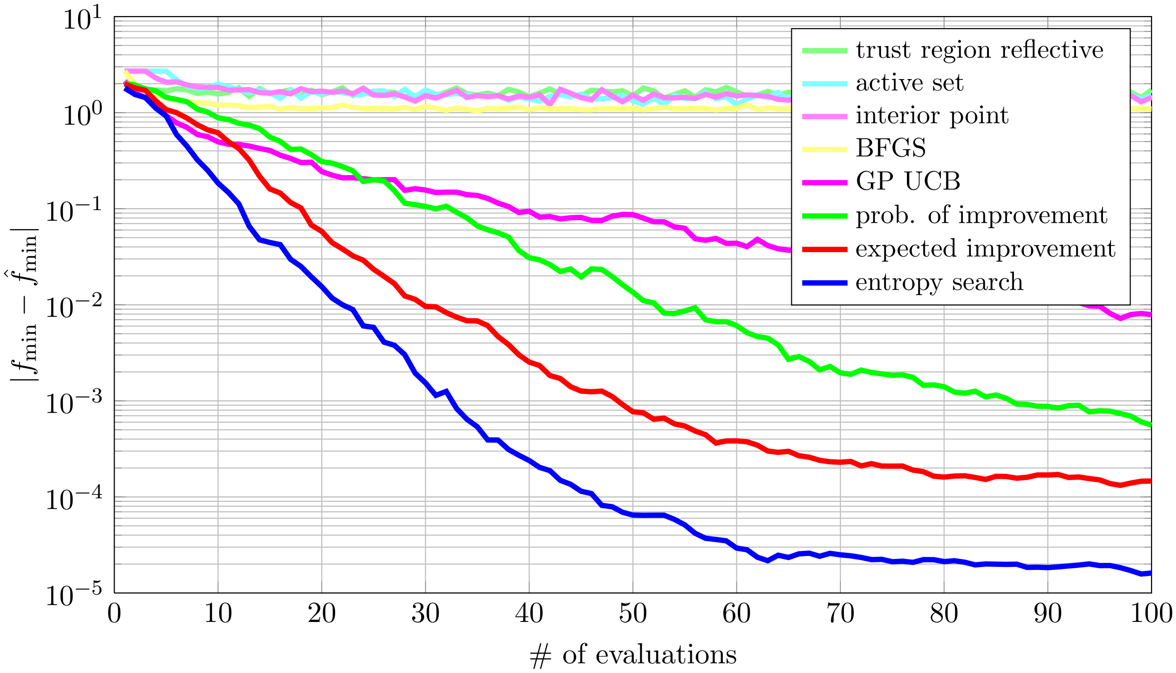

Figure 10: Distance of function value at optimizers' best guess for $x_{\min}$ from true global minimum. Log scale.

