Active Learning for Level Set Estimation *(long version)*

Alkis Gotovos
ETH Zurich

Nathalie Casati
ETH Zurich &
IBM Research – Zurich

Gregory Hitz
ETH Zurich

Andreas Krause
ETH Zurich

Abstract

Many information gathering problems require determining the set of points, for which an unknown function takes value above or below some given threshold level. We formalize this task as a classification problem with sequential measurements, where the unknown function is modeled as a sample from a Gaussian process (GP). We propose LSE, an algorithm that guides both sampling and classification based on GP-derived confidence bounds, and provide theoretical guarantees about its sample complexity. Furthermore, we extend LSE and its theory to two more natural settings: (1) where the threshold level is implicitly defined as a percentage of the (unknown) maximum of the target function and (2) where samples are selected in batches. We evaluate the effectiveness of our proposed methods on two problems of practical interest, namely autonomous monitoring of algal populations in a lake environment and geolocating network latency.

1 Introduction

Many information gathering problems require accurately determining the regions where the value of some unknown function lies above or below a given threshold level. Moreover, evaluating the function is usually a costly procedure and the measurements returned are noisy.

As a concrete example of such an application, consider the task of monitoring a lake environment for algal bloom, a phenomenon that is potentially harmful to other organisms of the ecosystem. One way to accomplish this, is by determining the regions of the lake where the levels of algae-produced chlorophyll are above some threshold value determined by field experts. These regions can be estimated by sampling at various locations of the lake using a mobile sensing device. However, each measurement is costly in terms of time and sensor battery power, therefore the sampling locations have to be picked carefully, in order to reduce the total number of measurements required.

Other example applications in the context of environmental monitoring [Rahimi *et al.*, 2004] include estimating level sets of quantities such as solar radiation, humidity, etc., and determining the extent of hazardous phenomena, e.g. air pollution or oil spills [Galland *et al.*, 2004]. In a different category are applications that consist in determining the subset of a parameter space that represents "acceptable" hypotheses [Bryan *et al.*, 2005] or designs [Ramakrishnan *et al.*, 2005].

We consider the problem of estimating some function level set in a sequential decision setting, where, at each time step, the next sampling location is to be selected given all previous measurements. For solving this problem, we propose the Level Set Estimation (LSE) algorithm, which utilizes Gaussian processes [Rasmussen and Williams, 2006] to model the target function and exploits its inferred confidence bounds to drive the selection process. We also provide an information-theoretic bound on the number of measurements needed to achieve a certain accuracy, when the underlying function is sampled from a Gaussian process.

Furthermore, we extend the LSE algorithm to two more settings that naturally arise in practical applications. In the first setting, we do not a priori have a specific threshold level at our disposal, but would still like to perform level set estimation with respect to an *implicit* level that is expressed as a percentage of the function maximum. In the second setting, we want to select at each step a *batch* of next samples. A reason for doing so is that, in problems such as the lake sensing example outlined above, apart from the cost of actually making each measurement, we also have to take into account the cost incurred by traveling from one sampling location to the next. Traveling costs can be dramatically reduced, if we plan ahead by selecting multiple points at a time. Another reason is that some problems allow for running multiple function evaluations in parallel, in which case selecting batches of points can lead to a significant increase in sampling throughput.

Related work. Previous work on level set [Dantu and Sukhatme, 2007; Srinivasan *et al.*, 2008] and boundary [Singh *et al.*, 2006] estimation and tracking in the context of mobile sensor networks has primarily focused on controlling the movement and communication of sensor nodes, without giving much attention to individual sampling locations and the choice thereof.

In contrast, we consider the problem of level set estimation in the setting of *pool-based active learning* [Settles, 2009], where we need to make sequential decisions by choosing sampling locations from a given set. For this problem, Bryan *et al.* [2005] have proposed the *straddle* heuristic, which selects where to sample by trading off uncertainty and proximity to the desired threshold level, both estimated using GPs. However, no theoretical justification has been given for the use of straddle, neither for its extension to composite functions [Bryan and Schneider, 2008]. Garnett *et al.* [2012] consider the problem of *active search*, which is also about sequential sampling from a domain of two (or more) classes (in our case the super- and sublevel sets). In contrast to our goal of detecting the class boundaries, however, their goal is to sample as many points as possible from one of the classes.

In the setting of multi-armed bandit optimization, which is similar to ours in terms of sequential sampling, but different in terms of objectives, GPs have been used both for modeling, as well as for sample selection [Brochu *et al.*, 2010]. In particular, the GP-UCB algorithm makes use of GP-inferred upper confidence bounds for selecting samples and has been shown to achieve sublinear regret [Srinivas *et al.*, 2010]. An extension of GP-UCB to the multi-objective optimization problem has been proposed by Zuluaga *et al.* [2013], who use a similar GP-based classification scheme to ours to classify points as being Pareto-optimal or not.

Existing approaches for performing multiple evaluations in parallel in the context of GP optimization, include *simulation matching* [Azimi *et al.*, 2010], which combines GP modeling with Monte-Carlo simulations, and the GP-BUCB [Desautels *et al.*, 2012] algorithm, which obtains similar regret bounds to GP-UCB, and from which we borrow the main idea for performing batch sample selection.

To our knowledge, there has been no previous work on actively estimating level sets with respect to implicitly defined threshold levels.

Contributions. The main contributions of this paper can be summarized as follows:

- We introduce the LSE algorithm for sequentially estimating level sets of unknown functions and also extend it to select samples in batches.
- We consider for the first time the problem of estimating level sets under implicitly defined threshold levels and propose an extension of LSE for this problem.
- We prove theoretical convergence bounds for LSE and its two extensions when the target function is sampled from a known GP.
- We evaluate LSE and its extensions on two real-world datasets and show that they are competitive with the state-of-the-art.

2 Background and Problem Statement

Given a function $f : D \to \mathbb{R}$, where D is a finite subset of $\mathbb{R}^d$, and a threshold level $h \in \mathbb{R}$, we define the *level set estimation problem* as the problem of classifying every point $\boldsymbol{x} \in D$ into a *superlevel set* $H = \{\boldsymbol{x} \in D \mid f(\boldsymbol{x}) > h\}$ and a *sublevel set* $L = \{\boldsymbol{x} \in D \mid f(\boldsymbol{x}) \leq h\}$.

In the *strictly sequential* setting, at each step $t \geq 1$, we select a point $\boldsymbol{x}_t \in D$ to be evaluated and obtain a noisy measurement $y_t = f(\boldsymbol{x}_t) + n_t$. In the *batch* setting we select B points at a time and only obtain the resulting measurements after all of the B points have been selected.

When an explicit level is not available, we can define an *implicit threshold level* with respect to the function maximum in either absolute or relative terms. We use the relative definition in our exposition with $h = \omega \max_{\boldsymbol{x} \in D} f(\boldsymbol{x})$ and $\omega \in (0, 1)$.

Gaussian processes. Without any assumptions about the function f, attempting to estimate level sets from few samples is a hopeless endeavor. Modeling f as a sample from a Gaussian process (GP) provides an elegant way for specifying properties of the function in a nonparametric fashion. A GP is defined as a collection of random variables, any finite subset of which is distributed according to a multivariate Gaussian in a consistent way [Rasmussen and Williams, 2006]. A GP is denoted as $\mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$ and is completely specified by its mean function $\mu(\boldsymbol{x})$, which can be assumed to be zero w.l.o.g., and its covariance function or kernel $k(\boldsymbol{x}, \boldsymbol{x}')$, which encodes smoothness properties of functions sampled from the GP.

Assuming a GP prior $\mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$ over f and given t noisy measurements $\boldsymbol{y_t} = [y_1, \ldots, y_t]^T$ for points in $A_t = \{x_1, \ldots, x_t\}$, where $y_i = f(\boldsymbol{x}_i) + n_i$ and $n_i \sim \mathcal{N}(0, \sigma^2)$ (Gaussian i.i.d. noise) for $i = 1, \ldots, t$, the posterior over f is also a GP and its mean, covariance, and variance functions are given by the following analytic formulae:

$$\mu_t(\boldsymbol{x}) = \boldsymbol{k}_t(\boldsymbol{x})^T \left(\boldsymbol{K}_t + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}_t \tag{1}$$

$$k_t(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_t(\boldsymbol{x})^T \left(\boldsymbol{K}_t + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{k}_t(\boldsymbol{x})$$

$$\sigma_t^2(\boldsymbol{x}) = k_t(\boldsymbol{x}, \boldsymbol{x}), \tag{2}$$

where $\boldsymbol{k}_t(\boldsymbol{x}) = [k(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_t, \boldsymbol{x})]^T$ and $\boldsymbol{K}_t$ is the kernel matrix of already observed points, defined as $\boldsymbol{K}_t = [k(\boldsymbol{x}, \boldsymbol{x}')]_{\boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{A}_t}$.

