Active Area Search via Bayesian Quadrature

Yifei Ma
Machine Learning Department
Carnegie Mellon University
yifeim@cs.cmu.edu

Roman Garnett
Knowledge Discovery Department
University of Bonn
rgarnett@uni-bonn.de

Jeff Schneider
Robotics Institute
Carnegie Mellon University
schneide@cs.cmu.edu

Abstract

The selection of data collection locations is a problem that has received significant research attention from classical design of experiments to various recent active learning algorithms. Typical objectives are to map an unknown function, optimize it, or find level sets in it. Each of these objectives focuses on an assessment of individual points. The introduction of set kernels has led to algorithms that instead consider labels assigned to sets of data points. In this paper we combine these two concepts and consider the problem of choosing data collection locations when the goal is to identify regions whose set of collected data would be labeled positively by a set classifier. We present an algorithm for the case where the positive class is defined in terms of a region's average function value being above some threshold with high probability, a problem we call *active area search*. To this end, we model the latent function using a Gaussian process and use Bayesian quadrature to estimate its integral on predefined regions. Our method is the first which directly solves the active area search problem. In experiments it outperforms previous algorithms that were developed for other active search goals.

1 INTRODUCTION

Traditionally, active learning assumes that a label is associated with each observable data point, which may be revealed upon sampling. Here we consider an alternative setting, where the labels defining our objective cannot be observed directly but must rather be inferred from auxillary observations. In particular, we consider actively gathering point observations of a smooth function so as to identify regions with large average value quickly.

An important application of this problem is found in environmental monitoring. Consider a small, autonomous boat equipped with water quality sensors (such as the ones described by Valada et al. [2012]). The variability of the sensors and environmental conditions on a river mean that no single sensor reading will ever be sufficient to identify a significant pollution issue. However, a set of readings within a certain region that satisfy some pattern can indicate a real pollution problem. Although a boat gives us the capability to take a measurement anywhere, it does not provide the sensing bandwidth to monitor every location all the time. Besides, sensing cost dominates travel cost in many cases.[1] Therefore we need an algorithm to sequentially choose sensing locations with a goal of identifying polluted regions. This is an example of an active search problem [Garnett et al., 2012] that we refer to as the *active area search problem*. A critical distinction between point-based active search and this setting is that the class labels (i.e., high concentration of pollutants) are not directly accessible to the learning algorithm. They must be inferred by sampling some underlying unknown function that defines them. In this paper that definition will simply be that the average of the function over the region exceeds some threshold.

We will assume that our function of interest is defined on a domain that has been *a priori* subdivided into a set of regions (for example, into a grid). We further assume that the low-level response is smooth over the entire space. Our job is to actively decide where to take measurements so that we can find as many polluted re-

---

[1]A typical dissolved oxygen sensor requires about one minute for the reading to settle down after moving [Valada et al., 2012], which is enough time for the small boat to travel end-to-end in the areas we've considered so far. Similarly, any application requiring *in situ* lab analysis of samples would have this property.

gions with high confidence as possible. Mathematically, we assume that the low-level response is a random function with a Gaussian process (GP) prior, $f \sim \mathcal{GP}(\mu, \kappa)$, whose hyperparameters are externally designed.[2] We define the reward of a given region to be a binary variable indicating whether the average function value in this region is higher than a given threshold τ with probability greater than a given confidence θ (equivalently, $(1 - \theta)$ can be interpreted as the allowed false-positive rate). The GP prior on f implies a normal distribution on its average value in a given region, a property that has been applied to estimate difficult integrals under the names *Bayesian quadrature* (BQ), *Bayesian Monte Carlo,* and *Bayes–Hermite quadrature* [O'Hagan, 1991; Minka, 2000; Ghahramani and Rasmussen, 2002]. We exploit this property as well.

The particular strategy we use for active search of interesting regions is a one-step lookahead fashion heuristic. Given a potential observation location, we use BQ to compute the distribution of our updated beliefs on all regions, marginalizing the unknown function value under our current model. This allows us to calculate the expected sum of binary rewards for each region after incorporating an observation at that location. We then observe f where this expectation is maximized. The resulting algorithm naturally achieves exploration and exploitation, and, surprisingly, is related to optimal experiment design for BQ in the single-region case.

1.1 Related Work

In Bayesian literature, modeling the smoothness of a latent function by imposing a Gaussian process prior [Rasmussen, 2006] is commonplace. Previous work has developed algorithms to efficiently infer a function over its entire domain with few samples [Krause et al., 2008; Houlsby et al., 2011], to estimate the shape of a particular level set [Gotovos et al., 2013; Bryan et al., 2005], and to search for its global optimum [Jones et al., 1998; Osborne et al., 2009; Tesch et al., 2011]. Here, we introduce and provide an algorithm to do active area search, where our goal is to successively evaluate the latent function to quickly identify regions with high averge value.

A related line of research is *level-set estimation,* where the goal is to identify the set of points in the domain where the function value equals a given number τ'. Bryan et al. [2005] were among the first to address this problem with GP assumptions. Their heuristic chooses points where there is high uncertainty in both the latent function and whether the point is above or below the

level set. Gotovos et al. [2013] took a step further and derived a theoretically justified bandit optimization algorithm, which they call LSE, that is similar to the well-known GP-UCB algorithm [Srinivas et al., 2010], which further traces back to Auer et al. [2002]. The LSE algorithm guarantees that, with high probability, all regions ε-close to the level τ' can be found and regions with higher and lower function values marked. Although level set seeking algorithms could be applied in active area search, they actually solve a different problem. They are fundamentally point-based, which means they seek to accurately represent whether each individual point is above or below the level set threshold. Active area search is concerned with the properties of areas rather than points. Practically speaking, the problem with using level set seeking methods is the large amount of variability and transients in the environment. The true level set boundaries could be very convoluted and require impossibly many samples to identify, whereas finding positive areas need not require nearly as many.