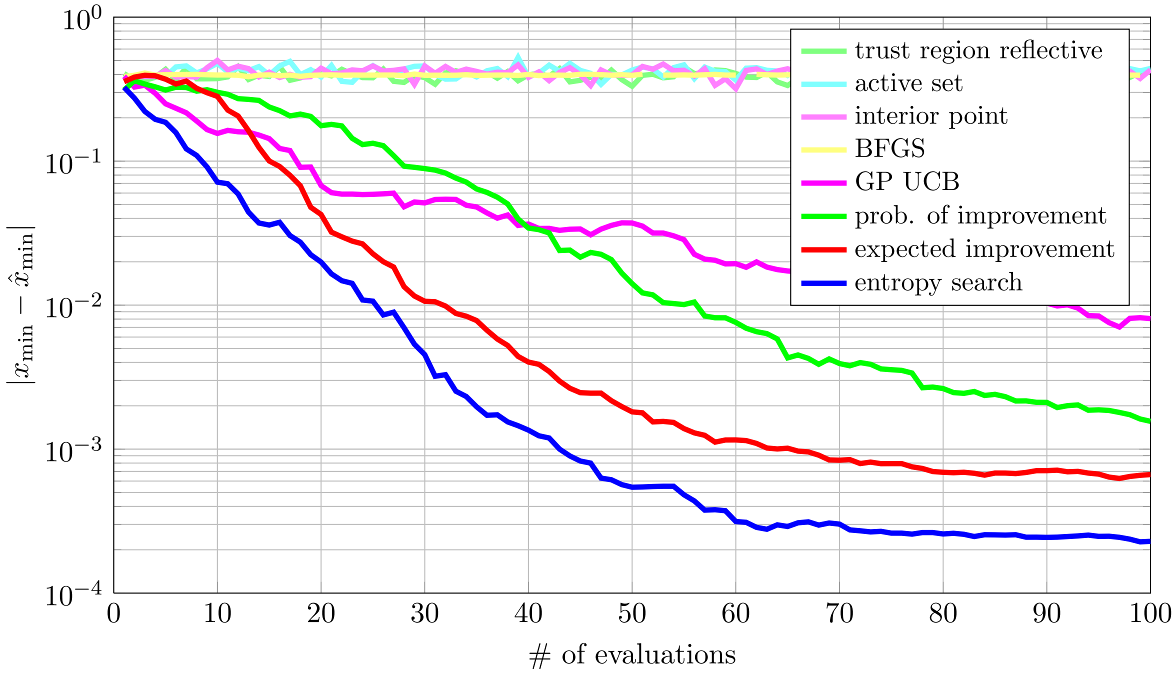

Figure 11: Euclidean distance of optimizers' best guess for $x_{\min}$ from truth. Log scale.

thus happen that this best guess is overly optimistic, and the algorithm then explores too aggressively in later stages.

Figure 11 shows data from the same experiments as the previous figure, but plots Euclidean distance from the true global optimum in input space, rather than in function value space. The results from this view are qualitatively similar to those shown in Figure 10.

Since Entropy Search attempts to optimize information gain from evaluations, one would also like to compare to algorithms on the entropy loss function. However, this is challenging. First, the local optimization algorithms provide no probabilistic model of the function and can thus not provide this loss. But even for the optimization algorithms based on Gaussian process measures, it is challenging to evaluate this loss *globally* with good resolution. The only option is to approximately calculate entropy, using the very algorithm introduced in this paper. Doing so amounts to a kind of circular experiment that Entropy Search wins by definition, so we omit it here.

We pointed out in Section 1.1 that the bandit setting differs considerably from the kind of optimization discussed in this paper, because bandit algorithms try to minimize regret, rather than improve an estimate of the function's optimum. To clarify this point further, Figure 12 shows the regret

$$r(T) = \sum_{t=1}^{T} [y_t - f_{\min}] \tag{24}$$

for each of the algorithms. Notice that probability of improvement, which performs worst among the global algorithms as seen from the previous two measures of performance, achieves the lowest regret. The intuition here is that this heuristic focusses evaluations