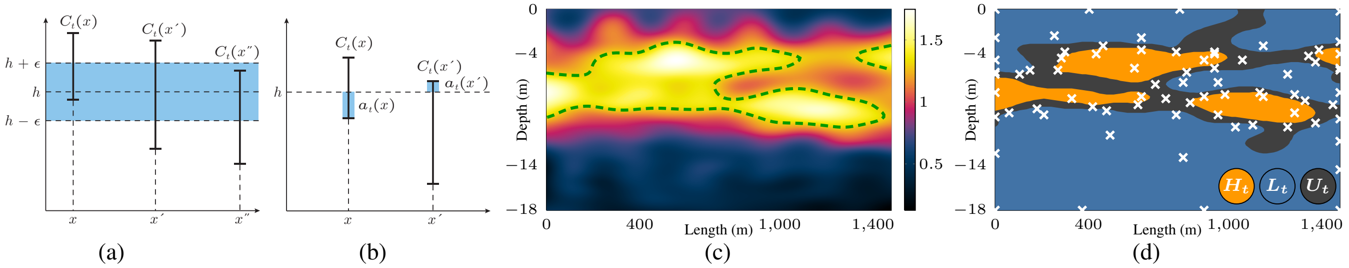
3 The LSE Algorithm

We now present our proposed Level Set Estimation (LSE) algorithm for the strictly sequential setting with explicit thresholds. LSE is similar in spirit to the GP-UCB [Srinivas *et al.*, 2010] bandit optimization algorithm in that it uses a GP to model the underlying function and facilitates the inferred mean and variance of the GP to guide the selection of points to be evaluated.

More concretely, the inferred mean and variance of (1) and (2) can be used to construct a *confidence interval*

$$Q_t(\boldsymbol{x}) = \left[\mu_{t-1}(\boldsymbol{x}) \pm \beta_t^{1/2} \sigma_{t-1}(\boldsymbol{x})\right]$$

for any point $\boldsymbol{x} \in D$, which captures our uncertainty about $f(\boldsymbol{x})$ after having already obtained noisy evaluations of f at points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$. The parameter β_t acts as a scaling factor and its choice is discussed later. The above-defined confidence intervals serve two purposes in our algorithm: first, they allow us to judge whether a point can be classified into the super- or sublevel sets or whether the decision should be deferred until more information is available; second, they guide the sampling process towards points that are likely to be informative with respect to the desired level set.



(a) (b) (c) (d)

Figure 1: (a) Example of the three possible configurations of confidence regions. (b) Ambiguities of two example points. (c) Chlorophyll concentration in relative fluorescence units (RFU) inferred from 2024 measurements on a vertical transect plane of Lake Zurich (the level set at $h = 1.3$ RFU is shown dashed). (d) LSE after $t = 50$ iterations on a grid of 100×100 points: regions of already classified points in orange (H_t) and blue (L_t), of yet unclassified points (U_t) in black, and observed points ($\{\boldsymbol{x}_i\}_{1\leq i\leq t}$) as white marks. Note how the sampling process focuses on the ambiguous regions around the desired level set.

The pseudocode of Algorithm 1 depicts in detail the operation of LSE. Our algorithm maintains a set of yet unclassified points U_t, as well as a superlevel set H_t and a sublevel set L_t, which are updated at each iteration. Furthermore, the algorithm maintains for each $\boldsymbol{x}$ a monotonically decreasing *confidence region* $C_t(\boldsymbol{x})$, which results from intersecting successive confidence intervals, i.e.

$$C_t(\boldsymbol{x}) = \bigcap_{i=1}^{t} Q_i(\boldsymbol{x}) = C_{t-1}(\boldsymbol{x}) \cap Q_t(\boldsymbol{x}).$$

Initially, all points $\boldsymbol{x} \in D$ are unclassified and the confidence regions have infinite range (line 1). At each iteration, the confidence regions of all unclassified points are updated (line 6) and each of these points is either classified into one of H_t or L_t, or is left unclassified (lines 7–10). Then, the next point is selected and evaluated (lines 11–12) and the new GP posterior is computed (line 13). The algorithm terminates when all points in D have been classified, in which case the estimated super- and sublevel sets $\hat{H}$ and $\hat{L}$ are returned (line 16).

Classification. The classification of a point $\boldsymbol{x}$ into H_t or L_t depends on the position of its confidence region with respect to the threshold level h. Intuitively, if all of $C_t(\boldsymbol{x})$ lies above h, then with high probability $f(\boldsymbol{x}) > h$ and $\boldsymbol{x}$ should be moved into H_t. Similarly, if $C_t(\boldsymbol{x})$ lies below h, then $\boldsymbol{x}$ should be moved into L_t. Otherwise, we are still uncertain about the class of $\boldsymbol{x}$, therefore it should, for the moment, remain unclassified. As can be seen in the classification rules of lines 7 and 9, we relax these conditions by introducing an accuracy parameter ϵ, which trades off classification accuracy for sampling cost. The resulting classification scheme is illustrated by the example of Figure 1a, in which point $\boldsymbol{x}$ would be classified into H_t and point $\boldsymbol{x}''$ into L_t, while point $\boldsymbol{x}'$ would remain in U_t as unclassified. Note that LSE uses a monotonic classification scheme, meaning that once a point has been classified, it stays so until the algorithm terminates.

Algorithm 1 The LSE algorithm

Input: sample set D, GP prior ($\mu_0 = 0, k, \sigma_0$), threshold value h, accuracy parameter ϵ
Output: predicted sets $\hat{H}$, $\hat{L}$

```
1:  $H_0 \leftarrow \varnothing, L_0 \leftarrow \varnothing, U_0 \leftarrow D, C_0(\boldsymbol{x}) \leftarrow \mathbb{R}$, for all $\boldsymbol{x} \in D$
2:  $t \leftarrow 1$
3:  while $U_{t-1} \neq \varnothing$ do
4:      $H_t \leftarrow H_{t-1}, L_t \leftarrow L_{t-1}, U_t \leftarrow U_{t-1}$
5:      for all $\boldsymbol{x} \in U_{t-1}$ do
6:          $C_t(\boldsymbol{x}) \leftarrow C_{t-1}(\boldsymbol{x}) \cap Q_t(\boldsymbol{x})$
7:          if $\min(C_t(\boldsymbol{x})) + \epsilon > h$ then
8:              $U_t \leftarrow U_t \setminus \{\boldsymbol{x}\}, H_t \leftarrow H_t \cup \{\boldsymbol{x}\}$
9:          else if $\max(C_t(\boldsymbol{x})) - \epsilon \leq h$ then
10:             $U_t \leftarrow U_t \setminus \{\boldsymbol{x}\}, L_t \leftarrow L_t \cup \{\boldsymbol{x}\}$
11:     $\boldsymbol{x}_t \leftarrow \operatorname{argmax}_{\boldsymbol{x} \in U_t}(a_t(\boldsymbol{x}))$
12:     $y_t \leftarrow f(\boldsymbol{x}_t) + n_t$
13:     Compute $\mu_t(\boldsymbol{x})$ and $\sigma_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in U_t$
14:     $t \leftarrow t + 1$
15: $\hat{H} \leftarrow H_{t-1}, \hat{L} \leftarrow L_{t-1}$
```

Sample selection. For selecting the next point to be evaluated at each iteration, we define the following quantity

$$a_t(\boldsymbol{x}) = \min\{\max(C_t(\boldsymbol{x})) - h, h - \min(C_t(\boldsymbol{x}))\},$$

which we call *ambiguity* and, as it names suggests, quantifies our uncertainty about whether $\boldsymbol{x}$ belongs to H_t or L_t (see Figure 1b). The intuition of sampling at areas of the sample space with large classification uncertainty, expecting to gain more information about the problem at hand when sampling at those areas, manifests itself in LSE by choosing to evaluate at each iteration the point with the largest ambiguity amongst the yet unclassified (see Figures 1c and 1d).

We can make an interesting observation at this point. If we use the confidence intervals $Q_t(\boldsymbol{x})$ instead of the confidence regions $C_t(\boldsymbol{x})$ in the definition of ambiguity, we get

$$\begin{aligned} a'_t(\boldsymbol{x}) &= \min\{\max(Q_t(\boldsymbol{x})) - h,\ h - \min(Q_t(\boldsymbol{x}))\} \\ &= \beta_t^{1/2}\sigma_{t-1}(\boldsymbol{x}) - |\mu_{t-1}(\boldsymbol{x}) - h|. \end{aligned}$$

For $\beta_t^{1/2} = 1.96$, this is identical to the *straddle* [Bryan *et al.*, 2005] selection rule, which can thus be intuitively explained in terms of classification ambiguity.

Theoretical analysis. The convergence analysis of LSE rests on quantifying the complexity of the GP prior for f in information-theoretic terms. The information gain [Cover and Thomas, 2006] about f from observing t noisy measurements $\boldsymbol{y}_t = (y_i)_{1\leq i\leq t}$ is

$$I(\boldsymbol{y}_t; f) = H(\boldsymbol{y}_t) - H(\boldsymbol{y}_t \mid f).$$

Srinivas *et al.* [2010] used the *maximum information gain* over all possible sets of t observations

$$\gamma_t = \max_{\boldsymbol{y}_t} I(\boldsymbol{y}_t; f)$$

for bounding the regret of the GP-UCB algorithm. Even though the problem we consider is different, we can use the same quantity to bound the number of LSE iterations required to achieve a certain classification quality.

To quantify the quality of a solution $(\hat{H}, \hat{L})$ with respect to a single point $\boldsymbol{x} \in D$ we use the misclassification loss

$$\ell_h(\boldsymbol{x}) = \begin{cases} \max\{0, f(\boldsymbol{x}) - h\}, & \text{if } \boldsymbol{x} \in \hat{L} \\ \max\{0, h - f(\boldsymbol{x})\}, & \text{if } \boldsymbol{x} \in \hat{H} \end{cases} .$$

The overall quality of a solution can then be judged by the largest misclassification loss among all points in the sample space, i.e. $\max_{\boldsymbol{x} \in D} \ell_h(\boldsymbol{x})$. Intuitively, having a solution with $\max_{\boldsymbol{x} \in D} \ell_h(\boldsymbol{x}) \leq \epsilon$ means that every point $\boldsymbol{x}$ is correctly classified with respect to a threshold level that deviates by at most ϵ from the true level h; we call such a solution ϵ-*accurate*. The following theorem establishes a convergence bound for LSE in terms of γ_t for any given accuracy ϵ.