Krause et al. [2008] discretized the input space and used active learning to maximize the (mutual) information gain between the labeled and unlabeled points. The goal in that work is to infer the latent function over the entire domain. Again, inferring the entire latent function is useful for active area search but requires far more samples than needed for the job. For example, accurately mapping an area that is clearly not postive is unnecessary.

A final line of related active sampling methods aims to search for the global optimum of an unknown function [Jones et al., 1998; Osborne et al., 2009; Tesch et al., 2011]. These methods also balance exploration and exploitation over the function domain, but are point-based optimizers.

The validity of using a GP prior to approximately study the average-case or worst-case error of the region average for a whole class of functions is justified by previous work [O'Hagan, 1991; Minka, 2000]. For integrals on many function classes, e.g., polynomials of limited degrees and linear splines, the optimal designs for evaluating the function and the multipliers of the function values for guaranteed worst-case errors can be well approximated by GPs with proper kernel construction.

2 PROBLEM DEFINITION

Let $f \colon \mathbb{R}^d \to \mathbb{R}$ be an unknown, smooth function on the input space, and let $p_g \colon \mathbb{R}^d \to \mathbb{R}^+, g = 1, \ldots, G$ be a family of given probability density functions. We consider estimating the expected value of f under each

---

[2]In a real world scenario where multiple domains are studied in a sequence, the GP prior parameters can be trained on a pilot dataset which is more densely surveyed as a result of an earlier stage.

p_g, and we will denote this random varibable $\mathcal{M}_g$:

$$\mathcal{M}_g \triangleq \int f(x) p_g(x) \, \mathrm{d}x. \qquad (1)$$

We are particularly interested in the case when the densities p_g are uniform on two-dimensional box-bounded regions $A_g \triangleq [\ell_{g,1}, u_{g,1}] \times [\ell_{g,2}, u_{g,2}]$. Now the quantities of interest are the averages of f on the A_g:

$$\mathcal{M}_g = \frac{1}{|A_g|} \int_{A_g} f(x) \, \mathrm{d}x. \qquad (2)$$

According to Minka [2000], for many classes of smooth random functions, it is reasonable to assume a Gaussian process prior on f:

$$p(f) \triangleq \mathcal{GP}(f; \mu, \kappa).$$

Throughout we will assume a zero prior mean function μ. We assume that we can make noisy observations of f at chosen points in the domain. That is, given an arbitrary $x_* \in \mathbb{R}^d$, we may receive the associated observation

$$y_* \triangleq f(x_*) + \varepsilon,$$

where ε is zero-mean iid Gaussian distributed with variance σ_n^2.

Suppose now that we have been given a set of observations $\mathcal{D} \triangleq \{(x_i, y_i)\}_{i=1}^N = (X, y)$. Our reward with these observations is the number of significantly interesting regions $r = \sum_{g=1}^G r_g$ under the posterior distribution, where

$$r_g \triangleq \begin{cases} 1 & \text{if } \Pr(\mathcal{M}_g > \tau \mid \mathcal{D}) > \theta \\ 0 & \text{otherwise.} \end{cases} \qquad (3)$$

Here τ is the minimum for the average value of f in a region to be considered interesting and θ is the minimal test power. For convenience, we define the tail probability:

$$T_g \triangleq \Pr(\mathcal{M}_g > \tau \mid \mathcal{D}). \qquad (4)$$

The total reward expressed with this new notation is:

$$r = \sum_g r_g = \sum_g \mathbf{1}\{T_g > \theta\}. \qquad (5)$$

The goal for an active area search strategy is to decide which points to observe in order to maximize the reward under a sampling budget.

2.1 Gaussian Processes

We now instantiate the above results with Gaussian processes. The goal is to write (3) in an explicit form. The derivation for $\mathcal{M}_g$ recap discussions in [O'Hagan, 1991; Ghahramani and Rasmussen, 2002].

The observations that we have made, $\mathcal{D}$, change our belief about f and $\mathcal{M}_g$ in an explicit way. For convenience, define $V \triangleq (\kappa(X, X) + \sigma_n^2 I)$. Standard Gaussian process inference gives us the posterior distribution on f given $\mathcal{D}$:

$$p(f \mid \mathcal{D}) = \mathcal{GP}(f; \mu_{f|\mathcal{D}}, \kappa_{f|\mathcal{D}}),$$

where

$$\mu_{f|\mathcal{D}}(x) = \kappa(x, X) V^{-1} y; \qquad (6)$$
$$\kappa_{f|\mathcal{D}}(x, x') = \kappa(x, x') - \kappa(x, X) V^{-1} \kappa(X, x'). \qquad (7)$$

The closure property of Gaussian processes under linear transformations also gives us a simple form for the posterior distribution of $\mathcal{M}_g$ given our observations:

$$p(\mathcal{M}_g \mid \mathcal{D}) = \mathcal{N}(\mathcal{M}_g; \alpha_g, \beta_g^2), \qquad (8)$$

where

$$\alpha_g = \mathbb{E}[\mathcal{M}_g] = \int \mu_{f|\mathcal{D}}(x) p_g(x) \, \mathrm{d}x; \qquad (9)$$

$$\beta_g^2 = \mathrm{Var}[\mathcal{M}_g] = \iint \kappa_{f|\mathcal{D}}(x, x') p_g(x) p_g(x') \, \mathrm{d}x \, \mathrm{d}x'. \qquad (10)$$

