Foundations of Data Science

doi:10.3934/fods.2024050

A FINITE-HORIZON APPROACH TO ACTIVE LEVEL SET ESTIMATION

Phillip Kearns[✉1], Bruno Jedynak[✉2] and John Lipor[✉*3]

[1]Department of Chemistry and Biochemistry, University of Oregon, Eugene, OR 97403 USA

[2]Department of Mathematics and Statistics,
Portland State University, Portland, OR 97201 USA

[3]Department of Electrical and Computer Engineering,
Portland State University, Portland, OR 97201 USA

(Communicated by Marco Iglesias)

Abstract. We consider the problem of active learning for level set estimation (LSE), where the goal is to localize all regions where a function of interest lies above/below a given threshold as quickly as possible. We present a finite-horizon search procedure to perform LSE in one dimension while optimally balancing both the final estimation error and the distance traveled during active learning for a fixed number of samples. A tuning parameter is used to trade off between the estimation accuracy and distance traveled. We show that the resulting optimization problem can be solved in closed form and that the resulting policy generalizes existing approaches to this problem. We then show how this approach can be used to perform level set estimation in two dimensions, under some additional assumptions, under the popular Gaussian process model. Empirical results on synthetic data indicate that as the cost of travel increases, our method's ability to treat distance nonmyopically allows it to significantly improve on the state of the art. On real air quality data, our approach achieves roughly one fifth the estimation error at less than half the cost of competing algorithms.

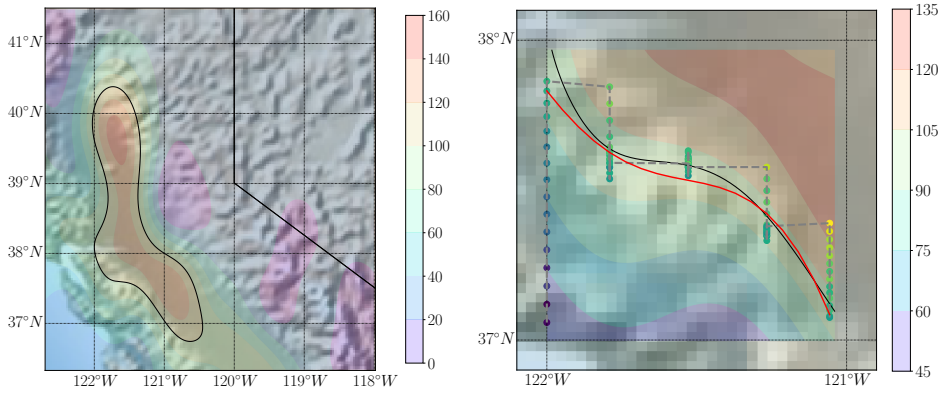
1. **Introduction.** In recent years, there has been a growing interest in autonomously sampling real-world environmental phenomena, owing in part to the increasing occurrence of extreme events such as wildfires in the United States. In particular, the problem of adaptively sampling the environment to determine all regions where a phenomenon of interest is above or below a critical threshold—a problem known as *level set estimation* (LSE)—has received a great deal of attention within the signal processing and machine learning communities [21, 7, 32]. The resulting algorithms are often designed with the goal of deployment on an autonomous, mobile sampling vessel, such as an unmanned aerial vehicle (UAV) [37]. Since these vehicles are tasked with covering regions on the order of hundreds of square kilometers, a key

2020 *Mathematics Subject Classification.* Primary: 68T05, 62P12; Secondary: 90C39.

Key words and phrases. Adaptive sampling, autonomous systems, dynamic programming, Gaussian processes, level set estimation, mobile sensors.

The second author is supported by NSF grant 2136228 and NIH awards RO1AG021155, R01EY032284, and R01AG027161. The third author is supported by NSF grant CIF-2046175 and DARPA award D22AP00135.

*Corresponding author: John Lipor.



(A) (B)

FIGURE 1. Map of PM 2.5 following the California Camp Fire on November 18, 2018. (a) Full level set boundary (black) denoting all locations where PM 2.5 is above 100 μg / m^3. (b) Subset of region and samples collected via proposed method, which performs a series of one-dimensional searches. Dots denote sample locations with measurement values indicated by their color, red solid line is estimated boundary, and gray dashed line is path traversed by sensor ground truth. Corresponding sub-region is approximately 111 km $\times$ 111 km.

component of adaptive sampling methods is the ability to account for the costs associated with both the number of measurements taken and the distance traveled throughout the sampling procedure.

As a motivating problem, we consider the task of tracking wildfires, where our goal is to determine the spatial extent of particulate matter 2.5 (PM 2.5) caused by wildfire smoke (see Fig. 1). Algorithms designed to rapidly determine the boundary of such a region fall within the category of *active learning* or *adaptive sampling* [45, 11] and typically try to maximize a notion of information gain per sample. However, this approach fails to account for the distance traveled throughout the sampling procedure. Hence, standard approaches to active learning based in search space reduction [38, 14, 55] or adaptive submodularity [20], which seek to minimize only the number of samples taken, will be accompanied by potentially dramatic drawbacks in terms of total sampling cost. While the approaches in [23, 7] account for arbitrary costs, these treat costs myopically, failing to account for the expected future cost after sampling a given location. Newer, bisection-style search methods such as quantile search (QS) [34] and its extension [33] both achieve an explicit tradeoff between the number of samples and distance traveled. Although these improve upon previous methods in terms of total sampling time, neither guarantees to find the optimal search procedure.

For any fixed $N \geq 1$, our procedure collects N measurements such that the weighted combination of distance traveled and final estimation error is optimally balanced. At its extremes, this algorithm minimizes either the final entropy or total distance traveled, with a tradeoff achieved by varying a user-specified tuning parameter. We show that for a one-dimensional step function, fixing N allows

the resulting cost to be optimized in closed form, eschewing the need for dynamic programming. We prove that the resulting policy is indeed a global minimum, as opposed to a critical point only. We then present a method for handling noisy measurements and prove this approach converges almost surely to the true change point of a one-dimensional step function. We extend this idea to show how the proposed search algorithm can be used to perform LSE in Gaussian processes (GPs), a topic that has received a great deal of attention in the machine learning literature [21, 24, 7, 48]. We provide extensive simulations on both synthetic data, as well as air quality data obtained from the AirNow database [52]. Finally, we compare our approach with the state-of-the-art in Gaussian process level set estimation (GP-LSE) [7] and demonstrate that the proposed method is capable of estimating the level set at a lower sampling cost while requiring a fraction of the computation time.

2. Problem formulation & related work.

As stated in the introduction, we are ultimately concerned with the problem of level set estimation in spatial domains, i.e., of estimating the superlevel set

$$S = \left\{x \in \mathbb{R}^d : f(x) \geq \gamma\right\}, \tag{1}$$

where $f : [0,1]^d \to \mathbb{R}$ is a function governing some phenomenon of interest and γ is a user-defined threshold. In this work, we are primarily concerned with the case where f is a one-dimensional step function belonging to the class

$$\mathcal{F} = \{f_\theta : f_\theta(x) = \mathbb{1}\left\{[0,\theta)\right\}, \theta \in [0,1]\},$$

where $\mathbb{1}\{E\}$ denotes the indicator function, which takes the value one when $x \in E$ and zero otherwise. In this case, the superlevel set is $S = \{x \in [0,1] : x < \theta\}$, and LSE is equivalent to estimating the change point θ. Although this model may seem highly restrictive, we will show that two-dimensional level set boundaries can be estimated using a series of one-dimensional step functions, and that such an approach outperforms state-of-the-art algorithms that consider the two-dimensional problem directly. An example of using a series of one-dimensional searches to estimate a two-dimensional boundary is illustrated in Fig. 1(b).

To perform boundary estimation, our sampling proceeds as follows. Assume we obtain observations $\{Y_n\}_{n=1}^N \in \{0,1\}^N$ from the sample locations $\{X_n\}_{n=1}^N$ in the unit interval in a sequential fashion according to $Y_n = \mathbb{1}\{x \in S\}$, where S is the superlevel set defined in (1). In the case of one-dimensional step functions, each sample obtained reduces the interval in which the change point may lie. We treat the unknown change point θ as a random variable. Our goal is then to estimate the change point location while minimizing the sampling cost for a fixed number of samples, a function of both the final expected interval size *and* expected distance traveled.

2.1. Related work.

A variety of adaptive sampling methods have been proposed throughout the literature from various communities. Several popular approaches to LSE assume that the underlying function of interest is a GP [42], which yields a posterior distribution on the value at each point. In [21], the authors leverage the GP model to construct confidence intervals around the value of each point, sequentially sampling points of highest ambiguity. This approach was extended in [24] and [7], providing novel approaches for sampling while also accounting for the distance traveled between points. Another method for path-efficient LSE that seeks to reduce the distance traveled by the mobile sensor is proposed in [8], but this method

assumes the vehicle can continuously acquire measurements with a negligible cost. A closely related body of work exists within the uncertainty quantification literature, where LSE is sometimes referred to as *excursion set estimation* [1]. Within these approaches, many rely on the stepwise uncertainty reduction (SUR) strategy [18, 3], which maximizes the uncertainty reduction according to a given criterion. In [3], the authors aim to minimize a quadratic loss on the volume (measure) of the superlevel set. This approach is extended in [2], where a criterion is introduced that gives direct control over false positives by using a conservative estimate of the superlevel set. This conservative estimate is closely related to the confidence intervals utilized by [21, 24, 7]. In [17], a variation of the Bichon criterion is proposed for SUR with the goal of obtaining a greater degree of exploration. These approaches allow for selecting a single point or a batch of points to reduce uncertainty. However, they all begin with an initial set of points distributed throughout the region of interest (e.g., via Latin hypercube sampling), and none considers the additional cost of moving a sensor throughout the sampling region.