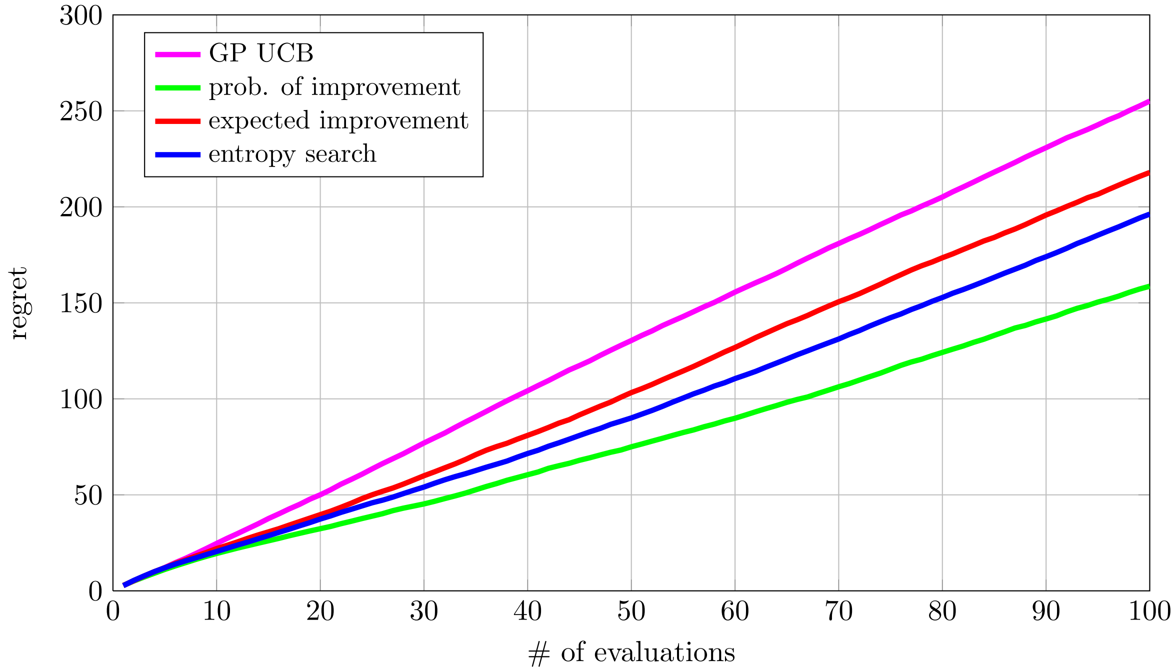

Figure 12: Regret as a function of number of evaluations.

on regions known to give low function values. In contrast, the actual value of the function *at the evaluation point* has no special role in Entropy Search. The utility of an evaluation point only depends on its expected effect on knowledge about the minimum of the function.

Surprisingly, the one algorithm explicitly designed to achieve low regret, GP-UCB, performs worst in this comparison. This algorithm chooses evaluation points according to (Srinivas et al., 2010)

$$x_{\text{next}} = \arg\min_x [\mu(x) - \beta^{1/2}\sigma(x)] \qquad \text{where} \qquad \beta = 4(D+1)\log T + C(k,\delta) \tag{25}$$

with T, the number of previous evaluations, D, the dimensionality of the input domain, and $C(k, \delta)$ is a constant that depends on some analytic properties of the kernel k and a free parameter, $0 < \delta < 1$. We found it hard to find a good setting for this δ, which clearly has influence on the algorithm's performance. The results shown here represent the best performance over a set of 4 experiments with different choices for δ. They appear to be slightly worse than, but comparable to the empirical performance reported by the original paper on this algorithm (Srinivas et al., 2010, Figure 5a).

3.2 Out-of-Model Comparison

In the previous section, the algorithms attempted to find minima of functions sampled from the prior used by the algorithms themselves. In real applications, one can rarely hope to be so lucky, but hierarchical inference can be used to generalize the prior and construct a relatively general algorithm. But what if even the hierarchically extended prior class does not contain the true function? Qualitatively, it is clear that, beyond a certain