Of course, to find α_g and β_g, we must again calculate the potentially difficult integrals (9–10). First we pull constants out of the integral:

$$\alpha_g = \int \left(\kappa(x, X) V^{-1} y \right) p_g(x) \, \mathrm{d}x$$
$$= \left[\int \kappa(x, X) p_g(x) \, \mathrm{d}x \right] V^{-1} y.$$

We define a vector ω_g element-wise as:

$$\omega_{i,g} \triangleq \int \kappa(x, x_i) p_g(x) \, \mathrm{d}x. \qquad (11)$$

Now $\alpha_g = \omega_g^\top V^{-1} y$. If we can evaluate (11) for our choice of κ and $p(x)$, then we will have a simple analytic expression for the expected value of $\mathcal{M}_g$.

Given a closed-form expression for ω_g, we can nearly additionally work out an analogous expression for the variance β_g^2. A similar calculation as above gives us

$$\beta_g^2 = Z_g - \omega_g^\top V^{-1} \omega_g, \qquad (12)$$

where we have defined

$$Z_g = \iint \kappa(x, x') p_g(x) p_g(x') \, \mathrm{d}x \, \mathrm{d}x'. \qquad (13)$$

Finally, the reward associated with region g can be expressed by comparing the tail probability,

$$T_g = \Phi\left(\frac{\alpha_g - \tau}{\beta_g} \right), \qquad (14)$$

where Φ is the standard normal cumulative density function, against the predefined probability threshold θ, complying with (5).

3 EXPECTED REWARD

To decide which point(s) to observe next, we apply expected reward maximization (ERM) in a one-step look-ahead fashion. ERM infers a posterior distribution of the function value at every candidate location and builds statistics conditioned on every draw from this distribution. Suppose the data already collected is $\mathcal{D} = (X, y)$; the steps to evaluate the expected reward for any new available candidate location x_* are:

1. Infer the conditional distribution of $y_* \mid \mathcal{D}, x_*$. We put tildes on top of quantities updated with one possible instantiation of y_* as:

$$\tilde{X} \triangleq (X^\top, x_*)^\top, \; \tilde{y} \triangleq (y^\top, y_*)^\top, \; \tilde{\mathcal{D}} \triangleq (\tilde{X}, \tilde{y}),$$

   which serve as the bases for the look-ahead.

2. Find the distribution of $\mathcal{M}_g \mid \tilde{\mathcal{D}}$ conditioned on every possible draw of the unknown y_*. Further, compute the tail probability, which is implicitly dependent on y_*, as:

$$\tilde{T}_g \triangleq \Pr\big[\mathcal{M}_g > \tau \mid \tilde{D}\big], \tag{15}$$

   and the reward, also dependent on y_*, as:

$$\tilde{r}_g \triangleq 1\Big\{\Pr\big[\mathcal{M}_g > \tau \mid \tilde{D}\big] > \theta\Big\}. \tag{16}$$

3. Marginalize out the unknown y_*. The expected reward over all regions is to be maximized:

$$\max_{x_*} u(x_*; G, \mathcal{D}) \triangleq \sum_g u_g(x_*; \mathcal{D}) \triangleq \sum_g \mathbb{E}(\tilde{r}_g)$$
$$= \sum_g \Pr\{\tilde{T}_g > \theta\} \tag{17}$$

Different terms in the above pipeline for one region can be visualized in Figure 1. For multiple regions, imagine multiple plots like Figure 1 where $\mathcal{M}_g$ and $\tilde{T}_g$ become different, yet y_* can be shared.

3.1 Active Area Search on GPs

We again instantiate the above discussion for our concrete example where the involvement of GP prior becomes explicit. In the end, we will have a nice analytical form for every addend of the summation in (17).

To express the distribution of $\mathcal{M}_g \mid \tilde{\mathcal{D}}$ and thus (15–17), we consider what happens to α_g and β_g given an additional observation, y_*, at a point[3] $x_* \in \mathbb{R}^d$. Define $\tilde{\alpha}_g$ and $\tilde{\beta}_g$ to be the updated mean and variance such that

$$\mathcal{M}_g \mid \tilde{\mathcal{D}} \sim \mathcal{N}\big(\tilde{\alpha}_g, \; \tilde{\beta}_g\big), \tag{18}$$

---

[3]Although most applications would consider only adding one point at a time, the expressions also work with multiple updates in a minibatch, by substituting (X_*, y_*) properly.
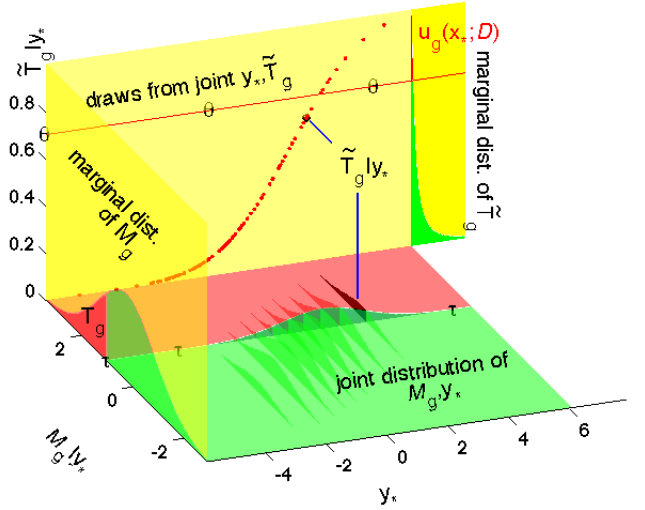

Figure 1: Lookahead variables for one region, g. Given $\mathcal{D}, x_*$ (omitted in figure), y_* is the possible low-level response, $(\mathcal{M}_g, \tilde{T}_g \mid y_*)$ are statistics for g, and $u_g(x_*; \mathcal{D})$ is the expected reward of candidate x_* in g.

which yields $\tilde{T}_g = \Phi\Big(\frac{\tilde{\alpha}_g - \tau}{\tilde{\beta}_g}\Big)$, and thus $\tilde{\alpha}_g$ and $\tilde{\beta}_g$ summarize the randomness of y_*.