The authors of [46] introduce the idea of adaptive data collection for mobile path planning, or *informative path planning*, where previous samples are used to guide the motion of the sensing vehicles for further sampling. Algorithms for informative path planning typically focus on maximizing information gain over a scalar field for an underwater autonomous vehicle. The approaches presented in [46, 6, 57] accommodate a wide range of sampling scenarios that include varied sampling time, path constraints, and limited battery. However, these methods often require a coarse sampling of the entire region of interest, which is not feasible for the large spatial regions considered here. In contrast, boundary detection methods like those in [36, 10, 29] use mobile sensors to map a spatial threshold as closely as possible. These methods provide efficient and accurate mappings of a binary classification boundary but unfortunately do not account for the cost of obtaining each sample.

Among the approaches from active learning, many rely on the principle of search space reduction [14, 55, 38], which aims to rapidly reduce the set of points where the level set boundary may lie through intelligent sampling. In general, these methods do not permit the inclusion of distance-based or other penalties, and as a result they tend to yield bisection-type solutions [9] that require few samples but may travel large distances. These methods can also be viewed as "greedy" approaches that aim to maximize search space reduction at each step. While greedy methods have been shown to be near optimal in terms of sample complexity [38, 14], they frequently ignore additional costs that may be incurred during the sampling procedure. Moreover, methods such as [16] that greedily incorporate realistic costs into the algorithm formulation have been shown to perform worse than the alternative approaches when applied to distance-penalized searches [34].

A popular greedy approach to active learning relies on the concept of adaptive submodularity (AS) [19]. AS is a diminishing returns principle, which informally states that samples are more valuable early in the search procedure. The work of [20] shows that a greedy procedure is optimal up to a constant factor for several active learning problems, including the case of nonuniform label costs. However, AS is a property of set functions, and does not consider a sequential dependency among sampling locations. A notion of submodular optimization with sequential dependencies was presented in [49], but the proposed algorithm relies on a reordering procedure that is not applicable to our problem. While [23] provides a theoretical analysis of greedy active learning with non-uniform costs, the authors only consider

the case of query costs being fixed. In contrast, our scenario has non-uniform and dynamic costs, where travel time depends on the distance between points.

Of primary relevance to the work presented in this paper is the work of [34], which introduces the quantile search (QS) algorithm for determining the change point of a one-dimensional step function while balancing the above costs. QS is a generalization of binary bisection [12, 26, 9], where the idea is that by successively sampling a fixed fraction $1/m$, where $m > 2$, into the remaining feasible interval, the desired tradeoff between number of samples and distance traveled can be achieved. The authors characterize the expected error after a fixed number of samples as well as the distance traveled; they further provide an approach to handling noisy measurements and prove its convergence. This work was extended in [33], introducing the uniform-to-binary (UTB) algorithm, where the key observation is that QS can be improved by allowing the search parameter m to vary with time. While both QS and UTB provide promising empirical results, neither algorithm provides guarantees of optimality in terms of the total sampling cost. Most recently, an optimal approach based on dynamic programming was presented in [54], where the search procedure is cast as a stochastic shortest path problem [5, Ch. 2]. However, the use of dynamic programming requires an increased computation time, and the resulting solution depends heavily on the discretization used. In this work, we present an approach that strictly generalizes the QS algorithm while still admitting a closed-form solution that can be easily deployed on a mobile sensing device.

3. Finite horizon search.

In this section, we describe our approach to distance-penalized LSE, which we refer to as *finite horizon search* (FHS). To appropriately penalize distance, FHS considers a fixed number of measurements (i.e., a finite sampling horizon) and optimizes the weighted sum of distance traveled and entropy in the posterior distribution of the change point θ after obtaining these measurements. A tuning parameter allows the user to control the importance of distance penalization, resulting in binary search at one extreme and sampling adjacent locations at the other. In the case of noiseless measurements, we show that the optimal sampling policy can be obtained in closed form. We then show how this policy can be extended to handle noisy observations and prove the resulting method converges almost surely to the true change point. Finally, we describe an approach to the well-studied GP-LSE problem in the case where the level set boundary can be written as a function in one dimension.

3.1. Noiseless measurements.

We first consider the simple case of noiseless, binary-valued measurements, where our goal is to estimate the change point θ on the unit interval. It is convenient, while not restrictive, to define search strategies in terms of the *fraction of the remaining interval* to move at each step, whether forward or backward, in an analogous fashion to [34, 33]. The resulting class of policies is adaptive to the unknown location of θ and non-restrictive in the sense that any optimal policy will not sample in locations with probability zero (locations outside the remaining interval).

Begin with a uniform prior on the change point θ, and let the N fractions be $\{z_n\}_{n=1}^N$, where $z_n \in [0, 1]$ for $n = 1, \dots, N$. A straightforward Bayesian update yields the posterior distribution for θ after each sample. Let H_N be the entropy of the posterior distribution after N observations, D_N be the total distance traveled, and $\lambda \geq 0$ be a tuning parameter that governs the tradeoff between these costs.

We define the expected sampling cost after N observations as the weighed sum of exponentiated entropy and distance

$$J(z_1, \ldots, z_N) = \mathbb{E}_\theta \left[e^{H_N} + \lambda D_N \right]. \tag{2}$$

Note that for a uniform distribution on an interval of length a, $e^{H_N} = e^{\log(a)} = a$; thus, when beginning with a uniform prior on θ, (2) is equivalent to minimizing a weighted combination of the (expected) final interval length and distance traveled. In what follows, we will derive a closed-form solution to this problem, as well as a means of computing the number of samples required to obtain an expected interval length below a given threshold (see Alg. 1).

3.1.1. *Closed-form solution.* We now demonstrate that the global optimum of (2) can be found in closed form. We first define the *feasible interval* as the interval in which the change point may lie. Formally,

Definition 1. Assume n measurements at locations $X_1, \ldots, X_n$ have been obtained and define

$$X_l = \max \{ X_i \in X_1, \ldots, X_n : Y_i = 1,\ i \leq n \}$$

and

$$X_u = \min \{ X_i \in X_1, \ldots, X_n : Y_i = 0,\ i \leq n \}.$$

Then the *feasible interval* after n samples is $[X_l, X_u]$.

To derive the optimal sampling fractions, we begin by rewriting (2) in terms of the expected size of the feasible interval at each step, recognizing that the distance traveled at a given step is equal to the product of the interval size and the sampling fraction. The resulting cost function can be differentiated to find a critical point. The principle of dynamic programming verifies that the resulting solution is indeed a global optimum (see Thm. 3.2).

Theorem 3.1. *Let $\lambda \in [0, 2)$ and assume the unknown change point has distribution $\theta \sim$ Unif$([0, 1])$. Further, assume the N measurements are defined via N fractions $z_1, \ldots, z_N \in [0, 1]$ denoting the proportion of the current feasible interval to sample. Then the critical points of the cost function (2) are of the form*

$$z_k^* = \frac{1}{2} - \lambda \frac{1}{4\rho_k}, \quad k = 1, \ldots, N, \tag{3}$$

where $\rho_N = 1$, $\xi_i^* = (z_i^*)^2 + (1 - z_i^*)^2$, and

$$\rho_k = \prod_{i=k+1}^{N} \xi_i^* + \lambda \sum_{i=k+1}^{N} z_i^* \prod_{j=k+1}^{i-1} \xi_j^*, \quad k = 1, \ldots, N-1, \tag{4}$$

depends only on the fractions $z_{k+1}, \ldots, z_N$.

Proof. The proof begins by rewriting the cost function in terms of the expected length of the feasible interval. Let H_N be the entropy of the posterior distribution after N measurements, so that $H_0 = 0$. Let $\xi_i = z_i^2 + (1 - z_i)^2$ and define $\xi_0 = 1$. Then by Lemma A.1 (see Appendix A), we have that

$$\mathbb{E}\left[e^{H_N} \right] = \prod_{i=1}^{N} \left(z_i^2 + (1 - z_i)^2 \right) = \prod_{i=1}^{N} \xi_i.$$

Let D_N be the distance traveled after N samples. Note that this distance is exactly the product of the interval length and the fraction of the interval to be traveled at each step, i.e.,

$$D_N = \sum_{i=1}^{N} z_i e^{H_{i-1}}.$$

Therefore

$$\begin{aligned}\mathbb{E}\left[D_N\right] &= \mathbb{E}\left[\sum_{i=1}^{N} z_i e^{H_{i-1}}\right] \\ &= \sum_{i=1}^{N} z_i \mathbb{E}\left[e^{H_{i-1}}\right].\end{aligned}$$

Applying Lemma A.1 then yields

$$\mathbb{E}\left[D_N\right] = \sum_{i=1}^{N} z_i \prod_{j=0}^{i-1} \xi_j.$$

We can rewrite the cost function (2) as

$$\begin{aligned}J\left(z_1, \ldots, z_N\right) &= \mathbb{E}\left[e^{H_N}\right] + \lambda \mathbb{E}\left[D_N\right] \\ &= \prod_{i=1}^{N} \xi_i + \lambda \sum_{i=1}^{N} z_i \prod_{j=0}^{i-1} \xi_j. \end{aligned} \tag{5}$$

After rewriting in the form (5), we can easily compute the gradient to be

$$\frac{\partial J}{\partial z_l} = \left(\prod_{i=1}^{l-1} \xi_i\right)\left(\left(4z_l - 2\right)\rho_l + \lambda\right), \tag{6}$$

and setting the gradient to zero yields

$$z_l = \frac{1}{2} - \lambda \frac{1}{4\rho_l}.$$

□

Thm. 3.1 characterizes the critical points of (2). Although setting (6) to zero yields a unique solution, this is not sufficient to guarantee global optimality (a global optimum could lie on the boundary of $[0,1]^N$). Further, even in its simplified form (5), the cost function is a high-order polynomial whose convexity is difficult to analyze.