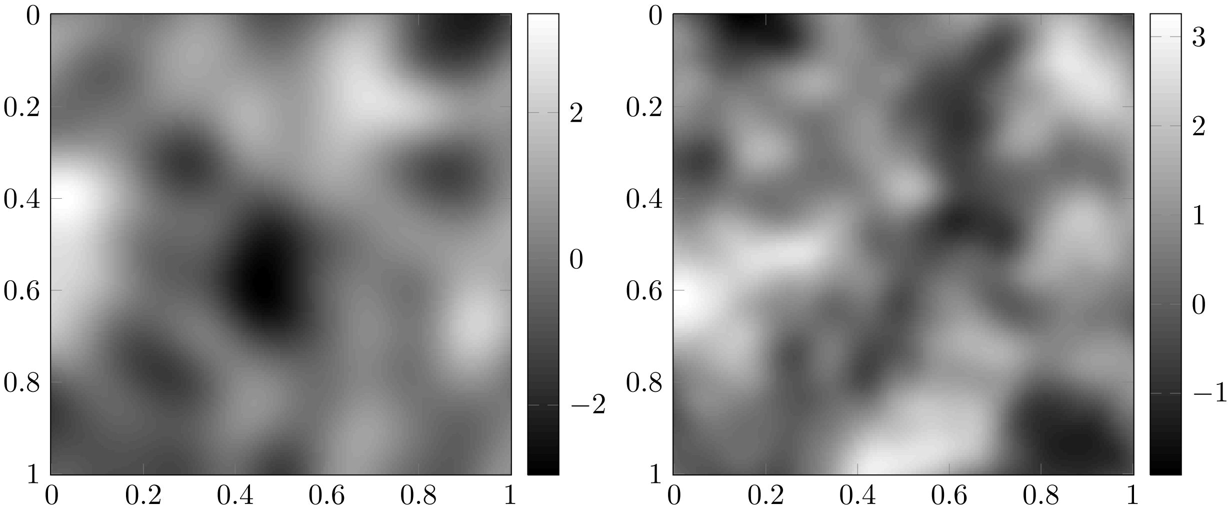

Figure 13: **Left:** A sample from the GP prior with squared exponential kernel used in the on-model experiments of Section 3.1. **Right:** Sample from prior with the rational quadratic kernel used for the out of model comparison of Section 3.2.

point of model-mismatch, all algorithms can be made to perform arbitrarily badly. The poor performance of local optimizers (which may be interpreted as building a quadratic model) in the previous section is an example of this effect. In this section, we present results of the same kind of experiments as in the previous section, but on a set of 30 two-dimensional functions sampled from a Gaussian process prior with *rational quadratic* kernel, with the same length scale and signal variance as above, and scale mixture parameter $\alpha = 1$ (see Equation (7)). This means samples evolve over an infinite number of different length scales, including both longer and shorter scales than those covered by the priors of the algorithms (Figure 13). Figure 14 shows error on function values, Figure 15 Euclidean error in input space, Figure 16 regret. Note the different scales for the ordinate axes relative to the corresponding previous plots: While Entropy Search still (barely) outperforms the competitors, all three algorithms perform worse than before; and their errors become more similar to each other. However, they still manage to discover good regions in the domain, demonstrating a certain robustness to model-mismatch.

4. Conclusion

This paper presented a new probabilistic paradigm for global optimization, as an inference problem on the minimum of the function, rather than the problem of collecting iteratively lower and lower function values. We argue that this description is closer to practitioners' requirements than classic response surface optimization, bandit algorithms, or other, heuristic, global optimization algorithms. In the main part of the paper, we constructed Entropy Search, a practical probabilistic global optimization algorithm, using a series of analytic assumptions and numerical approximations: A particular family of priors over functions