Theorem 1. *For any $h \in \mathbb{R}$, $\delta \in (0, 1)$, and $\epsilon > 0$, if $\beta_t = 2 \log(|D|\pi^2 t^2/(6\delta))$, LSE terminates after at most T iterations, where T is the smallest positive integer satisfying*

$$\frac{T}{\beta_T \gamma_T} \geq \frac{C_1}{4\epsilon^2},$$

where $C_1 = 8/\log(1 + \sigma^{-2})$.

Furthermore, with probability at least $1 - \delta$, the algorithm returns an ϵ-accurate solution, that is

$$\Pr\left\{\max_{\boldsymbol{x} \in D} \ell_h(\boldsymbol{x}) \leq \epsilon\right\} \geq 1 - \delta.$$

The proof of Theorem 1 can be outlined as follows. The choice of β_t guarantees the inclusion of $f(\boldsymbol{x})$ in the confidence regions $C_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in D$. From the monotonic classification scheme and the maximum ambiguity selection rule, it follows that the ambiguities of the selected points, $a_t(\boldsymbol{x}_t)$, are decreasing with t and, using γ_t, they can be shown to decrease as $\mathcal{O}((\frac{\beta_t \gamma_t}{t})^{\frac{1}{2}})$. Finally, the classification rules of LSE connect the number of iterations to the accuracy parameter ϵ and guarantee an ϵ-accurate solution with high probability.

Note that bounds on γ_T have been established for commonly used kernels [Srinivas *et al.*, 2010] and can be plugged into Theorem 1 to obtain concrete bounds on T. For example, for a d-dimensional sample space and a squared exponential GP kernel, $\gamma_T = \mathcal{O}((\log T)^{d+1})$, and the expression in the bound of Theorem 1 becomes $T/(\log T)^{d+2} \geq C/\epsilon^2$, where, for any given sample space and kernel hyperparameters, C depends only on the choice of δ.

4 Extensions

We now extend LSE to deal with the two problem variants introduced in Section 2. We highlight the key differences in the extended versions of the algorithm and the resulting implications about the convergence bound of Theorem 1.

4.1 Implicit Threshold Level

The substitution of the explicit threshold level by an implicit level $h = \omega \max_{\boldsymbol{x} \in D} f(\boldsymbol{x})$ requires modifying the classification rules as well as the selection rule of LSE, which results in what we call the LSE$_{\text{imp}}$ algorithm.

Algorithm 2 The LSE$_{\text{imp}}$ extension

```
Input: sample set $D$, GP prior ($\mu_0 = 0$, $k$, $\sigma_0$),
       threshold ratio $\omega$, accuracy parameter $\epsilon$
Output: predicted sets $\hat{H}$, $\hat{L}$
// initialization as in LSE; in addition: $Z_0 \leftarrow D$
while $U_{t-1} \neq \varnothing$ do
    // new set definitions as in LSE
    for all $\boldsymbol{x} \in U_{t-1}$ do
        $C_t(\boldsymbol{x}) \leftarrow C_{t-1}(\boldsymbol{x}) \cap Q_t(\boldsymbol{x})$
        $h_t^{opt} \leftarrow \omega \max_{\boldsymbol{x} \in Z_{t-1}} \max(C_t(\boldsymbol{x}))$
        $f_t^{pes} \leftarrow \max_{\boldsymbol{x} \in Z_{t-1}} \min(C_t(\boldsymbol{x}))$, $h_t^{pes} \leftarrow \omega f_{pes}$
        if $\min(C_t(\boldsymbol{x})) + \epsilon \geq h_t^{opt}$ then
            $U_t \leftarrow U_t \setminus \{\boldsymbol{x}\}$
            if $\max(C_t(\boldsymbol{x})) < f_t^{pes}$ then $H_t \leftarrow H_t \cup \{\boldsymbol{x}\}$
            else $M_t^H \leftarrow M_t^H \cup \{\boldsymbol{x}\}$
        else if $\max(C_t(\boldsymbol{x})) - \epsilon \leq h_t^{pes}$ then
            $U_t \leftarrow U_t \setminus \{\boldsymbol{x}\}$
            if $\max(C_t(\boldsymbol{x})) < f_t^{pes}$ then $L_t \leftarrow L_t \cup \{\boldsymbol{x}\}$
            else $M_t^L \leftarrow M_t^L \cup \{\boldsymbol{x}\}$
    $Z_t \leftarrow U_t \cup M_t^H \cup M_t^L$
    $\boldsymbol{x}_t \leftarrow \text{argmax}_{\boldsymbol{x} \in Z_t}(w_t(\boldsymbol{x}))$
    // GP inference as in LSE
$\hat{H} \leftarrow H_{t-1} \cup M_{t-1}^H$, $\hat{L} \leftarrow L_{t-1} \cup M_{t-1}^L$
```

Since h is now an estimated quantity that depends on the function maximum, we have to take into account the uncertainty associated with it when making classification decisions. Concretely, we can obtain an *optimistic estimate* of the function maximum as $f_t^{opt} = \max_{\boldsymbol{x} \in U_t} \max(C_t(\boldsymbol{x}))$ and, analogously, a *pessimistic estimate* as $f_t^{pes} = \max_{\boldsymbol{x} \in U_t} \min(C_t(\boldsymbol{x}))$. The corresponding estimates of the implicit level are defined as $h_t^{opt} = \omega f_t^{opt}$ and $h_t^{pes} = \omega f_t^{pes}$, and can be used in a similar classification scheme to that of LSE. However, for the above estimates to be correct, we have to ensure that U_t always contains all points that could be maximizers of f, i.e. all points that satisfy $\max(C_t(\boldsymbol{x})) \geq f_t^{pes}$. For that purpose, points that should be classified, but are still possible function maximizers according to the above inequality, are kept in two sets M_t^H and M_t^L respectively, while a new set $Z_t = U_t \cup M_t^H \cup M_t^L$ is used in place of U_t to obtain the optimistic and pessimistic estimates h_t^{opt} and h_t^{pes}. The resulting classification rules are shown in Algorithm 2, where the conditions are again relaxed by an accuracy parameter ϵ.

In contrast to LSE, which solely focuses on sampling the most ambiguous points, in LSE$_{\text{imp}}$ it is also of importance to have a more exploratory sampling policy in order to obtain more accurate estimates f_t^{opt} and f_t^{pes}. To this end, we select at each iteration the point with the largest confidence region *width*, defined as

$$w_t(\boldsymbol{x}) = \max(C_t(\boldsymbol{x})) - \min(C_t(\boldsymbol{x})).$$

If confidence intervals were not intersected, this would be equivalent to maximum variance sampling (within Z_t).

Theoretical analysis. The main challenge in extending the results of Theorem 1 to the implicit threshold level setting, stems from the fact that achieving a certain level of classification accuracy, now also depends on having accurate estimates of h. This translates into appropriately bounding $h_t^{opt} - h_t^{pes}$, while, at the same time, guaranteeing that classification is still being performed correctly (by showing that $h_t^{opt} > h$ and $h_t^{pes} < h$). The following theorem expresses the resulting convergence bound for $\mathsf{LSE}_{\mathsf{imp}}$.

Theorem 2. *For any $\omega \in (0,1)$, $\delta \in (0,1)$, and $\epsilon > 0$, if $\beta_t = 2\log(|D|\pi^2 t^2/(6\delta))$, $\mathsf{LSE}_{\mathsf{imp}}$ terminates after at most T iterations, where T is the smallest positive integer satisfying*

$$\frac{T}{\beta_T \gamma_T} \geq \frac{C_1(1+\omega)^2}{4\epsilon^2},$$

where $C_1 = 8/\log(1+\sigma^{-2})$.

Furthermore, with probability at least $1-\delta$, the algorithm returns an ϵ-accurate solution with respect to the implicit level $h = \omega \max_{\boldsymbol{x}\in D} f(\boldsymbol{x})$, that is

$$\Pr\left\{\max_{\boldsymbol{x}\in D} \ell_h(\boldsymbol{x}) \leq \epsilon\right\} \geq 1-\delta.$$

Note that the sample complexity bound of Theorem 2 is a factor $(1+\omega)^2 \leq 4$ larger than that of Theorem 1, and that $\omega = 0$ actually reduces to an explicit threshold of 0.

4.2 Batch Sample Selection

In the batch setting, the algorithms are only allowed to use the observed values of previous batches when selecting samples for the current batch. A naive way of extending LSE (resp. $\mathsf{LSE}_{\mathsf{imp}}$) to this setting would be to modify the selection rule so that, instead of picking the point with the largest ambiguity (resp. width), it chooses the B highest ranked points. However, this approach tends to select "clusters" of closely located samples with high ambiguities (resp. widths), ignoring the decrease in the estimated variance of a point resulting from sampling another point nearby.