Additionally, we define updates to the GP covariance matrix, $\tilde{V} \triangleq \big(\kappa(\tilde{X}, \tilde{X}) + \sigma_n^2 I\big)$, the region integral of the kernel functions, $\tilde{\omega}_g = \big(\omega_g^\top, \omega_{*,g}\big)^\top$, and a new vector to relate the two, $\tilde{\eta}_g \triangleq \tilde{V}^{-1}\tilde{\omega}_g$, which we further decompose into two parts: the first n entries $(\tilde{\eta}_{n,g})$, and the last entry $(\tilde{\eta}_{*,g})$.

It is helpful to define short-hand notions for the GP posterior inference (6) and (7), as $\kappa_*^\top \triangleq \kappa(x_*, X)$, $\mu_{*|\mathcal{D}} \triangleq \kappa_*^\top V^{-1} y$, and $V_{*|\mathcal{D}} \triangleq \kappa_{f|\mathcal{D}}(x_*, x_*) + \sigma_n^2$.

Notably, we will see that the variance of $\tilde{\alpha}_g$ is a quantity that is interesting enough to be assigned its own symbol:

$$\tilde{\nu}_g^2 = \operatorname{Var}(\tilde{\alpha}_g). \tag{19}$$

First, we note that the updated variance, $\tilde{\beta}_g^2$, does not depend on y_*, because similar to (12),

$$\tilde{\beta}_g^2 = Z - \tilde{\omega}_g^\top \tilde{V}^{-1} \tilde{\omega}_g. \tag{20}$$

However, $\tilde{\alpha}_g$ does depend on y_*. As before, given y_*, we have

$$\tilde{\alpha}_g = \tilde{\omega}_g^\top \tilde{V}^{-1} \tilde{y},$$

which breaks into two parts according to whether or not each part depends on y_*, as:

$$\tilde{\alpha}_g = \tilde{\eta}_g^\top \tilde{y} = \tilde{\eta}_{n,g}^\top y + \tilde{\eta}_{*,g}^\top y_*. \tag{21}$$

We can see that $\tilde{\alpha}_g$ is simply an affine transformation of y_*. Prior to observing y_*, we have a Gaussian

distribution on y_*:

$$p(y_* \mid \mathcal{D}) \sim \mathcal{N}\big(\mu_{*|\mathcal{D}}, \ V_{*|\mathcal{D}}\big).$$

Therefore, we again apply the closure property of Gaussians under affine transformations to find the marginal distribution of $\tilde{\alpha}_g$:

$$(\tilde{\alpha}_g \mid x_*, \mathcal{D}) \sim \mathcal{N}\big(\tilde{\eta}_{n,g}^\top y + \tilde{\eta}_{*,g}^\top \mu_{*|\mathcal{D}}, \ \tilde{\eta}_{*,g}^\top V_{*|\mathcal{D}} \tilde{\eta}_{*,g}\big). \quad (22)$$

Next, we look for simplifications of (20) and (22). We exploit the law of total probability:

$$\begin{aligned}
\big(\mathcal{M}_g \mid \mathcal{D}, x_*\big) &= \mathbb{E}\big[\big(\mathcal{M}_g \mid \tilde{\mathcal{D}}\big) \mid \mathcal{D}, x_*\big] \\
&= \mathbb{E}\Big[\big(\mathcal{M}_g \mid \tilde{\alpha}_g, \tilde{\beta}_g^2\big) \mid \mathcal{D}, x_*\Big] \quad (23)
\end{aligned}$$

where the last expression is a Gaussian variable with mean $\mathbb{E}(\tilde{\alpha}_g)$ and variance $\mathrm{Var}(\tilde{\alpha}_g) + \tilde{\beta}_g^2$. By (8) and the fact that $\mathcal{M}_g$ is independent to x_* alone, we have:

$$\mathbb{E}(\tilde{\alpha}_g) = \alpha_g, \quad \tilde{\beta}_g^2 = \beta_g^2 - \mathrm{Var}(\tilde{\alpha}_g) = \beta_g^2 - \tilde{\nu}_g^2. \quad (24)$$

Finally, we address $\tilde{\eta}_{*,g}$ in $\tilde{\nu}_g^2 = \mathrm{Var}(\tilde{\alpha}_g)$ by brute force. Notice that the rank-one update of $\tilde{V}^{-1}$ is:

$$\tilde{V}^{-1} = \begin{pmatrix} V^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -V^{-1}\kappa_* \\ 1 \end{pmatrix} V_{*|\mathcal{D}}^{-1} \big(-\kappa_*^\top V^{-1}, 1 \big).$$