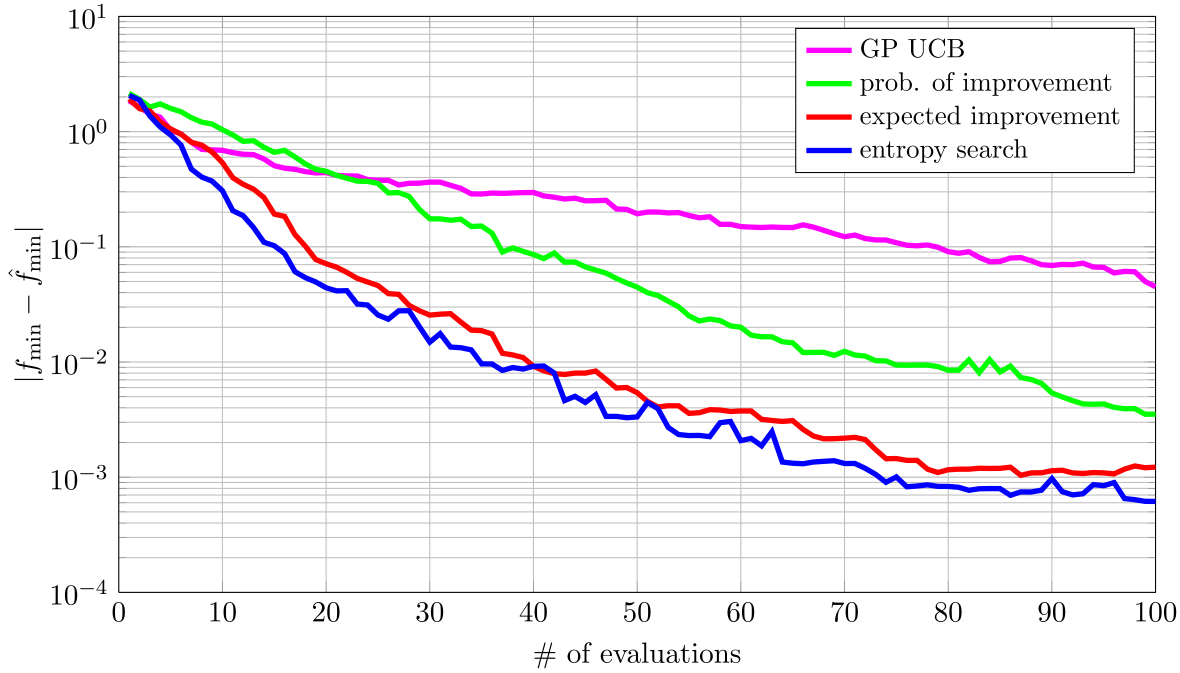

Figure 14: Function value error, off-model tasks.