Theorem 3.2. *The critical point characterized by Thm. 3.1 is the global optimum of the cost function (2).*

Proof. The argument is based on the dynamic programming lemma [4], restated here for convenience.

Lemma 3.3. *Suppose a sequence $z_1^*, \ldots, z_N^* \in [0,1]^N$ is such that for any $z_1, \ldots, z_N$ (also lying in the unit interval)*

$$J(z_1, \ldots, z_N) \geq J(z_1, .., z_{N-1}, z_N^*) \tag{7}$$

and for $1 \leq p \leq N-1$ and any $z_1, \ldots, z_p$ that

$$J(z_1, \ldots, z_p, z_{p+1}^*, \ldots, z_N^*) \geq J(z_1, \ldots, z_p^*, z_{p+1}^*, \ldots, z_N^*). \tag{8}$$

Then for all sequences $z_1, \ldots, z_N$

$$J(z_1, \ldots, z_N) \geq J(z_1^*, \ldots, z_N^*). \tag{9}$$

We verify that the local minimum defined in Thm. 3.1 satisfies the hypothesis of Lemma 3.3. For any $z_1, \ldots, z_p$, define $\pi_p = \prod_{i=1}^{p} \xi_i$. To verify the statement (7), for a fixed $z_1, \ldots, z_{n-1}$, we let

$$\begin{aligned} f_n(z) &= J(z_1, \ldots, z_{n-1}, z) \\ &= \pi_{n-1}\xi + \lambda z \pi_{n-1} + \lambda \sum_{i=1}^{n-1} z_i \pi_{i-1}, \end{aligned}$$

where $\xi = z^2 + (1-z)^2$. The final term above does not depend on z, indicating that $f_n(z)$ is a second-order polynomial in z. Moreover, $\pi_{n-1} > 0$, so the above is strictly convex, and hence a unique global minimizer can be found by differentiation. It is easily verified that this corresponds to the critical point found in Thm. 3.1. We next verify statement (8). Let

$$\begin{aligned} f_p(z) &= J(z_1, \ldots, z_{p-1}, z, z_{p+1}^*, \ldots, z_n^*) \\ &= \pi_{p-1} \left(\xi \prod_{i=p+1}^{n} \xi_i^* + \right. \\ &\qquad \left. \lambda \left(z + z_{p+1}^* \xi + \cdots + z_n^* \xi \prod_{i=p+1}^{n-1} \xi_i^* \right) \right) + \lambda \sum_{i=1}^{p-1} z_i \pi_{i-1} \\ &= \pi_{p-1} \left(\lambda z + \xi \rho_p \right) + \lambda \sum_{i=1}^{p-1} z_i \pi_{i-1}, \end{aligned}$$

where ρ_p is defined in (4). The above is again a second-order polynomial in z and therefore convex. Minimization through differentiation again yields a global minimizer that corresponds with the critical point defined in Thm. 3.1. Therefore, both statements of Lemma 1 are satisfied for the sequence $z_1^*, \ldots, z_N^*$ defined in Thm. 3.1, indicating that the critical points are indeed global minimizers of the cost function (2). □

The above results show that the optimal N-step FHS policy can be obtained in closed form. Further, the sampling fractions can be computed in linear time, beginning with $z_N = 1/2 - \lambda/4$ and proceeding backwards. While the above considers the case of the unit interval, it is straightforward to show that the cost (2) is linear in the length of the interval, and hence the search fractions are independent of the initial length. As a first observation, we note that when $N = 1$, FHS is a greedy approach that minimizes the one-step lookahead for the value function without concern for future consequences. In this case, the policy samples a constant fraction into the feasible interval, independent of the size of this interval. This is exactly the QS approach described in [34], and thus QS may be considered an instance of our proposed method with $N = 1$.

Examining Thm. 3.1, we see that ρ_k is monotonically increasing in k, and hence the sampling fractions are monotonically increasing with k, as can be seen in Fig. 2. The resulting behavior is to perform small movements early on in the sampling procedure, when the feasible interval is large, avoiding large movements at the sacrifice of information gain. As the feasible interval is reduced, the steps get

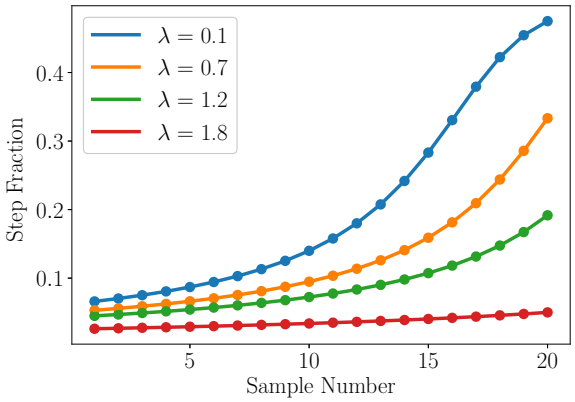


FIGURE 2. Sampling behavior of proposed FHS policy, showing fractions of the interval to travel at each step of a 20-step policy, where a larger λ penalizes distance traveled more heavily. The FHS policy increases the sampling fraction over time to avoid traveling large distances.

proportionally larger and emphasize information gain/entropy reduction, since the incurred distance penalty is smaller. This extends the intuition behind the UTB sampling procedure of [33] in a more principled manner.

The closed-form policy also provides insight into the range of admissible values of λ. Taking $\lambda = 0$ results in all fractions taking the value of $1/2$, which is consistent with the well-known fact that binary bisection performs entropy minimization [53]. A higher value for the distance penalty parameter λ results in a less aggressive policy, as the higher cost for potential overshoot encourages smaller steps. When $\lambda \geq 2$, the cost of traveling to obtain a measurement, λz_1, is larger than the expected reduction in entropy, $1 - \xi_1$, and the trivial sample which requires no displacement is preferred. This is seen most directly by the fact that the final step is $z_N = 0$ in this case, making $\rho_k = 1$ for all k, and therefore all sampling fractions identically zero.

Finally, we note that (2) may be solved directly using dynamic programming by discretizing the interval and allowing states to correspond to the possible lengths of the feasible interval. The corresponding cost is then the λ-penalized distance traveled at each step, with a terminal cost corresponding to the length of the final interval. In this light, the closed-form policy above may be viewed as an instance of dynamic programming, where each subproblem is solved in closed form.

3.1.2. *Searching with a fixed estimation error.* In certain instances, a user may wish to terminate the search procedure when the length of the feasible interval is below a given threshold to guarantee a fixed estimation error. By noting that the exponential entropy is equal to the feasible interval length, the result of Lemma A.1 can be used to determine the number of steps required to reduce the expected interval length below the threshold $\varepsilon > 0$. In particular, we leverage the fact that the tail subproblem of length $N-k$ is equivalent to the solution to the $(N-k)$-step problem for any $k \in \{1, \ldots, N-1\}$, and hence the expected interval size can be computed sequentially until it is below the threshold ε. Further, it is easy to show

Algorithm 1 Policy calculation for fixed estimation error

1: **Input:** stopping error $\varepsilon > 0$, distance penalty $\lambda \in [0, 2)$, initial interval length L
2: **Initialize:** $z_N \leftarrow \frac{1}{2} - \frac{\lambda}{4}$, $\xi_N = \frac{1}{2} + \frac{\lambda^2}{8}$, $k \leftarrow 0$
3: **while** $L \prod_{i=N-k}^{N} \xi_i > \varepsilon$ **do**
4: $\quad k \leftarrow k + 1$
5: $\quad$ compute ρ_{N-k} according to (4)
6: $\quad z_{N-k} \leftarrow \frac{1}{2} - \lambda/(4\rho_{N-k})$
7: $\quad \xi_{N-k} \leftarrow z_{N-k}^2 + (1 - z_{N-k})^2$
8: **end while**
9: $N \leftarrow k$

that intervals of arbitrary length L are reduced by the same fraction, and hence we are not restricted to intervals of unit length. Pseudocode for determining the expected number of samples and search fractions for each sample is given in Alg. 1.

In the case where a search terminates only after a given estimation error has been obtained, we follow a two-phase procedure. Before the search begins, we use the method presented in Alg. 1 to calculate the N steps such that the expected final interval size is less than ε. Then, in the first search stage, samples are taken according to this N-step policy. If the feasible interval is smaller than the desired threshold before all N samples have been taken, the search terminates. Otherwise, the algorithm performs a greedy search (optimal 1-step policy, line 7) until the interval is sufficiently small. This corresponds to the maximum entropy reduction among all N step sizes computed by the policy. Since entropy reduction has been shown to be optimal in terms of sample complexity [28], our choice may be viewed as minimizing the number of further samples required subject to a small amount of distance penalization. Pseudocode for this procedure is provided in Alg. 2.