We use its last row, $\big(\tilde{V}^{-1}\big)_*^\top$, to compute

$$\tilde{\eta}_{*,g} = \big(\tilde{V}^{-1}\big)_*^\top \tilde{\omega}_g = V_{*|\mathcal{D}}^{-1} \big(\omega_{*,g} - \kappa_*^\top V^{-1} \omega_g \big). \quad (25)$$

As a result,

$$\tilde{\nu}_g^2 = (\omega_{*,g} - \kappa_*^\top V^{-1} \omega_g)^\top V_{*|\mathcal{D}}^{-1} (\omega_{*,g} - \kappa_*^\top V^{-1} \omega_g), \quad (26)$$

and the look-ahead statistics in this section reduce to:

$$p(\mathcal{M}_g \mid \tilde{\mathcal{D}}) = p(\mathcal{M}_g \mid \tilde{\alpha}_g, \tilde{\beta}_g^2) \sim \mathcal{N}(\alpha_g, \beta_g^2 - \tilde{\nu}_g^2). \quad (27)$$

Every addend of (17) becomes

$$u_g(x_*; \mathcal{D}) = \Pr \left\{ \Phi\Big(\frac{\tilde{\alpha}_g - \tau}{\tilde{\beta}_g}\Big) > \theta \right\} \quad (28)$$

$$= \Phi\left(\frac{\alpha_g - \tau - \tilde{\beta}_g \Phi^{-1}(\theta)}{\tilde{\nu}_g} \right), \quad (29)$$

whose summation is the utility of a candidate location.

Remark: The dependency on $\tilde{y}$ is implicit through $\tilde{\alpha}_g$, updated by $\tilde{\alpha}_g = \alpha_g + \tilde{\eta}_{*,g}^\top (y_* - \mu_{*|\mathcal{D}})$.

3.2 Pseudocode

To summarize the above discussions, we proposed an algorithm which sequentially decides sampling locations in a GP that maximizes the expected reward (17). An explicit formulation can be found as a summation of terms in (29). The pseudocode for our algorithm is in Algorithm 1. We call our algorithm *active area search* (AAS) in the experiments. [4]

Algorithm 1 The active area search (AAS) algorithm.

Require: GP model $\mathcal{GP}(\mu, \kappa)$, PDFs $\{p_g\}_{g=1}^G$
   compute Z_g for all $1 \le g \le G$ // (13)
   $X \leftarrow \emptyset; Y \leftarrow \emptyset$
   repeat
     for unqueried x **do**
       compute $\omega_{*,g}$ // (11)
       compute $\tilde{\nu}_g^2$ // (26)
       compute $\tilde{\beta}_g^2$ // (24)
       compute expected reward given x // (29)
     end for
     $x_* \leftarrow \arg\max$ expected reward
     observe y_*
     $X \leftarrow [X; x_*]; Y \leftarrow [Y; y_*]$
   until budget depleted

3.3 Single Region Search

When there is only one predefined region, the problem becomes a Bayesian sequential test problem. The outcome affects the algorithm in only one way, which is to decide whether or not to stop data collection depending on if the confidence margin, $T_g > \theta \mid \mathcal{D}$, is obtained.

For our purposes, $\theta > 0.5$, which implies $\Phi^{-1}(\theta) > 0$. Omit subscript g. Suppose AAS is still running, i.e.,

$$\Phi\left(\frac{\alpha - \tau}{\beta}\right) \le \theta \quad \Leftrightarrow \quad \frac{\alpha - \tau}{\beta \Phi^{-1}(\theta)} \le 1.$$

We observe some properties about the optimal x_* from (29) that can further motivate the active learning strategy we proposed and make connections to experiment designs with Bayesian quadrature in the next section.

Notice that in (29), $\tilde{\nu}$ is the only variable because $\tilde{\beta} = \sqrt{\beta^2 - \tilde{\nu}^2}$. To find the maximum of (29), we take the gradient over $\tilde{\nu}$:

$$\frac{\partial u_g}{\partial \tilde{\nu}} = \frac{\mathcal{N}(\dots)\Phi^{-1}(\theta)}{\tilde{\nu}^2 \sqrt{\beta^2 - \tilde{\nu}^2}} \times \left(\beta^2 - \frac{\sqrt{\beta^2 - \tilde{\nu}^2}(\alpha - \tau)}{\Phi^{-1}(\theta)} \right).$$

For simplicity, define

$$v \triangleq \big(\tilde{\nu}/\beta, \ \sqrt{\beta^2 - \tilde{\nu}^2}/\beta\big)^\top \quad, \quad t \triangleq \frac{\alpha - \tau}{\beta \Phi^{-1}(\theta)} \le 1.$$

Now, the gradient of the expected reward becomes:

$$\frac{\partial u}{\partial \tilde{\nu}} = \mathcal{N}(\dots)\frac{\Phi^{-1}(\theta)}{v_1 v_2} \big(v_1^2 + v_2^2 - t v_2 \big).$$

which is always nonnegative for $t \le 1$, because $v_1^2 + v_2^2 = 1$ and thus $0 < v_2 < 1$. Thus our algorithm turns out to be finding the x_* that maximizes $\tilde{\nu}^2$ via (26).

One way to build intuition with unimodal stationary kernels for this somewhat surprising result is that the

points able to reduce the variance of the integral are also likely to disturb its mean. At these locations, we also have the best chance of "getting lucky" and significantly increasing the posterior mean of $\mathcal{M}$.