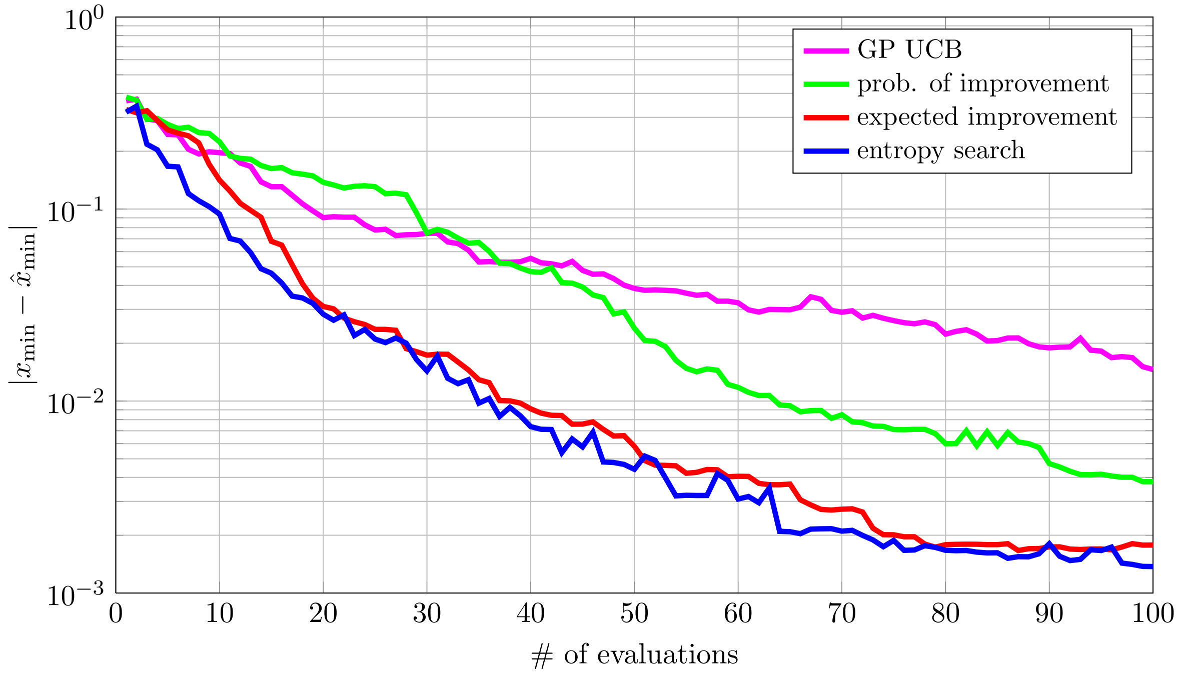

Figure 15: Error on $x_{\min}$, off-model tasks.

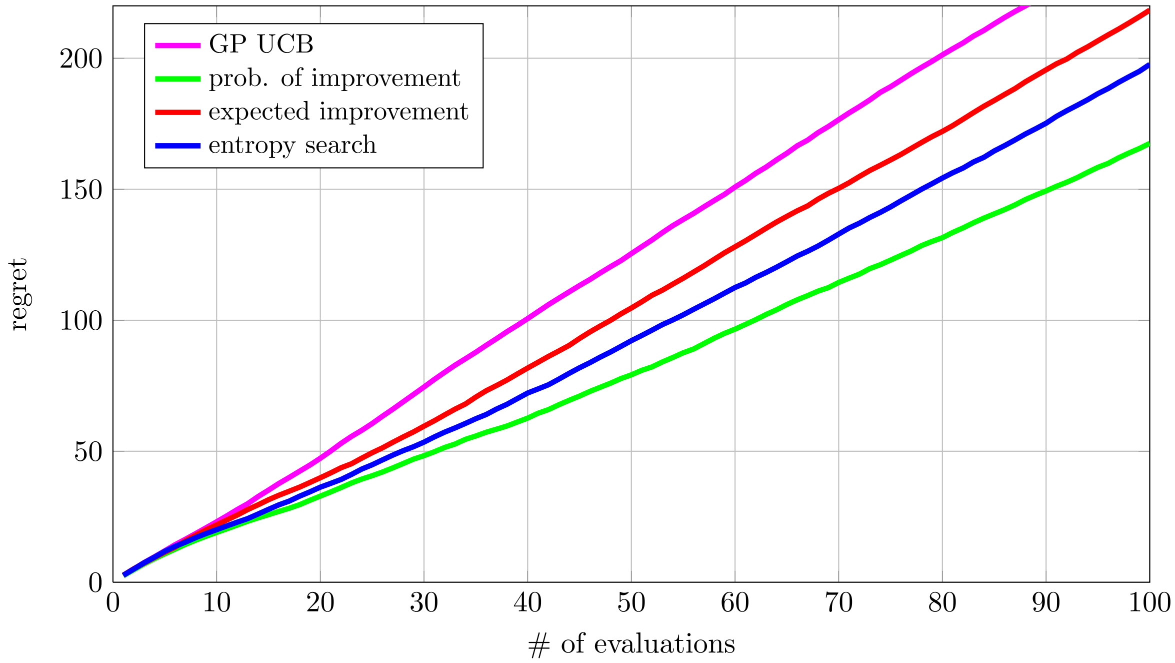

Figure 16: Regret, off-model tasks.

(Gaussian processes); constructing the belief $p_{\min}$ over the location of the minimum on an irregular grid to deal with the curse of dimensionality; and using Expectation Propagation toward an efficient analytic approximation. The Gaussian belief allows analytic probabilistic predictions of the effect of future datapoints, from which we constructed a first-order approximation of the expected change in relative entropy of $p_{\min}$ to a base measure. For completeness, we also pointed out some already known analytic properties of Gaussian process measures that can be used to generalize this algorithm. We showed that the resulting algorithm outperforms both directly and distantly related competitors through its more elaborate, probabilistic description of the problem. This increase in performance is exchanged for somewhat increased computational cost (Entropy Search costs are a constant multiple of that of classic Gaussian process global optimizers). So this algorithm is more suited for problems were evaluating the function itself carries considerable cost. Nevertheless, it provides a natural description of the optimization problem, by focusing on the performance under a loss function at the horizon, rather than function values returned during the optimization process. It allows the practitioner to explicitly encode prior knowledge in a flexible way, and adapts its behavior to the user's loss function.