Fortunately, we can handle the above issue by exploiting a key property of GPs, namely that the predictive variances (2) depend only on the selected points $\boldsymbol{x}_t$ and not on the observed values y_t at those points. Therefore, even if we do not have available feedback for each selected point up to iteration t, we can still obtain the following useful confidence intervals

$$Q_t^b(\boldsymbol{x}) = \left[\mu_{\mathrm{fb}[t]}(\boldsymbol{x}) \pm \eta_t^{1/2}\sigma_{t-1}(\boldsymbol{x})\right],$$

which combine the most recent available mean estimate ($\mathrm{fb}[t]$ being the index of the last available observation) with the always up-to-date variance estimate. Confidence regions $C_t^b(\boldsymbol{x})$ are defined as before by intersecting successive confidence intervals and are used without any further changes in the algorithms. However, to guarantee convergence we must compensate for using outdated mean estimates, by employing a more conservative (i.e., larger) scaling parameter η_t compared to β_t, in order to ensure that the resulting confidence regions $C_t^b(\boldsymbol{x})$ still contain $f(\boldsymbol{x})$ with high probability.

5 Experiments

In this section, we evaluate our proposed algorithms on two real-world datasets and compare them to the state-of-the-art. In more detail, the algorithms and their setup are as follows.

LSE/LSE$_{\mathsf{imp}}$: Since the bound of Theorem 1 is fairly conservative, in our experiments we used a constant value of $\beta_t^{1/2} = 3$, which is somewhat smaller than the values suggested by the theorem.

LSE$_{\mathsf{batch}}$/LSE$_{\mathsf{imp\text{-}batch}}$: We used $\eta_t^{1/2} = 4$ and $B = 30$.

STR: The state-of-the-art straddle heuristic, as proposed by Bryan *et al.* [2005], with the selection rule $\boldsymbol{x}_t = \mathrm{argmax}_{\boldsymbol{x}\in D}\,(1.96\sigma_{t-1}(\boldsymbol{x}) - |\mu_{t-1}(\boldsymbol{x}) - h|)$.

STR$_{\mathsf{imp}}$: For the implicit threshold setting, we have defined a variant of the straddle heuristic that uses at each step the implicitly defined threshold level with respect to the maximum of the inferred mean, i.e. $h_t = \omega \max_{\boldsymbol{x}\in D} \mu_{t-1}(\boldsymbol{x})$.

STR$_{\mathsf{rank}}$/STR$_{\mathsf{batch}}$: We have defined two batch versions of the straddle heuristic: $\mathsf{STR}_{\mathsf{rank}}$ selects the $B = 30$ points with the largest straddle score, while $\mathsf{STR}_{\mathsf{batch}}$ follows a similar approach to $\mathsf{LSE}_{\mathsf{batch}}$ by using the selection rule $\boldsymbol{x}_t = \mathrm{argmax}_{\boldsymbol{x}\in D}\left(1.96\sigma_{t-1}(\boldsymbol{x}) - |\mu_{\mathrm{fb}[t]}(\boldsymbol{x}) - h|\right)$.

VAR: The max. variance rule $\boldsymbol{x}_t = \mathrm{argmax}_{\boldsymbol{x}\in D}\,\sigma_{t-1}(\boldsymbol{x})$.

We assess the classification accuracy for all algorithms using the F_1-score, i.e. the harmonic mean of precision and recall, by considering points in the super- and sublevel sets as positives and negatives respectively. Finally, in all evaluations of LSE and its extensions, the accuracy parameter ϵ is chosen to increase exponentially from 2% up to 20% of the maximum value of each dataset.

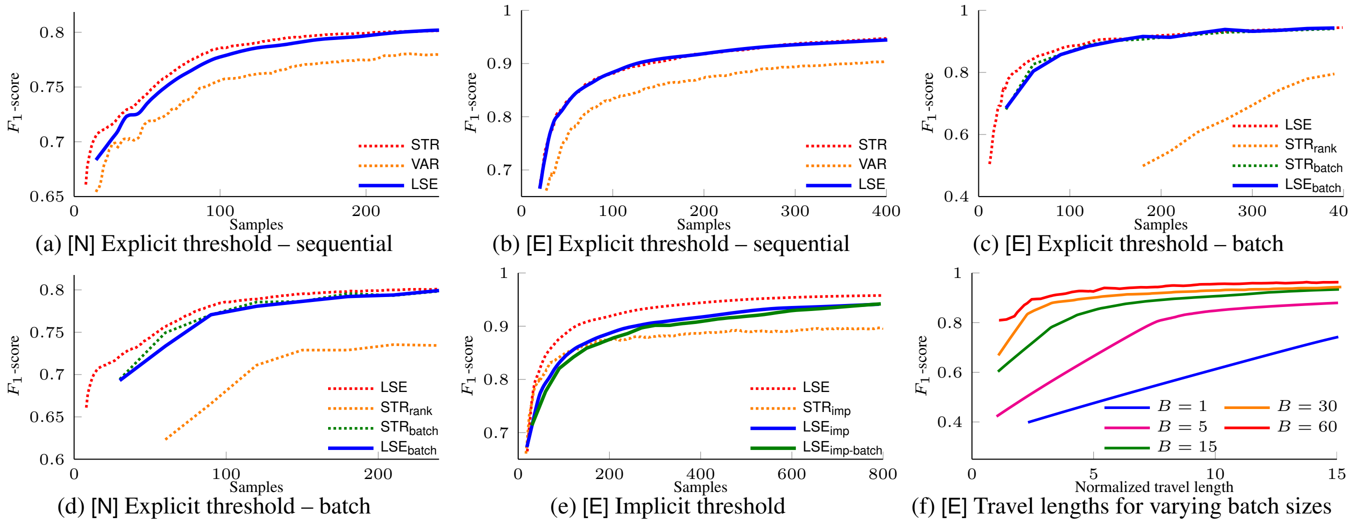
Dataset 1: Network latency. Our first dataset consists of round-trip time (RTT) measurements obtained by "pinging" 1768 servers spread around the world. The sample space consists of the longitude and latitude coordinates of each server, as determined by a commercial geolocation database[1]. Example applications for geographic RTT level set estimation, include monitoring global networks and improving quality of service for applications such as internet telephony or online games. Furthermore, selecting samples in batches results in significant time savings, since sending out and receiving multiple ICMP packets can be virtually performed in parallel.

We used 200 randomly selected measurements to fit suitable hyperparameters for an anisotropic Matérn-5 [Rasmussen and Williams, 2006] kernel by maximum likelihood and the remaining 1568 for evaluation. The threshold level we chose for the experiments was $h = 200$ ms.

Dataset 2: Environmental monitoring. Our second dataset comes from the domain of environmental monitoring of inland waters and consists of 2024 *in situ* measurements of chlorophyll concentration within a vertical transect plane, collected by an autonomous surface vessel in Lake Zurich [Hitz *et al.*, 2012]. Since chlorophyll levels can vary throughout the year, in addition to having a fixed threshold concentration, it can also be useful to be able to detect relative "hotspots" of chlorophyll, i.e. regions of high concentration with respect to the current maximum. Furthermore, selecting batches of points can be used to plan sampling paths and reduce the required traveling distances.

In our evaluation, we used $10,000$ points sampled in a 100×100 grid from the GP posterior that was derived using the 2024 original measurements (see Figure 1c). Again, an

[1] http://www.maxmind.com



(a) [N] Explicit threshold – sequential (b) [E] Explicit threshold – sequential (c) [E] Explicit threshold – batch

(d) [N] Explicit threshold – batch (e) [E] Implicit threshold (f) [E] Travel lengths for varying batch sizes

Figure 2: Results for the network latency [N] and environmental monitoring [E] datasets. (a), (b) LSE is competitive with STR, while both clearly outperform VAR. (c), (d) LSE$_{\text{batch}}$ and STR$_{\text{batch}}$ are only slightly worse than STR, while the naive STR$_{\text{rank}}$ is far inferior. (e) LSE$_{\text{imp}}$ and LSE$_{\text{imp-batch}}$ achieve high accuracy at a slower rate than the explicit threshold algorithms, while STR$_{\text{imp}}$ fails to do so. (f) Using larger batch sizes for planning dramatically reduces the traveled path lengths.

anisotropic Matérn-5 kernel was used and suitable hyperparameters were fitted by maximizing the likelihood of a second available chlorophyll dataset from Lake Zurich. We used an explicit threshold level of $h = 1.3$ RFU and chose $\omega = 0.74$ for the implicit threshold case, so that the resulting implicit level is identical to the explicit one, which enables us to compare the two settings on equal ground. We evaluated the effect of batch sampling on the required traveling distance using an approximate Euclidean TSP solver to create paths connecting each batch of samples selected by LSE$_{\text{batch}}$.

Results and discussion. Figures 2a and 2b compare the performance of the strictly sequential algorithms on the two datasets. In both cases, LSE and STR are comparable in performance, which is expected given the similarity of their selection rules (see Section 3). Although VAR is commonly used for estimating functions over their entire domain, it is clearly outperformed by both algorithms and, thus, deemed unsuitable for the task of level set estimation.

In Figures 2c and 2d we show the performance of the batch algorithms on the two datasets. The LSE$_{\text{batch}}$ and STR$_{\text{batch}}$ algorithms, which use the always up-to-date variance estimates for selecting batches, achieve similar performance. Furthermore, there is only a slight performance penalty when compared to the strictly sequential STR, which can easily be outweighed by the benefits of batch point selection (e.g. in the network latency example, the batch algorithms would have about $B = 30$ times higher throughput). As expected, the STR$_{\text{rank}}$ algorithm, performs significantly worse than the other two batch algorithms, since it selects a lot of redundant samples in areas of high straddle score (cf. Section 4.2).