3.2. **Noisy measurements.** In Section 3.1, we assume the measurements are obtained in a noiseless manner, i.e., $Y_i = f_\theta(X_i)$ exactly. However, low-cost environmental sensors are known to obtain measurements corrupted by noise. Further, in most real-world scenarios, the measurements themselves are real valued and then discretized to values of 0 (below level set threshold) or 1 (above threshold). In this case, values obtained near the true level set boundary are more likely to be erroneous, since small perturbations of the measurement can result in an incorrect assignment. The work of [9, 12, 34] accounts for noisy binary-valued measurements by maintaining a posterior distribution over the change point θ and sampling at quantiles of this distribution. However, these assume both a constant search fraction and a constant probability of erroneous measurements (i.e., a bit flip with probability p). In this section, we show how the policy derived from FHS can be extended to handle continuous-valued measurements corrupted by Gaussian noise and prove that the resulting method converges almost surely to the true change point.

To handle noisy measurements, we utilize a probabilistic approach as in [9, 34], in which we sample a fraction into the posterior distribution on θ instead of the remaining interval. Beginning with a uniform prior over the change point, a posterior distribution $\pi_n(\theta)$ is obtained after each measurement via a Bayesian update. Consider sampling a fraction z into the distribution π_{n-1}, resulting in the measurement

Algorithm 2 Finite Horizon Search

Input: search fractions $z_1, \ldots, z_N$, stopping error ε
Initialize: $X_0 \leftarrow 0$, $Y_0 \leftarrow 1$, $a \leftarrow 0$, $b \leftarrow 1$, $n \leftarrow 1$
while $b - a > \varepsilon$ **do**
  if $n \leq N$ **then**
    $z \leftarrow z_n$
  else
    $z \leftarrow \frac{1}{2} - \frac{\lambda}{4}$
  end if
  if $Y_{n-1} = 1$ **then**
    $X_n \leftarrow X_{n-1} + z(b-a)$
  else
    $X_n \leftarrow X_{n-1} - z(b-a)$
  end if
  $Y_n \leftarrow f(X_n)$
  $a = \max\{X_i : Y_i = 1, i \leq n\}$
  $b = \min\{X_i : Y_i = 0, i \leq n\}$
  $\hat{\theta}_n \leftarrow \frac{a+b}{2}$
  $n \leftarrow n + 1$
end while

location X_n. In the case where $Y_n > \gamma$, the resulting update is

$$\pi_n(x) = \begin{cases} \frac{p_n}{z \circ p_n} \pi_{n-1}(x), & x \leq X_n \\ \frac{1-p_n}{z \circ p_n} \pi_{n-1}(x), & x > X_n, \end{cases} \tag{10}$$

where p_n is the probability of an erroneous binary measurement and

$$z \circ p := zp + (1-z)(1-p).$$

In this case, a positive measurement indicates that the change point likely lies to the right of X_n, but there is still a nonzero probability that the change point is to the left, due to the possible erroneous measurement. Similarly, for $Y_n < \gamma$, the Bayesian update becomes

$$\pi_n(x) = \begin{cases} \frac{1-p_n}{z * p_n} \pi_{n-1}(x), & x \leq X_n \\ \frac{p_n}{z * p_n} \pi_{n-1}(x), & x > X_n, \end{cases} \tag{11}$$

where

$$z * p := z(1-p) + (1-z)p.$$

After each update, the estimate $\hat{\theta}_n$ is taken to be the median of the resulting posterior distribution, and the expected absolute error is computed using the distribution π_n.

The above Bayesian updates assume binary-valued measurements and require the probability of an erroneous measurement. To handle more realistic sampling scenarios, we assume measurements are corrupted by zero-mean Gaussian noise, so that

$$Y_i = f(X_i) + \zeta_i \sim \mathcal{N}\left(f(X_i), \sigma^2\right), \tag{12}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. These measurements are then thresholded based on whether they are above or below the level set threshold γ. Let $\Phi(\cdot)$ denote the cumulative distribution function of

Algorithm 3 Probabilistic Finite Horizon Search

Input: search fractions $z_1, \ldots, z_N$, noise variance σ^2, stopping error ε
Initialize: $X_0 = 0$, $\pi_0(x) = 1$ for all $x \in [0,1]$, $n \leftarrow 1$
while $\mathbb{E}_{\theta \sim \pi_n} \left| \hat{\theta}_n - \theta \right| > \varepsilon$ **do**
  if $n \leq N$ **then**
    $z \leftarrow z_n$
  else
    $z \leftarrow \frac{1}{2} - \frac{\lambda}{4}$
  end if
  set $\tilde{X}_0, \tilde{X}_1$ such that

$$\int_0^{\tilde{X}_0} \pi_{n-1}(x) = z \quad \text{and} \quad \int_{\tilde{X}_1}^1 \pi_{n-1}(x) = 1 - z$$

  $X_n \leftarrow \arg\min_{X \in \{\tilde{X}_0, \tilde{X}_1\}} |X_{n-1} - X|$
  $Y_n \leftarrow f(X_n)$
  if $Y_n > \gamma$ **then**
    $p_n \leftarrow 1 - \Phi\left(\frac{Y_n - \gamma}{\sigma}\right)$

$$\pi_n(x) = \begin{cases} \frac{p_n}{z \circ p_n} \pi_{n-1}(x), & x \leq X_n \\ \frac{1-p_n}{z \circ p_n} \pi_{n-1}(x), & x > X_n \end{cases}$$

  else
    $p_n \leftarrow \Phi\left(\frac{Y_n - \gamma}{\sigma}\right)$

$$\pi_n(x) = \begin{cases} \frac{1-p_n}{z * p_n} \pi_{n-1}(x), & x \leq X_n \\ \frac{p_n}{z * p_n} \pi_{n-1}(x), & x > X_n \end{cases}$$

  end if
  $\hat{\theta}_n \leftarrow \text{median}_x \pi_n(x)$
  $n \leftarrow n + 1$
end while

a standard normal random variable. Under the measurement model (12), if we measure $Y_i < \gamma$ when $f(X_i) > \gamma$, an error occurs with probability $p_i = \Phi\left(\frac{Y_i - \gamma}{\sigma}\right)$. Similarly, if $Y_i > \gamma$ but $f(X_i) < \gamma$, an error occurs with probability $p_i = 1 - \Phi\left(\frac{Y_i - \gamma}{\sigma}\right)$. Note that in both cases, the probability of error p_i depends both on the noise variance σ^2 and the distance from the level set threshold. This is essential, as a measurement far from the level set threshold can handle much larger corruptions than one for which $|Y_i - \gamma|$ is small.

Our search procedure computes this noise level after each measurement, updating the posterior to reflect high uncertainty when samples are obtained near the change point. Note that for a given search fraction z_n, equal information is gained by moving to the z_n quantile of π_n or the $1 - z_n$ quantile. To account for the goal of minimizing the distance traveled, we move to the nearer of these two quantiles at each measurement. Finally, to ensure the algorithm always moves toward the median of the posterior, we follow the truncation approach of [34]. We refer to this algorithm as *probabilistic finite horizon search* (PFHS), and pseudocode is given in Alg. 3. In the case where $\sigma = 0$, PFHS is exactly equivalent to FHS.

Below we show that, given sufficiently many measurements, a discretized version of the PFHS algorithm converges almost surely to the true change point. This approach discretizes the unit interval into bins of width Δ and facilitates analysis more easily than the continuous version [11, 53]. We extend the analysis laid out in [34] to allow for varying noise level and varying step sizes.

Theorem 3.4. *Assume measurements are obtained following the noise model (12). Then a discretized version of the PFHS algorithm converges almost surely to the true change point.*

Proof. We wish to show that for any $\varepsilon > 0$ and any $\theta \in [0,1]$

$$\Pr\left(\limsup_{n\to\infty} \sup_{\theta\in[0,1]} \left|\hat{\theta}_n - \theta\right| > \varepsilon\right) = 0. \tag{13}$$

For any $\varepsilon > 0$, set $\Delta < \varepsilon$ such that $\Delta^{-1} \in \mathbb{N}$ and consider a discretized probabilistic search algorithm with bin size Δ. By [34], for any $\Delta > 0$, the discretized probabilistic search algorithm that samples a fraction z into the posterior with a noise level $p < 1/2$ satisfies

$$\sup_{\theta\in[0,1]} \Pr\left(|\hat{\theta}_n - \theta| > \Delta\right) \leq \frac{1-\Delta}{\Delta} t(z)^n, \tag{14}$$

where $\alpha = \sqrt{p}/(\sqrt{p} + \sqrt{1-p})$ and

$$t(z) := \frac{1-p}{2(1-\alpha)} + \frac{p}{2\alpha} + \left(\frac{1-p}{2(1-\alpha)} - \frac{p}{2\alpha}\right)(1-2\alpha)\,(1-2z)\,. \tag{15}$$

We first show that $t(z) < 1$ as long as $z > 0$ and $p < 1/2$. Algebraic manipulations indicate that for $p < 1/2$,

$$\frac{1-p}{2(1-\alpha)} + \frac{p}{2\alpha} + \left(\frac{1-p}{2(1-\alpha)} - \frac{p}{2\alpha}\right)(1-2\alpha) = 1.$$

Since $z > 0$, we have that $1 - 2z < 1$, and therefore (15) is of the form $a + bc$, where $a, b > 0$, $a + b = 1$, and $c < 1$. We wish to show that $a + bc < 1$. Using the fact that $b = 1 - a$, an equivalent statement is

$$a + (1-a)c < 1 \iff (1-a)c < 1 - a,$$

which holds as long as $c < 1$.