Acknowledgments

We would like to thank Martin Kiefel for valuable feedback, as well as Tom Minka for an interesting discussion.

Appendix A. Mathematical Appendix

The notation in Equation (1) can be read, sloppily, to mean "$p_{\min}(x)$ is the probability that the value of f at x is lower than at any other $\tilde{x} \in I$". For a continuous domain, though, there are uncountably many other $\tilde{x}$. To give more precise meaning to this notation, consider the following argument. Let there be a sequence of locations $\{x_i\}_{i=1,\dots,N}$, such that for $N \to \infty$ the density of points at each location converges to a measure $m(x)$ nonzero on every open neighborhood in I. If the stochastic process $p(f)$ is sufficiently regular to ensure samples are almost surely continuous (see footnote in Section 2.1), then almost every sample can be approximated arbitrarily well by a staircase function with steps of width $m(x_i)/N$ at the locations x_i, in the sense that $\forall \epsilon > 0\ \exists N_0 > 0$ such that, $\forall N > N_0 : |f(x) - f(\arg\min_{x_j, j=1,\dots,N} |x - x_j|)| < \epsilon$, where $|\cdot|$ is a norm (all norms on finite-dimensional vector spaces are equivalent). This is the original reason why samples from sufficiently regular Gaussian processes can be plotted using finitely many points, in the way used in this paper. We now *define* the notation used in Equation (1) to mean the following limit, where it exists.

$$
\begin{aligned}
p_{\min}(x) &= \int p(f) \prod_{\tilde{x} \neq x} \theta(f(\tilde{x}) - f(x))\,\mathrm{d}f \\
&\equiv \lim_{\substack{N \to \infty \\ |x_i - x_{i-1}| \cdot N \to m(x)}} \int p[f(\{x_i\}_{i=1,\dots,N})] \prod_{i=1; i \neq j}^{N} \theta[f(x_i) - f(x_j)] \mathrm{d}f(\{x_i\}_{i=1,\dots,N}) \cdot |x_i - x_{i-1}| \cdot N
\end{aligned}
\qquad (26)
$$

In words: The "infinite product" is meant to be the limit of finite-dimensional integrals with an increasing number of factors and dimensions, where this limit exists. In doing so, we have side-stepped the issue of whether this limit exists for any particular Gaussian process (i.e. kernel function). We do so because the theory of suprema of stochastic processes is highly nontrivial. We refer the reader to a friendly but demanding introduction to the topic by Adler (1990). From our applied standpoint, the issue of whether (26) is well defined for a particular Gaussian prior is secondary: If it is known that the true function is continuous and bounded, than it has a well-defined supremum, and the prior should reflect this knowledge by assigning sufficiently regular beliefs. If the actual prior is such that we expect the function to be discontinuous, it should be clear that optimization is extremely challenging anyway. We conjecture that the finer details of the region between these two domains have little relevance for communities interested in optimization.

References

R.J. Adler. *The geometry of random fields.* Wiley, 1981.

R.J. Adler. An introduction to continuity, extrema, and related topics for general Gaussian processes. *Lecture Notes-Monograph Series*, 12:i–iii+v–vii+ix+1–55, 1990.

Benoit. Note sûre une méthode de résolution des équations normales provenant de l'application de la méthode des moindres carrés a un système d'équations linéaires en nombre inférieure a celui des inconnues. Application de la méthode a la résolution d'un système défini d'équations linéaires. (procédé du Commandant Cholesky). *Bulletin geodesique*, 7(1):67–77, 1924.