Figure 2e shows the results of the implicit threshold experiments on the environmental monitoring dataset. The difficulty of estimating the function maximum at the same time as performing classification with respect to the implicit threshold level is manifested in the notably larger sampling cost of LSE$_{\text{imp}}$ required to achieve high accuracy compared to the explicit threshold experiments. As before, the batch version of LSE$_{\text{imp}}$ is only slightly worse that its sequential counterpart. More importantly, the naive STR$_{\text{imp}}$ algorithm completely fails to achieve high accuracy, as it gets stuck with a wrong estimate of the maximum and never recovers, since the straddle rule is not sufficiently exploratory.

Finally, Figure 2f displays the dramatically lower required travel length by using batches of samples for path planning: to achieve an F_1-score of 0.95 using sequential sampling requires more than six times larger traveling distance than planning ahead with $B = 30$ samples per batch.

6 Conclusion

We presented LSE, an algorithm for estimating level sets, which operates based on confidence bounds derived by modeling the target function as a GP. We considered for the first time the challenge of implicitly defined threshold levels and extended LSE to this more complex setting. We also showed how both algorithms can be extended to select samples in batches. In addition, we provided theoretical bounds on the number of iterations required to obtain an ϵ-accurate solution when the target function is sampled from a known GP. The experiments on two real-world applications showed that LSE is competitive with the state-of-the-art, while its extensions are successful in handling the corresponding problem variants and perform significantly better than naive baselines. We believe our results provide an important step towards addressing complex real-world information gathering problems.

Acknowledgments. We would like to thank the Limnological Station of University of Zurich and the Autonomous Systems Lab of ETH Zurich for providing the environmental monitoring datasets. This research was supported in part by SNSF grants 200021 137971 and CR22I2-130023, ERC StG 307036, and a Microsoft Research Faculty Fellowship.

References

[Azimi *et al.*, 2010] Javad Azimi, Alan Fern, and Xiaoli Fern. Batch bayesian optimization via simulation matching. In *Neural Information Processing Systems (NIPS)*, 2010.

[Brochu *et al.*, 2010] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv:1012.2599*, 2010.

[Bryan and Schneider, 2008] Brent Bryan and Jeff Schneider. Actively learning level-sets of composite functions. In *International Conference on Machine Learning (ICML)*, 2008.

[Bryan *et al.*, 2005] Brent Bryan, Jeff Schneider, Robert Nichol, Christopher Miller, Christopher Genovese, and Larry Wasserman. Active learning for identifying function threshold boundaries. In *Neural Information Processing Systems (NIPS)*, 2005.

[Cover and Thomas, 2006] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

[Dantu and Sukhatme, 2007] Karthik Dantu and Gaurav S. Sukhatme. Detecting and tracking level sets of scalar fields using a robotic sensor network. In *International Conference on Robotics and Automation (ICRA)*, 2007.

[Desautels *et al.*, 2012] Thomas Desautels, Andreas Krause, and Joel Burdick. Parallelizing exploration-exploitation tradeoffs with gaussian process bandit optimization. In *International Conference on Machine Learning (ICML)*, 2012.

[Galland *et al.*, 2004] Frédéric Galland, Philippe Rófrógier, and Olivier Germain. Synthetic aperture radar oil spill segmentation by stochastic complexity minimization. *IEEE Geoscience and Remote Sensing Letters (GRSL)*, 2004.

[Garnett *et al.*, 2012] Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard Mann. Bayesian optimal active search and surveying. In *International Conference on Machine Learning (ICML)*, 2012.

[Hitz *et al.*, 2012] Gregory Hitz, François Pomerleau, Marie-Eve Garneau, Cédric Pradalier, Thomas Posch, Jakob Pernthaler, and Roland Y. Siegwart. Autonomous inland water monitoring: Design and application of a surface vessel. *Robotics & Automation Magazine (RAM)*, 2012.

[Rahimi *et al.*, 2004] Mohammad Rahimi, Richard Pon, William J. Kaiser, Gaurav S. Sukhatme, Deborah Estrin, and Mani Srivastava. Adaptive sampling for environmental robotics. In *International Conference on Robotics and Automation (ICRA)*, 2004.

[Ramakrishnan *et al.*, 2005] Naren Ramakrishnan, Chris Bailey-Kellogg, Satish Tadepalli, and Varun N. Pandey. Gaussian processes for active data mining of spatial aggregates. In *SIAM International Conference on Data Mining (SDM)*, 2005.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[Settles, 2009] Burr Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[Singh *et al.*, 2006] Aarti Singh, Robert Nowak, and Parmesh Ramanathan. Active learning for adaptive mobile sensing networks. In *International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.

[Srinivas *et al.*, 2010] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

[Srinivasan *et al.*, 2008] Sumana Srinivasan, Krithi Ramamritham, and Purushottam Kulkarni. Ace in the hole: Adaptive contour estimation using collaborating mobile sensors. In *International Conference on Information Processing in Sensor Networks (IPSN)*, 2008.

[Zuluaga *et al.*, 2013] Marcela Zuluaga, Andreas Krause, Guillaume Sergent, and Markus Püschel. Active learning for multi-criterion optimization. In *International Conference on Machine Learning (ICML)*, 2013.

A Proof of Theorem 1

Lemma 1. *For any $\delta \in (0,1)$, if $\beta_t = 2\log(|D|\pi_t/\delta)$, where $\sum_{t\geq 1}\pi_t^{-1} = 1$ and $\pi_t > 0$, then the following holds with probability at least $1-\delta$*

$$|f(\boldsymbol{x}) - \mu_{t-1}(\boldsymbol{x})| \leq \beta_t^{1/2}\sigma_{t-1}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in D\ \forall t \geq 1.$$

In particular, we can choose $\pi_t = \pi^2 t^2/6$.

Proof. See Lemma 5.1 in [Srinivas *et al.*, 2010]. □

Corollary 1. *For any $\delta \in (0,1)$ and β_t as above, the following holds with probability at least $1-\delta$*

$$f(\boldsymbol{x}) \in C_t(\boldsymbol{x}),\ \forall \boldsymbol{x} \in D\ \forall t \geq 1.$$

Lemma 2. *The following holds for any $t \geq 1$*

$$a_t(\boldsymbol{x}_t) \leq \beta_t^{1/2}\sigma_{t-1}(\boldsymbol{x}_t).$$

Proof. By the definition of ambiguity

$$\begin{aligned} a_t(\boldsymbol{x}_t) &= \min\{\max(C_t(\boldsymbol{x}_t)) - h, h - \min(C_t(\boldsymbol{x}_t))\} \\ &\leq (\max(C_t(\boldsymbol{x}_t)) - \min(C_t(\boldsymbol{x}_t)))/2 \\ &\leq (\max(Q_t(\boldsymbol{x}_t)) - \min(Q_t(\boldsymbol{x}_t)))/2 \\ &= \beta_t^{1/2}\sigma_{t-1}(\boldsymbol{x}_t). \end{aligned}$$

□

Lemma 3. *While running LSE, $a_t(\boldsymbol{x}_t)$ is nonincreasing in t.*

Proof. From the definition of the confidence region of $\boldsymbol{x}_t$ via successive intersections $C_t(\boldsymbol{x}_t) = C_{t-1}(\boldsymbol{x}_t) \cap Q_t(\boldsymbol{x}_t)$, it follows that

$$\left.\begin{array}{l} \max(C_t(\boldsymbol{x}_t)) \leq \max(C_{t-1}(\boldsymbol{x}_t)) \\ \min(C_t(\boldsymbol{x}_t)) \geq \min(C_{t-1}(\boldsymbol{x}_t)) \end{array}\right\} \Rightarrow a_t(\boldsymbol{x}_t) \leq a_{t-1}(\boldsymbol{x}_t)$$

Furthermore, from the selection rule used in LSE $\boldsymbol{x}_t = \operatorname{argmax}_{\boldsymbol{x}\in U_t}(a_t(\boldsymbol{x}))$ and the monotonicity of U_t ($U_t \subseteq U_{t-1}$), it follows that $a_{t-1}(\boldsymbol{x}_t) \leq a_{t-1}(\boldsymbol{x}_{t-1})$. □

Lemma 4. *Denoting $\boldsymbol{y}_t = (y_i)_{1\leq i\leq t}$ and $\boldsymbol{f}_t = (f(\boldsymbol{x}_i))_{1\leq i\leq t}$, the information gain for the selected points up to iteration t can be expressed in terms of the predictive variances as follows*

$$I(\boldsymbol{y}_t; \boldsymbol{f}_t) = \frac{1}{2}\sum_{i=1}^{t}\log(1 + \sigma^{-2}\sigma_{i-1}^2(\boldsymbol{x}_i)).$$

Proof. See Lemma 5.3 in [Srinivas *et al.*, 2010]. □

Lemma 5. *While running LSE with β_t as in Lemma 1, it holds that*

$$a_t(\boldsymbol{x}_t) \leq \sqrt{\frac{C_1\beta_t\gamma_t}{4t}},\ \forall t \geq 1,$$

where $C_1 = 8/\log(1+\sigma^{-2})$.