Next, observe that since $t(z) < 1$, for any $\theta \in [0,1]$

$$\begin{aligned}\sum_{n=1}^{\infty} \Pr\left(|\hat{\theta}_n - \theta| > \Delta\right) &\leq \sum_{n=1}^{\infty} \frac{1-\Delta}{\Delta} t(z)^n \\ &= \frac{1-\Delta}{\Delta}\left(\frac{1}{1-t(z)} - 1\right) < \infty.\end{aligned}$$

By the Borel-Cantelli lemma, this guarantees that (13) holds. Finally, let $z = \min_i \{z_i\}_{i=1}^N$ and p be the maximum noise level observed throughout the sampling procedure. For $\lambda < 2$, we have $z > 0$. Further, since p is computed by taking the tail of a Gaussian distribution in the direction away from the mean, we have $p < 1/2$. This guarantees convergence of the discretized form of the PFHS algorithm. □

The above result also holds for the more general case of sub-Gaussian noise with parameter σ, where the probability of error p_i is bounded via Hoeffding's inequality

$$p_i \leq \mathbb{P}\left\{|Y_i - f(X_i)| \geq |Y_i - \gamma|\right\} \leq 2\exp\left(-\frac{|Y_i - \gamma|^2}{2\sigma^2}\right). \tag{16}$$

In this case, Thm. 3.4 requires the additional assumption that $|Y_i - \gamma| \geq \sigma\sqrt{2\log(4)}$ to ensure that $p \leq 1/2$, i.e., the more general case requires a greater gap between the measurement and the level set threshold. If this condition is not met, repeated measurements could be obtained until the probability of error falls below $1/2$.

The analysis of probabilistic search algorithms has been a topic of significant study over more than fifty years [26, 9, 41, 30, 39, 53, 50]. While the search fractions used in PFHS are derived from the noiseless setting and therefore suboptimal, deriving an optimal policy for the noisy case is intractable due to the combinatorial explosion of potential posterior distributions. Determining approximate solutions for the noisy case via reinforcement learning is an important topic that lies beyond the scope of this work.

3.3. **Gaussian process level set estimation.** Given a means of handling noisy measurements, we now consider the problem of LSE in the case where the underlying function f is a GP. Formally, a GP is a collection of random variables, one for each value of $f(x)$, for which every finite subset forms a Gaussian random vector [42]. A GP is characterized by its mean function $\mu(x) = \mathbb{E}[f(x)]$ and its covariance/kernel function $k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$, which governs the smoothness of the function over the domain, which in our case is $[0, 1]^2$. In the GP-LSE problem, we assume measurements are corrupted by Gaussian noise, so that $Y_i = f(X_i) + \zeta_i$ with $\zeta_i \sim \mathcal{N}(0, \sigma^2)$. After obtaining measurements $Y_1, \ldots, Y_n$ at corresponding locations $X_1, \ldots, X_n$, the posterior mean and covariance can be obtained as

$$\begin{aligned} \mu_n(x) &= k_n(x)^T (K_n + \sigma^2 I)^{-1} y_n \\ k_n(x, x') &= k(x, x') - k_n(x)^T (K_n + \sigma^2 I)^{-1} k_n(x'), \end{aligned}$$

where $k_n(x) = \begin{bmatrix} k_n(x, X_1) & k_n(x, X_2) & \ldots & k_n(x, X_n) \end{bmatrix}^T \in \mathbb{R}^n$, $K_n \in \mathbb{R}^{n \times n}$ is the positive-definite kernel matrix whose i, jth entry is $k_n(X_i, X_j)$, and $y_n \in \mathbb{R}^n$ is the vector of measurements. The GP model is frequently used in environmental applications [15, 56, 35] (often referred to as *kriging* in this context [47]), and the ability to measure posterior variance has led to a number of approaches to active learning in GPs [21, 24, 25, 7, 27]. To apply our proposed FHS to the GP-LSE problem, we consider a subset of GPs wherein one coordinate of the level set boundary is a function of the other coordinate, as depicted in Fig. 3. This assumption is similar to the *boundary fragment* assumption, which has been widely studied within the nonparametric active learning literature [43, 12, 31]. This assumption reduces the superlevel set to a single, simply-connected region, which commonly holds in environmental applications [13, 58].

Under the above assumption, our approach to GP-LSE is as follows. Assume the level set boundary ∂S is a function of one coordinate. We split the unit interval into a series of transects and perform a one-dimensional PFHS to localize ∂S along each transect, initializing transects as described in Sec. 3.3.1. This process is depicted in Fig. 3. As a result, our goal is to estimate the one-dimensional function ∂S assuming it is a GP. Along each transect, we run PFHS and treat the estimated change point as a noisy measurement of ∂S at the transect location. Clearly the accuracy of

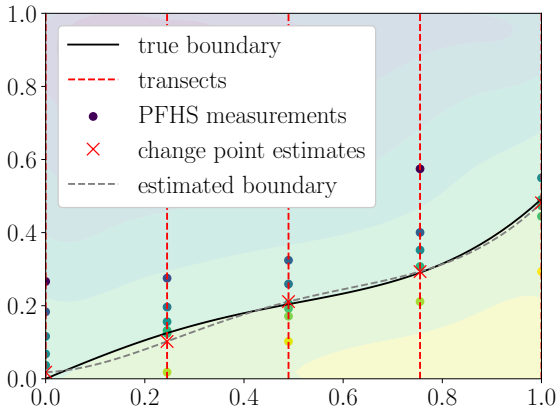


FIGURE 3. Example Gaussian process level set estimation when the boundary is a function of the first coordinate. The unit interval is split into five equally-spaced transects, and a PFHS procedure is used to localize the change point along each transect. Change point estimates are then used to estimate the boundary via GP regression.

estimating ∂S is governed by the number of transects and the accuracy of localizing the change point along each transect. In this case, the number of transects governs the approximation error and the stopping error along each transect defines the estimation error.

Thm. 3.4 shows that PFHS converges almost surely for a single transect. To obtain an ε-accurate guarantee similar to those provided in [21, 7], further assumptions must be made on the boundary ∂S. For example, if one assumes the boundary is L-Lipschitz on the unit interval, we obtain the following result.

Theorem 3.5. *Assume the level set boundary ∂S is L-Lipschitz in the first coordinate, and let $\widehat{\partial S}$ be the estimate of the level set boundary across B transects, as described in Sec. 3.3. Take the number of samples per transect n to be such that $\frac{1-\Delta}{\Delta} t(z)^n \leq \frac{\delta}{B}$, where $t(z)$ is defined in the proof of Thm. 3.4. Then with probability at least $1-\delta$, we have*

$$\left\| \widehat{\partial S} - \partial S \right\|^2 \leq \left(\Delta + \frac{L}{2B} \right)^2 \tag{17}$$

Proof. First, consider a search along a fixed transect. As shown in (14), after n measurements along the transect we have

$$\Pr\left(|\hat{\theta}_n - \theta| > \Delta \right) \leq \frac{1-\Delta}{\Delta} t(z)^n,$$

where $t(z) < 1$. To obtain an error of at most Δ over all B transects simultaneously with probability at most $1-\delta$, we set

$$\frac{1-\Delta}{\Delta} t(z)^n \leq \frac{\delta}{B}$$

and apply the union bound.

Next, let $I_1, I_2, \ldots, I_B$ partition the unit interval into B subsets of length $1/B$. Under the Lipschitz assumption, the squared error can be written as

$$\begin{aligned}
\left\| \widehat{\partial S} - \partial S \right\|^2 &= \int_0^1 \left| \partial S(t) - \widehat{\partial S}(t) \right|^2 dt \\
&= \sum_{b=1}^{B} \int_{I_b} \left| \partial S(t) - \widehat{\partial S}(t) \right|^2 dt \\
&\leq \sum_{b=1}^{B} \frac{1}{B} \left(\Delta + \frac{L}{2B} \right)^2 \\
&= \left(\Delta + \frac{L}{2B} \right)^2,
\end{aligned}$$

where the inequality follows from the fact that each transect has an error of at most Δ and ∂S is L-Lipschitz. □

Thm. 3.5 gives clear insight into the impact of the stopping error Δ and number of transects B, showing that each can be used to reduce the overall error. In our experiments, we tune these parameters though a grid search, and a theoretical characterization of the optimal balance between these parameters is an important topic of future study. Finally, the theorem implies a total sample complexity of

$$\tilde{n} \geq B \frac{\log\left(\frac{\delta}{B}\frac{\Delta}{1-\Delta}\right)}{\log(t(z))}.$$

To gain insight into the above, consider probabilistic binary search, where $z = 1/2$. In this case, as $p \to 0$, $t(z) \to 1/2$, and as $p \to 1/2$, $t(z) \to 1$. In either case, $\log(t(z))$ converges to a negative absolute constant, and the sample complexity $\tilde{n} = O\left(B \log\left(\frac{B}{\delta\Delta}\right)\right)$, where we make the approximation $1 - \Delta \approx 1$ for small Δ. While the Lipschitz assumption of Thm. 3.5 may be overly simple, it provides insight into the relationship between the stopping error, number of transects, and number of measurements required in our approach.