S.P. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge Univ Press, 2004.

C.G. Broyden et al. A class of methods for solving nonlinear simultaneous equations. *Math. Comp*, 19(92):577–593, 1965.

R.H. Byrd, M.E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

R.H. Byrd, J.C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

T.F. Coleman and Y. Li. On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds. *Mathematical programming*, 67(1):189–224, 1994.

T.F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

R.T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946.

J. Cunningham, P. Hennig, and S. Lacoste-Julien. Gaussian probabilities and expectation propagation. *under review. Preprint at arXiv:1111.6832 [stat.ML]*, November 2011.

R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3): 317, 1970.

D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computing*, 24(109):23–26, 1970.

S. Grünewälder, J.Y. Audibert, M. Opper, and J. Shawe-Taylor. Regret bounds for Gaussian process bandit problems. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

P. Hennig. Optimal reinforcement learning for Gaussian systems. In *Advances in Neural Information Processing Systems*, 2011.

M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

K. Itō. On stochastic differential equations. *Memoirs of the American Mathematical Society*, 4, 1951.

E.T. Jaynes and G.L. Bretthorst. *Probability Theory: the Logic of Science.* Cambridge University Press, 2003.

D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems*, 18, 2005.

A.N. Kolmogorov. Grundbegriffe der Wahrscheinlichkeitsrechnung. *Ergebnisse der Mathematik und ihrer Grenzgebiete*, 2, 1933.

D.G. Krige. A statistical approach to some basic mine valuation and allied problems at the Witwatersrand. Master's thesis, University of Witwatersrand, 1951.

S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

H. Lazard-Holly and A. Holly. Computation of the probability that a d-dimensional normal variable belongs to a polyhedral cone with arbitrary vertex. Technical report, Mimeo, 2003.

D.J. Lizotte. *Practical Bayesian Optimization.* PhD thesis, University of Alberta, 2008.

D.J.C. MacKay. Choice of basis for Laplace approximation. *Machine Learning*, 33(1):77–86, 1998a.

D.J.C. MacKay. Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998b.

B. Matérn. Spatial variation. *Meddelanden fran statens Skogsforskningsinstitut*, 49(5), 1960.

T.P. Minka. Deriving quadrature rules from Gaussian processes. Technical report, Statistics Department, Carnegie Mellon University, 2000.

T.P. Minka. Expectation Propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann. ISBN 1-55860-800-1.

I. Murray and R.P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. *arXiv:1006.0868*, 2010.

J. Nocedal and S.J. Wright. *Numerical optimization.* Springer Verlag, 1999.

M.A. Osborne, R. Garnett, and S.J. Roberts. Gaussian processes for global optimization. In *3rd International Conference on Learning and Intelligent Optimization (LION3)*, 2009.

R.L. Plackett. A reduction formula for normal multivariate integrals. *Biometrika*, 41(3-4): 351, 1954.

M.J.D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. *Nonlinear programming*, 3(0):27–63, 1978a.

M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. *Numerical analysis*, pages 144–157, 1978b.

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

L.M. Schmitt. Theory of genetic algorithms ii: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoretical Computer Science*, 310(1-3):181–231, 2004.

M. Seeger. Expectation propagation for exponential families. Technical report, U.C. Berkeley, 2008.

D.F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2010.

M.B. Thompson and R.M. Neal. Slice sampling with adaptive multivariate steps: The shrinking-rank method. *Arxiv preprint arXiv:1011.4722*, 2010.

G.E. Uhlenbeck and L.S. Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823, 1930.

A.W. van der Vaart and J.H. van Zanten. Information rates of nonparametric Gaussian process methods. *Journal of Machine Learning Research*, 12:2095–2119, 2011.

R.A. Waltz, J.L. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3):391–408, 2006.

N. Wiener and P. Masani. The prediction theory of multivariate stochastic processes. *Acta Mathematica*, 98(1):111–150, 1957.