Proof. Similarly to Lemma 5.4 in [Srinivas *et al.*, 2010], from Lemma 2 it follows that for any $i \geq 1$

$$\begin{aligned} a_i^2(\boldsymbol{x}_i) &\leq \beta_i\sigma_{i-1}^2(\boldsymbol{x}_i) \\ &\leq \beta_i\sigma^2(\sigma^{-2}\sigma_{i-1}^2(\boldsymbol{x}_i)) \\ &\leq \beta_i\sigma^2 C_2\log(1 + \sigma^{-2}\sigma_{i-1}^2(\boldsymbol{x}_i)), \end{aligned}$$

where $C_2 = \sigma^{-2}/\log(1+\sigma^{-2})$. Using Lemma 4 in the above expression, the fact that β_i is nondecreasing in i, and defining $C_1 = 8\sigma^2 C_2$, we get for any $t \geq 1$

$$\begin{aligned} C_1\beta_t\gamma_t &\geq C_1\beta_t I(\boldsymbol{y}_t; \boldsymbol{f}_t) \\ &\geq 4\sum_{i=1}^{t} a_i^2(\boldsymbol{x}_i) \\ &\geq \frac{4}{t}\left(\sum_{i=1}^{t} a_i(\boldsymbol{x}_i)\right)^2 && \text{(by Cauchy-Schwarz)} \\ &= 4t\left(\frac{1}{t}\sum_{i=1}^{t} a_i(\boldsymbol{x}_i)\right)^2 \\ &\geq 4t a_t^2(\boldsymbol{x}_t) && \text{(by Lemma 3)} \end{aligned}$$

□

Lemma 6. *While running LSE, if for some $t \geq 1$, $a_t(\boldsymbol{x}_t) \leq \epsilon$, then $U_{t+1} = \varnothing$.*

Proof. Assume that $U_{t+1} \neq \varnothing$, i.e. there exists a point $\boldsymbol{x} \in U_t$, which does not meet the classification conditions (lines 12 and 15) of Algorithm 1. Consequently, that point satisfies $\max(C_{t+1}(\boldsymbol{x})) > h + \epsilon$ and $\min(C_{t+1}(\boldsymbol{x})) < h - \epsilon$. It follows that

$$\begin{aligned} \epsilon &< \min\{\max(C_{t+1}(\boldsymbol{x})) - h, h - \min(C_{t+1}(\boldsymbol{x}))\} \\ &= a_{t+1}(\boldsymbol{x}) \\ &\leq a_t(\boldsymbol{x}) \\ &\leq a_t(\boldsymbol{x}_t), && \text{(by LSE's selection rule)} \end{aligned}$$

which contradicts the lemma's assumption. □

Corollary 2. *The LSE algorithm terminates after at most T iterations, where T is the smallest positive integer satisfying*

$$\frac{T}{\beta_T\gamma_T} \geq \frac{C_1}{4\epsilon^2}.$$

Lemma 7. *For any $h \in \mathbb{R}$ and $\delta \in (0,1)$, and $\epsilon > 0$, after running LSE with β_t as in Lemma 1, with probability at least $1-\delta$ the returned solution is ϵ-accurate, that is*

$$\Pr\left\{\max_{\boldsymbol{x}\in D}\ell_h(\boldsymbol{x}) \leq \epsilon\right\} \geq 1 - \delta.$$

Proof. The lemma follows directly from Corollary 1 and the classification conditions (lines 12 and 15) of Algorithm 1. □

Theorem 1 follows by combining Corollary 2 and Lemma 7.

B Proof of Theorem 2

Definition 1. *We label the inequalities that take part in* $\mathsf{LSE}_{\mathsf{imp}}$*'s classification rules as follows*

$$\min(C_t(\boldsymbol{x})) + \epsilon \geq h_t^{opt} \tag{Q1}$$
$$\max(C_t(\boldsymbol{x})) - \epsilon \leq h_t^{pes} \tag{Q2}$$
$$\max(C_t(\boldsymbol{x})) < f_t^{pes}. \tag{Q3}$$

Furthermore, we redefine here for convenience the following quantities from the main text

$$\begin{aligned} Z_t &= U_t \cup H_t^M \cup H_t^L \\ h &= \omega \max_{\boldsymbol{x}\in D} f(\boldsymbol{x}) \\ f_t^{opt} &= \max_{Z_{t-1}} \max(C_t(\boldsymbol{x})) \\ h_t^{opt} &= \omega f_t^{opt} \\ f_t^{pes} &= \max_{Z_{t-1}} \min(C_t(\boldsymbol{x})) \\ h_t^{pes} &= \omega f_t^{pes}. \end{aligned}$$

Lemma 8. *The following holds for any* $t \geq 1$

$$w_t(\boldsymbol{x}_t) \leq \beta_t^{1/2} \sigma_{t-1}(\boldsymbol{x}_t).$$

Proof. By the definition of the confidence region width

$$\begin{aligned} w_t(\boldsymbol{x}_t) &= \max(C_t(\boldsymbol{x}_t)) - \min(C_t(\boldsymbol{x}_t)) \\ &\leq \max(Q_t(\boldsymbol{x}_t)) - \min(Q_t(\boldsymbol{x}_t)) \\ &= 2\beta_t^{1/2} \sigma_{t-1}(\boldsymbol{x}_t). \end{aligned}$$

□

Lemma 9. *While running* $\mathsf{LSE}_{\mathsf{imp}}$*,* $w_t(\boldsymbol{x}_t)$ *is nonincreasing in* t*.*

Proof. Completely analogous to the proof of Lemma 3. □

Lemma 10. *While running* $\mathsf{LSE}_{\mathsf{imp}}$ *with* β_t *as in Lemma 1, it holds that*

$$w_t(\boldsymbol{x}_t) \leq \sqrt{\frac{C_1 \beta_t \gamma_t}{t}}, \; \forall t \geq 1,$$

where $C_1 = 8/\log(1+\sigma^{-2})$*.*

Proof. Completely analogous to the proof of Lemma 5, with the only difference being a factor of 2 in the bound of Lemma 8 compared to Lemma 2. □

Lemma 11. *While running* $\mathsf{LSE}_{\mathsf{imp}}$

$$h_t^{opt} - h_t^{pes} \leq \omega w_t(\boldsymbol{x}_t), \; \forall t \geq 1.$$

Proof. If we define

$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}\in Z_{t-1}} \max(C_t(\boldsymbol{x})), \tag{3}$$

then by (Q3) it follows that $\hat{\boldsymbol{x}} \in Z_t$. Consequently, we get

$$\begin{aligned} & f_t^{opt} - f_t^{pes} \\ &= \max_{\boldsymbol{x}\in Z_{t-1}} \max(C_t(\boldsymbol{x})) - \max_{\boldsymbol{x}\in Z_{t-1}} \min(C_t(\boldsymbol{x})) \\ &= \max(C_t(\hat{\boldsymbol{x}})) - \max_{\boldsymbol{x}\in Z_{t-1}} \min(C_t(\boldsymbol{x})) && \text{(by (3))} \\ &\leq \max(C_t(\hat{\boldsymbol{x}})) - \min(C_t(\hat{\boldsymbol{x}})) \\ &= w_t(\hat{\boldsymbol{x}}) \\ &\leq w_t(\boldsymbol{x}_t), && \text{(by } \hat{\boldsymbol{x}} \in Z_t\text{)} \end{aligned}$$

and, therefore, $h_t^{opt} - h_t^{pes} = \omega \left(f_t^{opt} - f_t^{pes}\right) \leq \omega w_t(\boldsymbol{x}_t)$. □

Lemma 12. *While running* $\mathsf{LSE}_{\mathsf{imp}}$*, if for some* $t \geq 1$*,* $w_t(\boldsymbol{x}_t) \leq 2\epsilon/(1+\omega)$*, then* $U_{t+1} = \varnothing$*.*

Proof. Assume that $U_{t+1} \neq \varnothing$, i.e. there exists a point $\boldsymbol{x} \in U_t \subseteq Z_t$, which does not meet (Q1) or (Q2). Consequently, that point satisfies $\min(C_{t+1}(\boldsymbol{x})) < h_{t+1}^{opt} - \epsilon$ and $\max(C_{t+1}(\boldsymbol{x})) > h_{t+1}^{pes} + \epsilon$. It follows that

$$\begin{aligned} 2\epsilon &< h_{t+1}^{opt} - h_{t+1}^{pes} + \max(C_{t+1}(\boldsymbol{x})) - \min(C_{t+1}(\boldsymbol{x})) \\ &\leq h_{t+1}^{opt} - h_{t+1}^{pes} + w_{t+1}(\boldsymbol{x}) \\ &\leq h_{t+1}^{opt} - h_{t+1}^{pes} + w_t(\boldsymbol{x}) \\ &\leq h_{t+1}^{opt} - h_{t+1}^{pes} + w_t(\boldsymbol{x}_t) && \text{(by } \mathsf{LSE}_{\mathsf{imp}}\text{'s selection rule)} \\ &\leq \omega w_{t+1}(\boldsymbol{x}_{t+1}) + w_t(\boldsymbol{x}_t) && \text{(by Lemma 11)} \\ &\leq \omega w_t(\boldsymbol{x}_t) + w_t(\boldsymbol{x}_t) && \text{(by Lemma 9)} \\ &\leq (1+\omega) w_t(\boldsymbol{x}_t), \end{aligned}$$

which contradicts the lemma's assumption. □

Corollary 3. *The* $\mathsf{LSE}_{\mathsf{imp}}$ *algorithm terminates after at most* T *iterations, where* T *is the smallest positive integer satisfying*

$$\frac{T}{\beta_T \gamma_T} \geq \frac{C_1(1+\omega)^2}{4\epsilon}.$$

Lemma 13. *While running* $\mathsf{LSE}_{\mathsf{imp}}$*,* f_t^{pes} *is nondecreasing in* t*.*