3.3.1. *Initialization and policy calculation.* Rather than beginning the search from the origin at each transect, we make use of information from previous transects and initialize the search from the previous change point estimate. In the noiseless case, this initial sample reduces the interval size, and we then compute the optimal policy for the resulting interval length using Alg. 1. In the noisy case, the initial sample alters the distribution π_n, and we wish to derive an equivalent notion of interval reduction in order to determine the appropriate policy for each transect. Recall that in the noiseless case, the length of the feasible interval corresponds to the exponentiated differential entropy. We therefore compute the *effective interval size* based on the exponentiated differential entropy after one sample has been obtained. Let X_0 denote the initial sample location and p denote the corresponding derived probability of error. Following the update equations (10) and (11), in the case where $Y_0 > \gamma$, we have

$$\begin{aligned}
H_0 = &- \int_0^{X_0} \frac{p}{X_0 \circ p} \log\left(\frac{p}{X_0 \circ p}\right) d\theta - \\
&\int_{X_0}^1 \frac{1-p}{X_0 \circ p} \log\left(\frac{1-p}{X_0 \circ p}\right) d\theta
\end{aligned}$$

$$= \log\left(X_0 \circ p\right) - \frac{1}{X_0 \circ p}\left(pX_0 \log(p) + (1-p)(1-X_0)\log(1-p)\right).$$

Exponentiating gives

$$e^{H_0} = (X_0 \circ p)\left(p^{-pX_0/(X_0 \circ p)}(1-p)^{-(1-p)(1-X_0)/(X_0 \circ p)}\right).$$

Similarly, in the case where $Y_0 < \gamma$, we have

$$e^{H_0} = (X_0 * p)\left(p^{-p(1-X_0)/(X_0 * p)}(1-p)^{-(1-p)X_0/(X_0 * p)}\right).$$

This generalizes the notion of interval length to the case of noisy measurements, and when $p = 0$ is exactly equal to the resulting interval length. For each transect, we obtain the initial measurement, compute the corresponding effective interval size (exponentiated entropy), and then compute the policy via Alg. 1 using $L = e^{H_0}$ as the initial interval length.

4. Simulations & experiments.

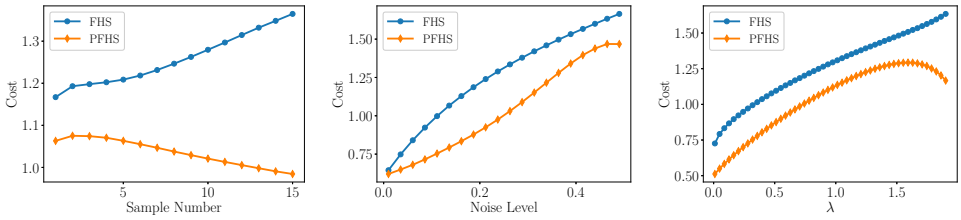
4.1. Performance on one-dimensional step functions.

In this section, we verify the performance of the proposed FHS and PFHS policies. We first demonstrate the reduced cost (as defined by Eq. 2) using PFHS compared to FHS. We then benchmark FHS and PFHS against the existing QS and UTB algorithms in a time-penalized search scenario.

4.1.1. *Comparison of FHS and PFHS.* We first compare the performance of FHS and PFHS in the case of binary measurements that are erroneous with constant probability p. We report performance over fifteen measurements, considering twenty noise levels between 0.01 and 0.49, and fifty values of λ between 0.01 and 1.9. For each configuration, we perform searches over 100 uniformly-spaced values of θ in the interval $[0, 1]$, running 100 Monte Carlo simulations for each value of θ, and report the average cost as defined by (2). In this noisy setting, the interval length for FHS does not reflect the uncertainty in the change point due to the erroneous measurements. By [34, Thm. 1], the entropy corresponds to four times the absolute error in the change point. Hence, in the noisy setting we report the algorithm cost as

$$J(z_1, \ldots, z_N) = \mathbb{E}_\theta\left[4\left|\hat{\theta}_N - \theta\right| + \lambda D_N\right]. \tag{18}$$

Fig. 4 shows the cost as a function of sample number (averaged over p, λ), noise level (averaged over N, λ), and tuning parameter λ (averaged over p, N) for each algorithm. First, we see that the benefits of PFHS are most apparent as more samples are obtained due to the convergent behavior of PFHS. For FHS, the error in estimating θ can actually increase with more samples, since one erroneous measurement can bias the estimate away from the true value. Second, while no clear trend in improvement versus noise level is seen, the greatest *percent* improvement is obtained at noise levels below 0.2. This behavior is likely due to the fact that PFHS uses the policy derived from the noiseless case, whose suboptimality is more apparent as the noise level increases. Third, the benefits of PFHS are more apparent as λ increases. A closer inspection of entropy and distance reveals that for large λ, both algorithms travel a similar distance (making only small movements), but PFHS has a much lower entropy due to its ability to incorporate knowledge of the



(A) (B) (C)

FIGURE 4. Performance of FHS and PFHS in the case of noisy measurements, where cost is defined by (2). (a) Cost as a function of number of samples. (b) Cost as a function of noise level. (c) Cost as a function of tuning parameter λ. The performance improvement is most significant when more measurements are taken and for large λ.

noise level. In all cases, PFHS outperforms FHS, with an average cost reduction ranging from 6% at $p = 0.01$ to 27% at $p = 0.14$.

4.1.2. *Cost as a function of sampling time.* To minimize the total time that a vehicle takes to complete a search, we consider a cost function of the form

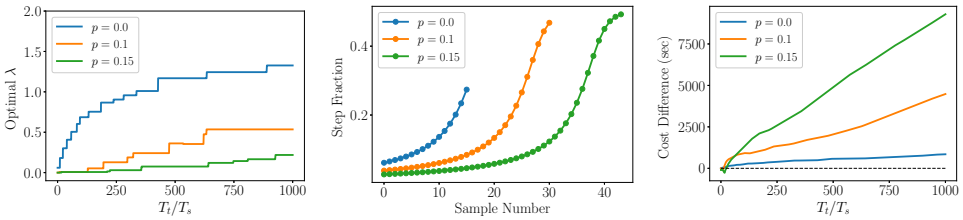
$$J_T(z_1, \ldots, z_N) = T_s N + T_t D, \tag{19}$$

where T_s and T_t represent the time per sample and time per unit distance traveled, respectively, and N and D represent the number of samples and total distance. In order to minimize this cost in expectation, we first calculate the number of samples, N_λ, and total distance, D_λ, expected for the optimal policy for each value of λ to reach a final interval size smaller than desired error ε using Alg. 1. We then select the value of λ that minimizes the total search time,

$$\lambda^* = \arg\min_\lambda T_s N_\lambda + T_t D_\lambda. \tag{20}$$

For the noisy setting, the expected interval size cannot be computed in closed form. We instead evaluate the sample mean of the interval size, computed over a range of 1,000 values of $\theta \in [0, 1]$ and 100 Monte Carlo trials for each value of θ.

We compare the performance of the above method with the existing probabilistic QS (PQS) algorithm for distance-penalized search in one dimension, where we compute the optimal parameter for PQS according to (20). We consider the same grid of 1,000 values of θ for 1,000 different ratios of T_t/T_s in the range of 1×10^{-4} to 1×10^3, taking $T_s = 100$ as the base sampling cost. Fig. 5(a) shows the value of λ^* selected by (20) for noise level $p \in \{0, 0.1, 0.15\}$. As expected, as T_t increases, a higher value of λ^* is selected, taking more samples while being less likely to overshoot the change point. Additionally, as the noise level increases, lower values of λ^* are selected. This results in a search that favors entropy reduction in order to account for the information loss incurred by noisy measurements. Fig. 5(b) shows the PFHS policies selected for each noise level at a ratio $T_t/T_s = 250$. As expected, more samples are required as the noise level increases, with the lower values of λ resulting in a larger maximum step size. The ability of PFHS to utilize small step fractions at early stages allows the algorithm to keep the total distance traveled low while still converging rapidly. Fig. 5(c) shows the cost difference between PQS



(A) (B) (C)

FIGURE 5. Performance of noisy search algorithms for varying noise level p when optimized for total sampling time as defined by (19). (a) Optimal tuning λ as a function of the ratio of travel time T_t to sampling time T_s. (b) Policies selected by PFHS for each noise level considered. (c) Cost difference between PQS and proposed PFHS algorithms. As the noise level increases, PFHS favors entropy reduction (smaller λ) over distance penalization.

and PFHS for each noise level. In nearly all cases, PFHS outperforms PQS, with a greater difference as both the noise level and the ratio T_t/T_s increase. Although difficult to see, PQS does outperform PFHS by a small amount for a noise level of $p = 0.15$ and a ratio of $T_t/T_s \leq 24$. However, PQS obtains a performance improvement of less than 4%, whereas FHS obtains as much as a 15% improvement as the ratio of travel to sample time increases.