3.4 Multiple Region Case

Contrary to the above, in the scenario where there are multiple predefined regions on the domain, observation outcomes play a more significant role in that they affect the next regions to search. Imagine on a GP with unimodal stationary kernels, there are two regions, g_1 and g_2, that are far away from each other and therefore independent. Evaluate (29) for $g_\ell, \ell = 1, 2$ under their respective maximizer $\tilde{\nu}_{g_\ell}$. Under the same conditions, a bigger α_g, which depends on big outcomes, will result in a higher expected reward. However, as the number of net observations increases in a region, β_g will increase and $\tilde{\nu}_g$ will decrease. This implies that after a long failure period, AAS gives up until every other region gets even worse. In general, AAS tends to explore the most promising region at the current time.

Moreover, even if in one region, a location may be chosen because of its influence at neighboring regions. The coupling of regions, which potentially mixes both interesting and boring regions in an unpredictable way, is another factor implicitly related to the outcomes.

4 RELATED PROBLEMS

4.1 Connection to Experiment Design for BQ

As long as the area has not been flagged as significantly interesting, AAS strives to maximize $\tilde{\nu}$, which equivalently minimizes the variance of the integral. This objective collides with experimental designs for BQ. For particular kernels in [Minka, 2000], we should attain similar search locations. This is a surprising connection, but only holds in the single-region case. For mutliple regions, the fact that the latent function is correlated across regions modifies the optimal poilicy.

4.2 Connection to Σ-optimality

Single-region search is also connected to Σ-optimality, as defined by Ma et al. [2013]. The maximization of $\tilde{\nu}$ is similar to the rank-one update in Σ-optimality, if we take $p(x)$ to be the counting measure of unlabeled points $x_u = \{x_{u_1}, \ldots, x_{u_p}\}$, including $x_* \in x_u$ and let the lookahead variable to be the latent variable rather than the observed variable, or $\sigma_{n*} = 0$. More specifically, $\omega_i = \sum_{u_j} \kappa(x_{u_j}, x_i)$, and

$$\omega_* - \omega^\top V^{-1} \kappa_* = \sum_{u_j} \left(\kappa(x_{u_j}, x_*) - \kappa(x_{u_j}, X) V^{-1} \kappa_* \right)$$
$$= \sum_{u_j} \kappa_{f|\mathcal{D}}(x_{u_j}, x_*). \tag{30}$$

On such condition,

$$\max \tilde{\nu}^2 = \sum_{u_j, u_{j'}} \kappa_{f|\mathcal{D}}(x_{u_j}, x_*) \kappa_{f|\mathcal{D}}^{-1}(x_*, x_*) \kappa_{f|\mathcal{D}}(x_*, x_{u_{j'}})$$
$$= \left(\sum_{u_j} \frac{\kappa_{f|\mathcal{D}}(x_{u_j}, x_*)}{\sqrt{\kappa_{f|\mathcal{D}}(x_*, x_*)}} \right)^2.$$

5 EXPERIMENTS

We used a list of simulated experiments to demonstrate properties and performance of AAS. More interestingly, we provide intuition about the behavior of AAS in multi-region cases, which we really care about. (Gateway to the actual codes is available at footnote 4.)

In all simulations, the input space was the 2-dimensional Euclidean space and our function was generated from a GP whose prior mean was constant zero and whose prior covariance was the following isotropic square exponential kernel:

$$\kappa(x, x') = \sigma_f^2 \exp \left\{ -\frac{1}{2\ell^2} (x - x')^\top (x - x') \right\} \tag{31}$$

where σ_f^2 and ℓ were set at different values in different cases to make the simulated problems interesting. Further, actual observations were simulated with additive noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$.

5.1 One Region Synthetic Data

The first demonstration/experiment was performed on a 2-dimensional unit square which contains only one region. The parameters used to generate the observations in (31) are $\ell = 0.33, \sigma_f^2 = 1/(2\pi\ell^2) = 1.21^2, \sigma^2 = 0.1^2$. We purposefully made the problem difficult, so that AAS can run for a longer time period, by keeping the *a priori* variance of the integral over the region small, only roughly $Z = 0.737^2$. As a result, the region is not guaranteed to have high average values with high probabilities. We kept sampling function values on a 33×33 dense grid until the average value in the unit square region is greater than the threshold $\tau = 1$. AAS is expected to sequentially sample observations until it believes that the regional average is greater than τ with probability at least $\theta = 0.8$.

Figure 2 (a) visualizes the sampling locations determined by AAS in a sequential order. After these updates, the posterior marginal bandwidth of every point is shown in (b) and the gray mesh at level 1.0 serves as a reference showing that the integral of the function, under posterior distribution, has high possibility to be greater than the threshold. The behavior of AAS is consistent with our analysis in Section 3.3. Before the algorithm terminates when it verifies that the region is significantly interesting, AAS explores locations which yield the maximal possible decrease of the variance