Proof. Assume that at some iteration t

$$\boldsymbol{x} = \operatorname*{argmax}_{\boldsymbol{x}\in Z_{t-1}} \min(C_t(\boldsymbol{x})). \tag{4}$$

Since $\min(C_t(\boldsymbol{x}))$ is nondecreasing in t, to have $f_{t+1}^{pes} < f_t^{pes}$ would mean that $\boldsymbol{x} \notin Z_t$. That, in turn, implies that $\boldsymbol{x}$ was moved to H_t or L_t, therefore (Q3) was satisfied

$$\begin{aligned} \max_{\boldsymbol{x}\in Z_{t-1}} \min(C_t(\boldsymbol{x})) &> \max(C_t(\boldsymbol{x})) \\ &\geq \min(C_t(\boldsymbol{x})) \\ &= \max_{\boldsymbol{x}\in Z_{t-1}} \min(C_t(\boldsymbol{x})), && \text{(by (4))} \end{aligned}$$

which is a contradiction and proves our lemma. □

Lemma 14. *While running* $\mathsf{LSE}_{\mathsf{imp}}$

$$f_t^{opt} = \max_{\boldsymbol{x}\in D} \max(C_t(\boldsymbol{x})), \forall t \geq 1.$$

Proof. The "$\leq$" follows from $Z_{t-1} \subseteq D$. Now, assume that "$<$" holds, i.e. there exists an $\boldsymbol{x} \in D \setminus Z_{t-1}$, such that

$$\max_{\boldsymbol{x} \in Z_{t-1}} \max(C_t(\boldsymbol{x})) < \max C_t(\boldsymbol{x}). \tag{5}$$

The fact that $\boldsymbol{x} \in D \setminus Z_{t-1}$ implies that $\boldsymbol{x}$ was moved during some iteration $i \leq t$ to H_i or L_i, therefore $\boldsymbol{x}$ satisfied (Q3) at that iteration. Putting everything together, we get

$$\begin{aligned}
\max_{\boldsymbol{x} \in Z_{t-1}} \max(C_t(\boldsymbol{x})) &< \max C_t(\boldsymbol{x}) && \text{(by (5))} \\
&\leq \max C_i(\boldsymbol{x}) \\
&\leq \max_{\boldsymbol{x} \in Z_{i-1}} \min(C_i(\boldsymbol{x})) && \text{(by (Q3))} \\
&\leq \max_{\boldsymbol{x} \in Z_{t-1}} \min(C_t(\boldsymbol{x})) \\
& && \text{(by Lemma 13)} \\
&\leq \max_{\boldsymbol{x} \in Z_{t-1}} \max(C_t(\boldsymbol{x})),
\end{aligned}$$

which is a contradiction and proves the lemma. □

Lemma 15. *While running LSE$_{imp}$, the following holds with probability at least* $1 - \delta$

$$h_t^{opt} \geq h, \ \forall t \geq 1.$$

Proof. The following (in)equalities hold with probability at least $1 - \delta$

$$\begin{aligned}
h_t^{opt} &= \omega \max_{\boldsymbol{x} \in Z_{t-1}} \max(C_t(\boldsymbol{x})) \\
&= \omega \max_{\boldsymbol{x} \in D} \max(C_t(\boldsymbol{x})) && \text{(by Lemma 14)} \\
&\geq \omega \max_{\boldsymbol{x} \in D} f(\boldsymbol{x}) && \text{(by Corollary 1)} \\
&= h.
\end{aligned}$$

□

Lemma 16. *While running LSE$_{imp}$, the following holds with probability at least* $1 - \delta$

$$h_t^{pes} \leq h, \ \forall t \geq 1.$$

Proof. The following (in)equalities hold with probability at least $1 - \delta$

$$\begin{aligned}
h_t^{pes} &= \omega \max_{\boldsymbol{x} \in Z_{t-1}} \min(C_t(\boldsymbol{x})) \\
&\leq \omega \max_{\boldsymbol{x} \in Z_{t-1}} f(\boldsymbol{x}) && \text{(by Corollary 1)} \\
&\leq \omega \max_{\boldsymbol{x} \in D} f(\boldsymbol{x}) && \text{(by } Z_{t-1} \subseteq D\text{)} \\
&= h.
\end{aligned}$$

□

Lemma 17. *For any $\omega \in (0,1)$, $\delta \in (0,1)$, and $\epsilon > 0$, after running LSE$_{imp}$ with β_t as in Lemma 1, with probability at least $1 - \delta$ the returned solution is ϵ-accurate, that is*

$$\Pr\left\{\max_{\boldsymbol{x} \in D} \ell_h(\boldsymbol{x}) \leq \epsilon\right\} \geq 1 - \delta.$$

Proof. From Lemma 15 and Lemma 16 it follows that, with probability at least $1-\delta$, (Q1) and (Q2) are stricter conditions than the following

$$\begin{aligned}
\min(C_t(\boldsymbol{x})) + \epsilon &\geq h \\
\max(C_t(\boldsymbol{x})) - \epsilon &\leq h,
\end{aligned}$$

which are identical to the ones used by LSE. Therefore, the solution of LSE$_{\text{imp}}$ achieves at least as high accuracy as the one provided for LSE by Lemma 7. □

Theorem 2 follows by combining Corollary 3 and Lemma 17.

C LSE$_{\text{batch}}$ in More Detail

The pseudocode of Algorithm 3 highlights the way in which evaluation feedback is obtained in LSE$_{\text{batch}}$. Variable t_{fb} holds the latest step for which there is available feedback at each iteration and the inferred mean is updated whenever new feedback is available, as dictated by $\text{fb}[t+1]$. However, note that the inferred variance is updated at each iteration, irrespectively of available feedback. The batch extension of LSE$_{\text{imp}}$ works in a completely analogous way.

Theoretical analysis. To formally analyze the batch setting, we use the notion of a feedback function, introduced by Desautels *et al.* [2012]. In particular, we assume that there is a function $\text{fb} : \mathbb{N} \to \mathbb{N} \cup \{0\}$, such that $\text{fb}[t] \leq t - 1$ for all $t \geq 1$, and when selecting the next point at time step t, we have access to evaluated measurements up to time step $\text{fb}[t]$. For selecting batches of size B we can define $\text{fb}[1] = \ldots = \text{fb}[B] = 0$, $\text{fb}[B+1] = \ldots = \text{fb}[2B] = B$, and so on, but the feedback function also allows for defining more complex cases of delayed feedback.

To appropriately adjust the confidence interval scaling parameter, in their analysis for extending the GP-UCB algorithm to the batch setting, Desautels *et al.* [2012] utilized the *conditional information gain*

$$I(\boldsymbol{y}_A; f \mid \boldsymbol{y}_{1:\text{fb}[t]}) = H(\boldsymbol{y}_A \mid \boldsymbol{y}_{1:\text{fb}[t]}) - H(\boldsymbol{y}_A \mid f),$$

which quantifies the reduction in uncertainty about f by obtaining a number of observations $\boldsymbol{y}_A$, given that we already

Algorithm 3 The LSE$_{\text{batch}}$ extension

```
Input: sample set $D$, GP prior ($\mu_0 = 0$, $k$, $\sigma_0$),
       threshold value $h$, accuracy parameter $\epsilon$
Output: predicted sets $\hat{H}$, $\hat{L}$
// initialization as in LSE
$t_{\text{fb}} \leftarrow 0$
while $U_{t-1} \neq \varnothing$ do
    // classification and next point selection as in LSE
    if $\text{fb}[t+1] > t_{\text{fb}}$ then
        for $i = t_{\text{fb}} + 1, \ldots, \text{fb}[t+1]$ do
            $y_i \leftarrow f(\boldsymbol{x}_i) + n_i$
        Compute $\mu_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in U_t$
        $t_{\text{fb}} \leftarrow \text{fb}[t+1]$
    Compute $\sigma_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in U_t$
    $t \leftarrow t + 1$
$\hat{H} \leftarrow H_{t-1}$, $\hat{L} \leftarrow L_{t-1}$
```

have observations $\boldsymbol{y}_{1:\mathrm{fb}[t]}$ available. Following a similar treatment, we extend the convergence bound of Theorem 1 to the batch selection setting of $\mathsf{LSE}_{\mathsf{batch}}$ via bounding the maximum conditional information gain, resulting in the following theorem.

Theorem 3. *Assume that the feedback delay $t - \mathrm{fb}[t]$ is at most B for all $t \geq 1$, where B is a known constant. Also, assume that for all $t \geq 1$ the maximum conditional mutual information acquired by any set of measurements since the last feedback is bounded by a constant $C \geq 0$, i.e.*

$$\max_{A \subseteq D, |A| \leq B-1} I(f; \boldsymbol{y}_A \mid \boldsymbol{y}_{1:\mathrm{fb}[t]}) \leq C$$

Then, for any $h \in \mathbb{R}$, $\delta \in (0,1)$, and $\epsilon \geq 0$, if $\eta_t = e^C \beta_{fb[t]+1}$, $\mathsf{LSE}_{\mathsf{batch}}$ terminates after at most T iterations, where T is the smallest positive integer satisfying

$$\frac{T}{\eta_T \gamma_T} \geq \frac{C_1}{4\epsilon^2},$$

where $C_1 = 8/\log(1 + \sigma^{-2})$.