4.2. **Performance on GP-LSE.** Next, we examine the performance of our approach to GP-LSE using the proposed PFHS algorithm. As a benchmark, we compare to the state-of-the-art Truncated Variance Reduction (TruVaR) algorithm [7], which is designed explicitly for cost-sensitive GP-LSE. To perform LSE, TruVaR maintains estimates of the super-level and sub-level sets, as well as a third set for points whose level set membership is uncertain. Points are placed into the super/sub-level set estimates only after the algorithm is sufficiently confident they lie above/below the level set threshold, with confidence estimates being obtained from the GP model. TruVaR selects samples based on the ratio of (truncated) variance reduction to cost

$$X_n = \arg\max_{x \in S} \frac{\tau(x)}{c(x)}, \tag{21}$$

where $\tau(x)$ is the truncated variance reduction after sampling location x

$$\tau(x) = \sum_{\bar{x} \in M_{n-1}} \max\left\{\beta_{(i)}\sigma^2_{n-1}(\bar{x}), \eta^2_{(i)}\right\} - \sum_{\bar{x} \in M_{n-1}} \max\left\{\beta_{(i)}\sigma^2_{n-1|x}(\bar{x}), \eta^2_{(i)}\right\}, \tag{22}$$

M_{n-1} denotes the set of points whose super-/sublevel set membership is currently uncertain, $\beta_{(i)}$ is a confidence parameter for the ith epoch of the algorithm, $\sigma^2_{n-1}(\bar{x})$ is the posterior variance at the point $\bar{x}$ after obtaining $n-1$ measurements, $\sigma^2_{n-1|x}(\bar{x})$ is the posterior variance after obtaining $n-1$ measurements in addition to sampling at location x, and $\eta_{(i)} > 0$ is a truncation parameter determined by the algorithm. In [7], simulations are performed on chlorophyll concentration data from Lake Zürich, and the authors use a sampling cost that considers both distance traveled and sensor depth. Since we do not consider sensor depth, we set

$c(x) = |X_{n-1} - x|$, i.e., the distance from the current location to the potential sampling location x.

We consider two-dimensional GP realizations with level set boundaries satisfying the assumption described in Section 3.3. In all simulations, we provide TruVaR with the true kernel parameters used to generate the two-dimensional GP realization under consideration. We set the parameter $\beta_{(i)} = a \log\left(M t_{(i)}^2\right)$ as in [7], where $t_{(i)}$ denotes the time at which the epoch starts, M denotes the number of elements in the two-dimensional realization, and a is a constant. We found the best performance resulted from setting $a = 0.0001$. All other parameters were set according to the recommendations in [7]. We measure the error in terms of the symmetric difference between the true and estimated super-level set divided by the total number of points in the realization, i.e.,

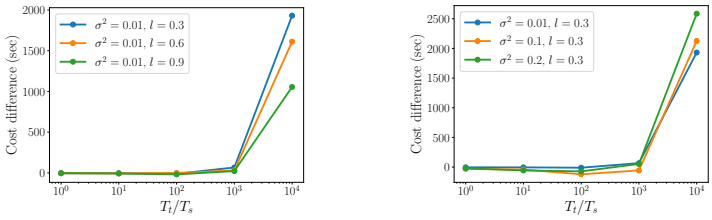
$$E = \frac{1}{M}\left|S \triangle \hat{S}\right|, \tag{23}$$

where M is the total number of points in the realization and $\hat{S}$ denotes the estimated super-level set. When considering the LSE problem as binary classification, the above may be viewed as the classification error.

4.2.1. *Synthetic GP data.* We first compare algorithm performance on synthetic GP data. To ensure the boundary assumption described in Section 3.3 holds, we begin by generating a one-dimensional GP realization defining the level set boundary. We then generate a two-dimensional GP realization by taking the true value to be the positive or negative distance from the boundary, obtaining 500 measurements corrupted by Gaussian noise with variance 0.0001. Finally, we fit a two-dimensional GP to the obtained measurements, which is then treated as the true function value $f(x)$ over the M points in the realization. An example boundary and two-dimensional realization are shown in Fig. 3. For both the one-dimensional boundary and the two-dimensional realization, we use the radial basis function (RBF) kernel. For the boundary, we consider lengthscales of 0.3, 0.6, and 0.9 to simulate varying degrees of smoothness in ∂S. The two-dimensional realization is always fit using a lengthscale of 0.1. Both PFHS and TruVaR are given the true kernel parameters when performing GP-LSE. In all experiments, we generate realizations of size $M = 21 \times 20$; this size is chosen largely due to the high computation time required by TruVaR. To test our approach to handling noisy measurements, we consider noise variances $\sigma^2 \in \{0.01, 0.1, 0.2\}$, which govern the additive noise on top of $f(x)$ as modeled by (12).

To select the number of transects and stopping error for PFHS, we generate 100 examples of GP realizations and corresponding level set boundaries using the above procedure and perform a grid search over both parameters. We then select the parameters that give the lowest total cost while maintaining an average error over all 100 realizations below 8% for noise variances $\sigma^2 \in \{0.01, 0.1\}$ and an error below 11% for $\sigma^2 = 0.2$.[1] This procedure is used to select the best parameters for each lengthscale and noise level under consideration, and an empirical evaluation of performance as a function of number of transects and stopping error can be found in Appendix B. Note that an equivalent procedure would be required to select the GP kernel and bandwidth parameters for TruVaR; however, to avoid the large computational cost of tuning these parameters, we provide TruVaR with the

[1]The latter choice was made to match the accuracy achieved by TruVaR for the given parameters.



(A) (B)

FIGURE 6. Difference in cost between TruVaR and proposed PFHS for GP-LSE on synthetic data. (a) Fixed noise level σ^2 and varying lengthscale l. (b) Varying noise level σ^2 and fixed lengthscale l. The proposed PFHS obtains the most significant benefit when the ratio of travel time to sample time (T_t/T_s) is large.

true kernel used to generate the GP realizations, not seen in the validation stage for selecting transects and stopping error parameters. Finally, we compare both algorithms on 100 separate GP realizations for each lengthscale. Since the resulting errors (23) are nearly identical, we compare the sampling cost between the two approaches.

Fig. 6 displays the average cost difference between TruVaR and PFHS over the 100 random realizations, showing the cost difference as a function of the ratio T_t/T_s for varying values of (a) lengthscale and (b) noise variance. In both cases, we see that for high ratios of travel-to-sample time, FHS outperforms TruVaR by a significant margin, while for low ratios, the performance is comparable. Fig. 6(a) shows that this improvement reduces with lengthscale, indicating that PFHS excels when the boundary is least smooth. Fig. 6(b) shows the improvement for different noise levels and indicates that PFHS obtains the largest improvement for higher noise levels. Although difficult to see from the figures, TruVaR does outperform FHS for $T_t/T_s \in \{1, 10, 100\}$. However, the mean and maximum improvement are 58 sec and 269 sec, respectively, whereas PFHS achieves a mean/maximum improvement of 1200/3900 sec. Further, PFHS typically obtains an error that is 1-4% lower than that of TruVaR for these values of T_t/T_s, indicating a more careful tuning of parameters may allow PFHS to obtain better performance. Hence, while FHS is most beneficial when travel time is significant relative to measurement time, it is still competitive even for low values of T_t. Further performance comparison and box plots can be found in Appendix B.

Finally, we comment that one additional drawback of TruVaR is that of computational complexity. To perform sample selection, TruVaR must compute the posterior variance after sampling for every location in the set of uncertain points. As a result, the average computation time for each search in the above experiments was 2.45 sec for TruVaR compared to 0.34 sec for PFHS. While both times are sufficiently small for practical applications, we remark that we chose a small realization size (21×20) that would result in limited resolution over large spatial regions. This consideration is especially important when attempting to deploy adaptive sampling algorithms on low-cost mobile sensing devices, which will have limited resources for computation and power. Investigation of speedups obtained by approximations of the posterior variance or through specialized hardware is an important topic

	Sampling Time (s) Velocity (km/hr)	8 32	8 65	30 32	30 65
PFHS	Search Cost (hr) Error (%)	9.58 2.87	4.90 3.26	10.46 3.16	5.54 3.37
TruVaR	Search Cost (hr) Error (%)	25.1 14.62	12.75 14.56	24.88 14.30	12.96 14.86

TABLE 1. Total sampling time (in seconds) and estimation error in air quality data following the Camp Fire in November 2018. Region considered and example sampling pattern of PFHS are depicted in Fig. 1. For all sampling times and velocities considered, PFHS achieves a significant reduction in both cost and estimation error.

for future research that would make GP-based approaches such as TruVaR more competitive in terms of computation time.

4.2.2. *Air quality data.* Finally, we compare the performance of PFHS, FHS, and TruVaR on real air quality data obtained from the AirNow database [52]. One potential application of LSE approaches is to provide a high-quality estimate of regions containing high levels of particulate matter. Of particular importance is the problem of rapidly estimating such regions during major wildfire events, such as the 2018 Camp Fire in California [51] or the more recent series of wildfires impacting the western U.S. in 2020, which resulted in the worst air quality in the world for major cities such as Portland, OR and San Francisco, CA [40].

We consider PM 2.5 data from November 18, 2018, using 124 sensors in the region of Butte County, CA. Since the measurements are spatially sparse, we interpolate the values using two-dimensional GP regression with a summation of RBF and bias kernels optimized and implemented via the GPy package [22]. We set the threshold at 100 μg / m^3, which corresponds to the "unhealthy for sensitive groups" level according to the AirNow standards [52]. We perform LSE over the region depicted in Fig. 1(b), which is approximately 111 km per side. We consider sampling times of 8 and 30 seconds, corresponding to the extremes of the settling time of the Sensirion SPS30 particulate matter sensor [44]. This sensor has a precision of ± 10 μg/m^3; treating errors uniformly throughout this range, we set the noise variance of the GP to that of a uniform distribution with support $[-10, 10]$, resulting in $\sigma^2 = 20^2/12$. This choice minimizes the Kullback-Leibler divergence between the uniform and normal distributions when computed over the support of the uniform distribution. Finally, we consider travel times of 32 km/hr and 65 km/hr based on the maximum speed of the DJI Matrice 600 UAV. We provide TruVaR with the two-dimensional kernel used to perform GP regression over the realization and set the parameter $a = 6$, as we found that the recommended parameter $a = 1$ resulted in very high estimation errors for the high noise variance considered. For PFHS, we model the boundary using a one-dimensional GP with RBF kernel having lengthscale and variance both set to unity. We search over five transects and set the stop error for each transect to 0.03.