(a) sampling location
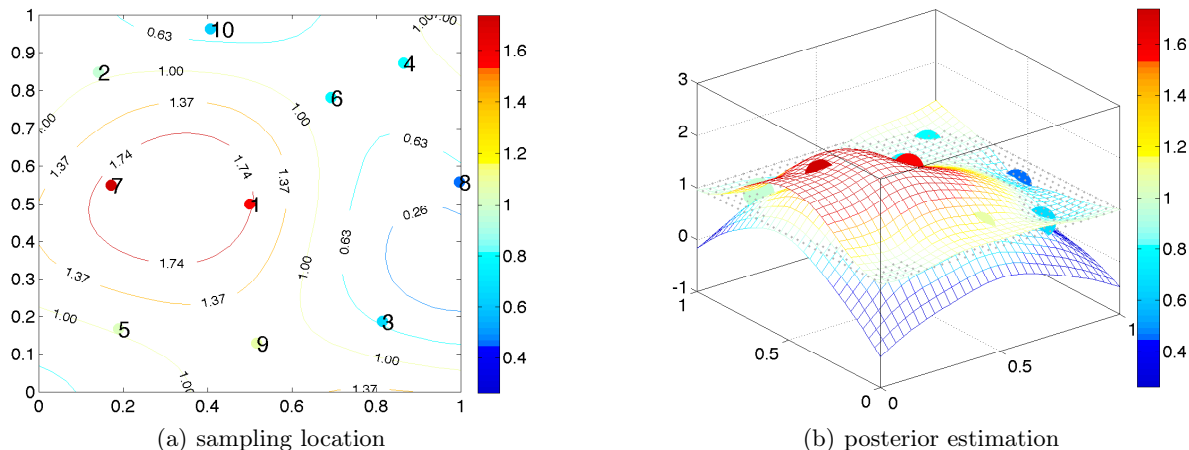

(b) posterior estimation

Figure 2: One region search. Samples are selected in hope that with posterior distributions, the integral over the entire unit square is greater than 1 with probability at least 0.8.

of the integral once sampled, similar to experimental designs in BQ. The intuition is that function values at these points are usually unexplored and may become the best bet to attain a reward.

5.2 Multi-Region Synthetic Data

In this experiment, we simulated random GPs on a 2-D space which is externally split into 10×10 unit square regions. The goal was to find as many interesting regions as possible. Similar to before, a region may be flagged and rewarded if the posterior average function value on this region is greater than $\tau = 1$ with probability at least $\theta = 0.8$.

To allow interactions between regions, we chose a larger length scale for the prior GP.[5] The parameters selected are $\ell = 1, \sigma_f^2 = 1, \sigma_n^2 = 0.1^2$. The prior variance of the integral over any region is $Z = 0.924^2$ (roughly 14% regions are interesting). An illustration is in Figure 3(a), where the color of a region indicates the average function value in that region. Level sets of the function value are also plotted in (a).

The rest of Figure 3 compare the following algorithms

- **Active area search** (AAS): Our proposed method.

- **Level set estimation** (LSE): Gotovos et al. [2013] proposed this theoretically justified algorithm for level set estimations, which is to determine the region in the input space where the function value is close to h. We hope that by finding level sets for $h = \tau$ and recognizing even higher/lower regions, interesting regions may be discovered. Several other parameters

[5]In reality, training can be done offline with pilot data. We usually match the order of region diameter and GP length scale when designing regions for preliminary real-world experiments.

were set as $\beta_t^{1/2} = 3, \varepsilon = 0.1$. (The original paper also set β_t fixed and broke theoretical guarantees in experiments.)

- **Uncertainty sampling** (UNC): Seo et al. [2000] used UNC to map the function value over the entire input space. UNC explores locations that have high marginal variance in the posterior distribution. The samples are sparse but blind to outcomes.

- **Random sampling** (RAND) serves as a baseline. It picks locations at random.

From these plots, we can see that AAS samples locations that are both sparse yet concentrate in regions which are more likely to have high average. It favors points on the boundary of multiple regions. It also explores new locations reasonably. The superiority of AAS in interesting region discovery is obvious.

LSE gives the second-best performance. While searching for level sets, LSE can identify positive regions inside. However, LSE is not aimed for this problem and thus it is hard to pin down which threshold and tolerance to ask for in LSE. Further, LSE may be too wasteful to precisely map the level set, and the observations that LSE makes may not lead to discovery of interesting regions. Finally, LSE may sometimes be pessimistic because of its theoretical guarantees and is sensitive to boundaries.

Finally UNC and RAND are the worst because they are generic and unspecific to the objective.

5.3 Repeated Experiments

We repeated our last experiment for 10 times with different functions generated through the same parameters. We report recall in Figure 4. Precision is a

(a) ground truth   (b) estimate from AAS   (c) estimate from LSE

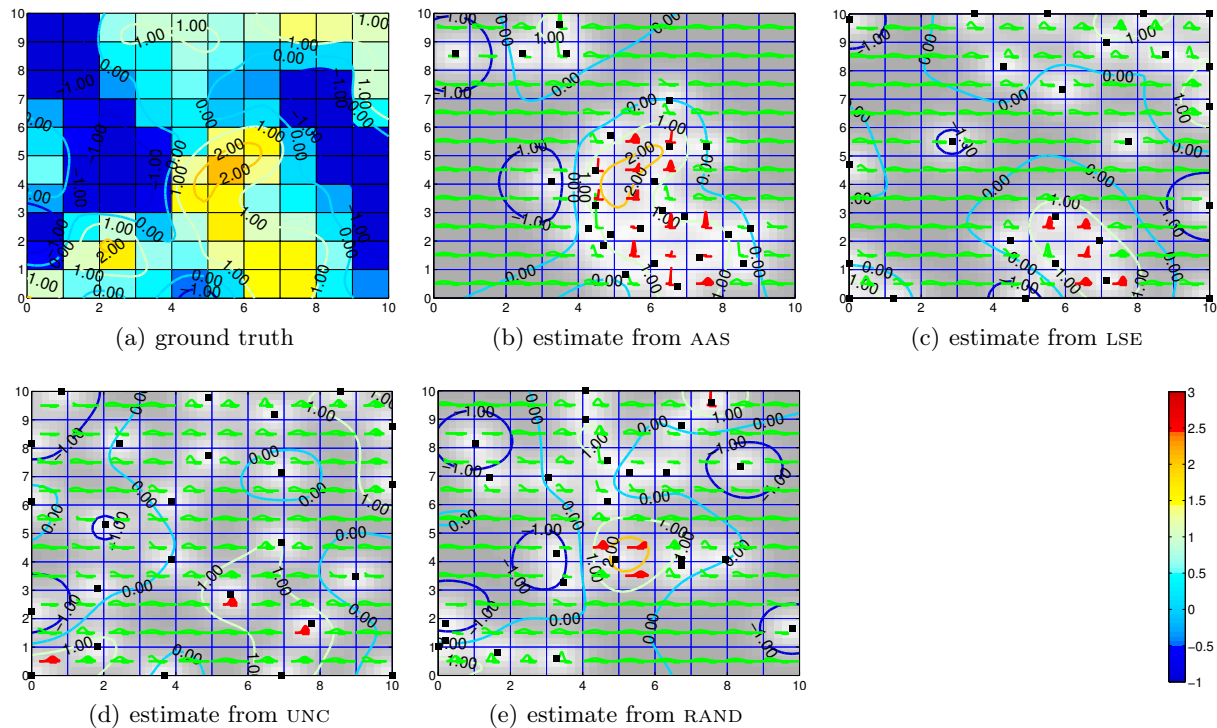(d) estimate from UNC   (e) estimate from RAND