Furthermore, with probability at least $1 - \delta$, the algorithm returns an ϵ-accurate solution, that is

$$\Pr\left\{\max_{\boldsymbol{x} \in D} \ell_h(\boldsymbol{x}) \leq \epsilon\right\} \geq 1 - \delta.$$

Note that, as intuitively described in the main text, the scaling parameter η_t has to increase by a factor of e^C to compensate for the outdated mean estimates used in the confidence regions $C_t^b(\boldsymbol{x})$. Normally, C depends on the batch size B. However, Desautels *et al.* [2012] have shown that, initializing their $\mathsf{GP\text{-}BUCB}$ algorithm with a number of sequentially selected maximum variance samples, results in a constant factor increase of η_t compared to β_t, independently of B. In Section 5, we also used maximum variance initialization in our experiments, which allowed us to select a constant value of η_t (larger than β_t) that worked well across different batch sizes.

D Proof of Theorem 3

Lemma 18. *For any $\boldsymbol{x} \in D$ and $t \geq 1$ the ratio of $\sigma_{fb[t]}(\boldsymbol{x})$ to $\sigma_{t-1}(\boldsymbol{x})$ is bounded as follows*

$$\frac{\sigma_{fb[t]}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})} \leq \exp\left\{I(f; \boldsymbol{y}_{fb[t]+1:t-1} \mid \boldsymbol{y}_{1:fb[t]}\right\}.$$

Proof. See Lemma 1 in [Desautels *et al.*, 2012]. □

Lemma 19. *Assume that for all $t \geq 1$ the maximum conditional mutual information acquired by any set of measurements since the last feedback is bounded by a constant $C \geq 0$, i.e.*

$$\max_{A \subseteq D, |A| \leq B-1} I(f; \boldsymbol{y}_A \mid \boldsymbol{y}_{1:fb[t]}) \leq C. \tag{6}$$

Then, if $\eta_t = e^{2C} \beta_{fb[t]+1}$, the following holds with probability at least $1 - \delta$

$$f(\boldsymbol{x}) \in Q_t^b(\boldsymbol{x}),\ \forall \boldsymbol{x} \in D\ \forall t \geq 1.$$

Proof. From Lemma 18 and (6), it follows that for any $\boldsymbol{x} \in D$ and $t \geq 1$

$$\frac{\sigma_{fb[t]}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})} \leq \exp\left\{I(f; \boldsymbol{y}_{fb[t]+1:t-1} \mid \boldsymbol{y}_{1:fb[t]}\right\} \leq e^C$$
$$\Rightarrow e^C \sigma_{t-1}(\boldsymbol{x}) \geq \sigma_{fb[t]}(\boldsymbol{x}).$$

Using this, the range of $Q_t^b(\boldsymbol{x})$ can be related to the range of $Q_{fb[t]+1}(\boldsymbol{x})$ as follows

$$2\eta_t^{1/2} \sigma_{t-1}(\boldsymbol{x}) = 2e^C \beta_{fb[t]+1}^{1/2} \sigma_{t-1}(\boldsymbol{x}) \geq 2\beta_{fb[t]+1}^{1/2} \sigma_{fb[t]}(\boldsymbol{x}).$$

Furthermore, $Q_t^b(\boldsymbol{x})$ and $Q_{fb[t]+1}(\boldsymbol{x})$ have the same midpoint, namely $\mu_{fb[t]}(\boldsymbol{x})$, therefore the above range inequality implies that

$$Q_t^b(\boldsymbol{x}) \supseteq Q_{fb[t]+1}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in D\ \forall t \geq 1. \tag{7}$$

Finally, from Lemma 1 we have that

$$\Pr\{f(\boldsymbol{x}) \in Q_{fb[t]+1}(\boldsymbol{x})\} \geq 1 - \delta,\ \forall \boldsymbol{x} \in D\ \forall t \geq 1$$
$$\overset{(7)}{\Rightarrow} \Pr\{f(\boldsymbol{x}) \in Q_t^b(\boldsymbol{x})\} \geq 1 - \delta,\ \forall \boldsymbol{x} \in D\ \forall t \geq 1.$$

□

Corollary 4. *Given the assumptions of Lemma 19, the following holds with probability at least $1 - \delta$*

$$f(\boldsymbol{x}) \in C_t^b(\boldsymbol{x}),\ \forall \boldsymbol{x} \in D\ \forall t \geq 1.$$

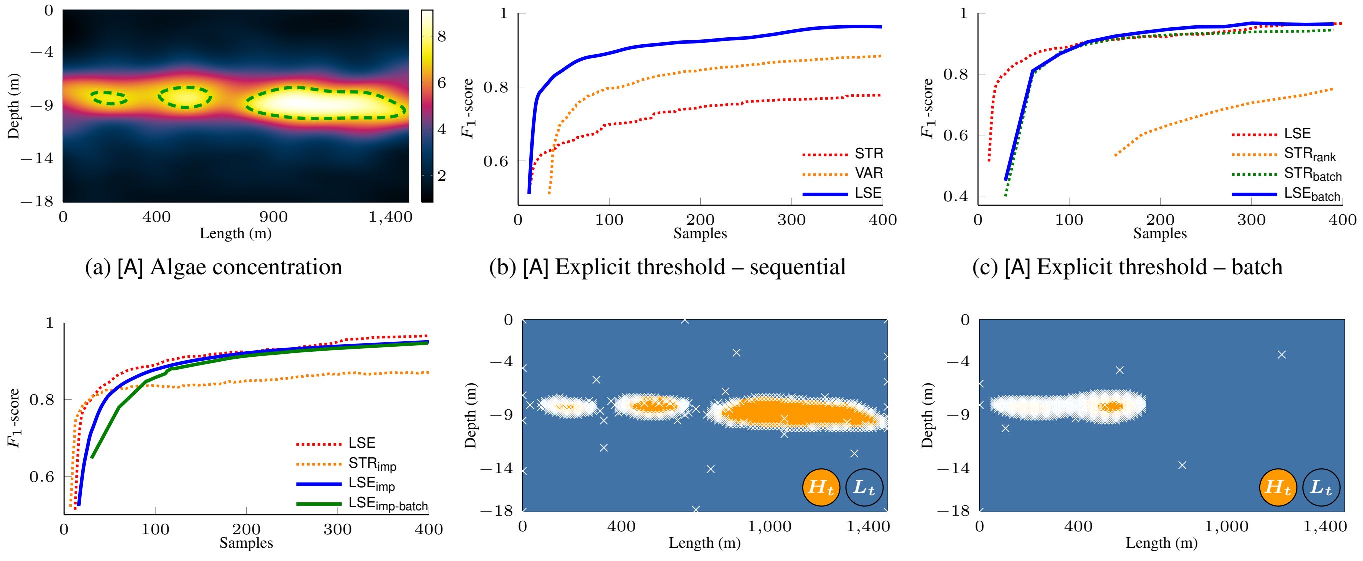
Note that Corollary 4 is completely analogous to Corollary 1. Thus, the results of Lemmas Lemma 2–7 also hold for the case of $\mathsf{LSE}_{\mathsf{batch}}$, provided that η_t, as defined in Lemma 19, is used in place of β_t, which proves Theorem 3.

E Further Experiments

We present here one more set of experiments on a dataset of *Planktothrix rubescens*[2] concentration in the same environmental monitoring setting that was presented in Section 5. The experimental setup is identical to the one we used for the dataset of chlorophyll measurements. In particular, we found suitable hyperparameters by maximum likelihood on a different dataset and a 100×100 grid of points was sampled from the GP posterior (see Figure 3a) to evaluate the algorithms. Furthermore, the threshold level was selected as $h = 7$ RFU with a corresponding value of $\omega = 0.75$.

In Figure 3b we see that STR performs significantly worse than LSE in this experiment. This is a consequence of its eagerness to sample near the estimated level set and its insufficient exploration to detect the superlevel areas on the left of the lake transect. Note that only after more than 500 samples does STR manage to explore these areas and achieve high classification accuracy, while LSE has already achieved near-perfect accuracy at about 300 samples. The difference between the samples selected by STR and LSE is depicted in Figures 3e and 3f. After 300 samples, LSE has collected a

[2] Planktothrix rubescens is a genus of blue-green algae that can produce toxins.



(a) [A] Algae concentration

(b) [A] Explicit threshold – sequential

(c) [A] Explicit threshold – batch

(d) [A] Implicit threshold

(e) [A] LSE sampling

(f) [A] STR sampling

Figure 3: Results for the algae concentration [A] dataset. (a) Algae concentration in RFU inferred from 2024 lake measurements. (b), (e), (f) LSE greatly outperforms VAR and STR, as STR gets stuck due to limited exploration. (c) LSE$_{\text{batch}}$ and STR$_{\text{batch}}$ are only slightly worse than LSE, while the naive STR$_{\text{rank}}$ is far inferior. (d) LSE$_{\text{imp}}$ and LSE$_{\text{imp-batch}}$ achieve high accuracy at a slower rate than LSE, while STR$_{\text{imp}}$ fails to do so.

large number of samples near all three components of the superlevel set shown in Figure 3a, while STR has exclusively focused on the right component.

In the batch experiments of Figure 3c, LSE$_{\text{batch}}$ is only slightly inferior to LSE, while STR$_{\text{rank}}$ performs poorly. It is notable that STR$_{\text{batch}}$ does not suffer from the limited exploration of STR that was noted above and achieves similar performance to LSE$_{\text{batch}}$. Finally, in the implicit setting of Figure 3d, LSE$_{\text{imp}}$ and LSE$_{\text{imp-batch}}$ have a somewhat higher sampling cost compared to LSE, while STR$_{\text{imp}}$, even after 500 measurements, fails to achieve an F_1-score of 0.9.