Figure 3: Multi-region. Shared color bar. (a) shows both function values and region averages. (b-e) show the first 25 locations sampled by different strategies (black dots). Gray scale indicate marginal variance. Red/green curves in region centers show the posterior tail distribution of the region averages. Red regions are reported.

function of θ which is the same in all experiments so it is not reported. The curves indicate the average percent of positive regions reported given different query budgets. Standard error of the average is also reported.
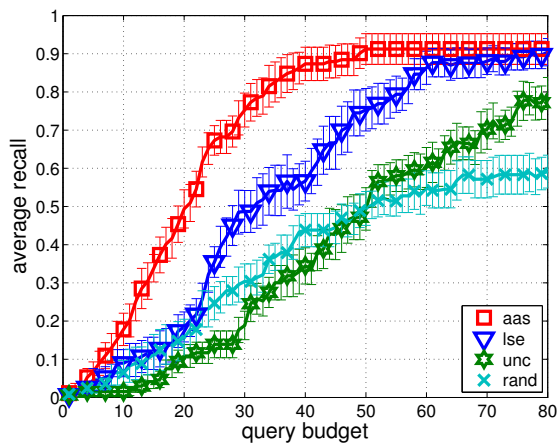

Figure 4: Repeated experiments on 10×10 regions

Figure 4 shows that AAS outperformed other methods by a large margin. With 20 observations, AAS was able to discover half of the interesting regions. Notice in Figure 3, with 25 points, most parts of the function space remain gray even for UNC. The success of AAS mainly attributes to its relevancy to the objective.

LSE performed second best, about 60% as efficient as

AAS. It can be observed from Figure 3 that LSE also biases towards areas near interesting regions. In contrast, neither UNC or RAND utilize sampling budgets efficiently. RAND is slightly better in the beginning because of its randomness yet UNC improves towards the end because it avoids the "coupon collector's problem."

6   CONCLUSION

We defined a new problem, *active area search,* wherein we wish to identify regions in a continuous space with large average function value. In comparison to typical active learning objectives, this setting is somewhat unusual in that we cannot observe the labels directly. Instead, we must infer the labels from observations of a continuous ancillary function. Our approach is to model the function using a Gaussian process and use Bayesian quadrature to infer its average value on the regions of interest. With this setup, we were able to derive a simple myopic expected reward maximization strategy for the active area search problem. Empirically, our algorithm identifies positive regions much faster than related previously proposed approaches.

An interesting future question is to apply more complex reward criteria, e.g., classifiers based on the distribution of outcomes in a region. We also hope to find theoretical bounds in the current setup and discuss path planning.

References

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

B. Bryan, J. Schneider, R. C. Nichol, C. J. Miller, C. R. Genovese, and L. Wasserman. Active learning for identifying function threshold boundaries. NIPS, 2005.

R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. P. Mann. Bayesian optimal active search and surveying. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, pages 1239–1246, Madison, WI, USA, 2012. Omnipress.

Z. Ghahramani and C. E. Rasmussen. Bayesian Monte Carlo. In *Advances in neural information processing systems*, pages 489–496, 2002.

A. Gotovos, N. Casati, G. Hitz, and A. Krause. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

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

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.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

Y. Ma, R. Garnett, and J. Schneider. Σ-optimality for active learning on Gaussian random fields. In *Advances in Neural Information Processing Systems 26*, 2013. To Appear.

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

A. O'Hagan. Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991.

M. A. Osborne, R. Garnett, and S. J. Roberts. Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15, 2009.

C. E. Rasmussen. Gaussian processes for machine learning. In *Adaptive Computation and Machine Learning*. The MIT Press, 2006.

S. Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: active data selection and test point rejection. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 241–246 vol.3, 2000.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *Proceedings of International Conference on Machine Learning*, pages 1015–1022, 2010.

M. Tesch, J. Schneider, and H. Choset. Using response surfaces and expected improvement to optimize snake robot gait parameters. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1069–1074. IEEE, 2011.

A. Valada, C. Tomaszewski, B. Kannan, P. Velagapudi, G. Kantor, and P. Scerri. An intelligent approach to hysteresis compensation while sampling using a fleet of autonomous watercraft. In C.-Y. Su, S. Rakheja, and H. Liu, editors, *Intelligent Robotics and Applications*, volume 7507 of *Lecture Notes in Computer Science*, pages 472–485. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33514-3.