Table 1 shows the resulting search cost and error for PFHS and TruVaR in the four scenarios considered. Compared with the results on synthetic data, we see that the sampling time required by TruVaR is significantly greater than that of PFHS in this high-noise regime. In all cases, PFHS achieves a lower cost and

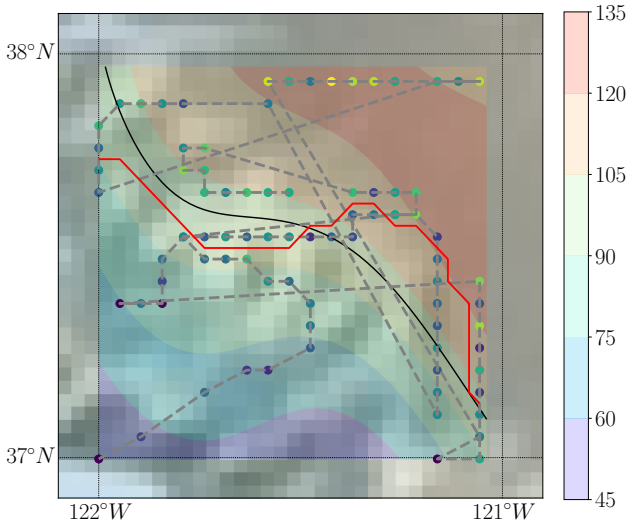


FIGURE 7. Map of sample locations and trajectory followed by TruVaR algorithm on Camp Fire data. Dots denote sample locations with measurement values indicated by their color, red solid line is estimated boundary, and gray dashed line is path traversed by sensor. Compared to PFHS (see Fig. 1(b)), TruVaR does not sufficiently penalize distance in this noisy regime.

lower estimation error, typically yielding an estimation error approximately one fifth that of TruVaR at less than half the cost. We display the sample locations and path traveled by TruVaR in Fig. 7. In this high-noise regime, TruVaR places too much emphasis on variance reduction and fails to appropriately penalize for distance traveled. Although not pictured, we also tested the lower-noise regime and found TruVaR to be more competitive in this setting, focusing samples near the level set boundary and traveling a smaller distance. Further, we note that PFHS relies on the assumption that the superlevel set is a single, connected region with a boundary that can be written as a function of one coordinate.

While this assumption is realistic in the case of tracking a wildfire front, it may not be appropriate in other settings (e.g., that considered in [7]), and TruVaR has the added flexibility of discovering superlevel sets consisting of multiple disjoint regions. Hence, we consider an additional region covering $[38°N, 39°N] \times [121°W, 122°W]$, which contains two distinct boundaries. We apply PFHS assuming that the boundary is a function of the vertical coordinate (latitude), using the same stopping error and number of transects as before. In this case, PFHS only discovers one of the two boundaries, resulting in an average error of almost 30%. For comparison, we report the cost required by TruVaR both to achieve an error of approximately 15% and an error of 30%. The average sampling costs and errors over 100 trials are given in Table 2. TruVaR is able to capture both boundaries, resulting in a much lower error at the cost of significantly greater sampling time. In the high error setting, TruVaR requires slightly more sampling time than PFHS and obtains an estimate that is slightly worse on average. Combining these two approaches to allow for the flexibility of TruVaR with the efficiency of PFHS is therefore an important topic of future research.

	Sampling Time (s) Velocity (km/hr)	8 32	8 65	30 32	30 65
PFHS	Search Cost (hr) Error (%)	5.07 29.43	2.73 29.13	5.96 29.18	3.33 29.32
TruVaR (low error)	Search Cost (hr) Error (%)	28.15 14.51	14.38 15.48	29.33 14.98	14.63 15.18
TruVaR (high error)	Search Cost (hr) Error (%)	6.04 32.29	2.93 32.09	6.58 30.70	3.29 30.97

TABLE 2. Total sampling time (in seconds) and estimation error in air quality data that does not meet the assumptions of PFHS for two-dimensional sampling. TruVaR has the potential to obtain much lower error in this setting but requires more time to achieve an error similar to PFHS.

5. **Conclusions & future work.** We have presented a finite-horizon approach to sensing the change point of a one-dimensional step function that optimally balances the distance traveled and number of samples acquired. We have shown that the resulting policy can be obtained in closed form, making it easily deployable on mobile sensors such as those mounted on a UAV. Aside from outperforming heuristic methods for one-dimensional search, our proposed FHS algorithm outperforms existing methods on the problem of Gaussian process level set estimation under certain assumptions on the level set boundary.

Our approach to two-dimensional sampling requires localizing the change point over a series of transects. While we optimized both the number of transects and the error per transect numerically, an important open problem is determining the optimal values of these quantities analytically. Another important next step is to incorporate other realistic vehicle costs, such as acceleration and battery life, into the policy calculation.

Appendix A. Proofs of technical results.

Lemma A.1. *Let H_N be the entropy of the posterior distribution after N measurements, so that $H_0 = 0$. Under the conditions of Theorem 3.1, we have*

$$\mathbb{E}\left[e^{H_N}\right] = \prod_{i=1}^{N} \left(z_i^2 + (1 - z_i)^2\right). \tag{24}$$

Proof. First note that under the uniform distribution on the unit interval, the exponentiated differential entropy is the length of the feasible interval after N samples. The proof will proceed by induction on N. Consider the base case, $N = 1$, for which it is trivial to show that

$$\mathbb{E}\left[e^{H_1}\right] = z_1^2 + (1 - z_1)^2 = \xi_1.$$

Now assume that (24) holds for some $N \in \mathbb{N}$. Sampling some fraction z_{N+1} into the remaining feasible interval e^{H_N} results in two potential entropies

$$e^{H_{N+1}} = \begin{cases} z_{N+1} e^{H_N}, & \text{w.p.} \quad z_{N+1} \\ (1 - z_{N+1}) e^{H_n}, & \text{w.p.} \quad 1 - z_{N+1}. \end{cases}$$

Therefore

$$\begin{aligned}\mathbb{E}[e^{H_{N+1}}] &= z_{N+1}^2\mathbb{E}[e^{H_N}] + (1-z_{N+1})^2\mathbb{E}[e^{H_N}] \\ &= \left(z_{N+1}^2 + (1-z_{N+1})^2\right)\mathbb{E}[e^{H_N}] \\ &= \prod_{i=1}^{N+1}\left(z_i^2 + (1-z_i)^2\right).\end{aligned}$$

□

Proof of Lemma 3.3. Define $z_{1:n} = z_1, \ldots, z_n$. Using the hypothesis of Lemma 3.3, For any $z_{1:n}$,

$$J(z_{1:n}) \geq J(z_{1:n-1}, z_n^*) \geq J(z_{1:n-2}, z_{n-1:n}^*) \geq \ldots \geq J(z_{1:n}^*). \tag{25}$$

□

Appendix B. Additional experimental results.

B.1. **Impact of transects and stopping error on PFHS.** In Figs. 8 - 11 below, we depict the super-level set estimation error (23) and sampling cost as a function of the number of transects and the stopping error used by PFHS on the synthetic GP data considered in Sec. 4.2.1. For each figure, the top row depicts results for a noise level of $\sigma^2 = 0.01$, and the bottom row uses a noise level of $\sigma^2 = 0.2$. The travel time increases per column, with values of $T_s = 1, 100$, and $10,000$. The horizontal orange line denotes the median, boxes indicate the inter-quartile range (IQR), whiskers cover 1.5 times the IQR, and circles denote values outside this range.

Fig. 8 shows the error as a function of the number of transects used for a fixed transect stopping error of 0.05. As stated in the text, we see that the error decreases with the number of transects, with minimal decrease in error after six transects. Fig. 9 shows the resulting sampling cost for the same scenarios and indicates a linear increase in sampling cost with the number of transects. For both the low- and high-noise settings, the error IQR remains stable and relatively small. As depicted in Fig. 9, the sampling cost IQR increases for larger travel times (right column) but remains approximately the same fraction of the total cost for all values of T_t. All settings show a small number of outliers, indicating stable performance across the range of parameter values.

In Figs. 10-11, we show the error and cost, respectively, as a function of the stopping error per transect. PFHS exhibits the expected tradeoff, with a larger per-transect error resulting in a higher overall error but requiring less sampling time.

B.2. **Box plots for synthetic data.** Below we display box plots for both PFHS and TruVaR on the synthetic test data from Sec. 4.2.1. Fig. 12 shows the cost for a fixed measurement noise of $\sigma^2 = 0.01$, with PFHS performance shown on the top row and TruVaR on the bottom row. The lengthscale increases per column, with values of $l = 0.3, 0.6$ and 0.9. Fig. 13 shows the cost for a fixed lengthscale $l = 0.3$, with the noise level increasing per column. Both figures show that PFHS has a higher variability than TruVaR, with the greatest variation in interquartile range at the highest lengthscale. However, PFHS still outperforms TruVaR across the majority of Monte Carlo trials. For example, for a ratio of $T_t/T_s = 10,000$ and measurement noise $\sigma^2 = 0.01$, the PFHS cost is lower than the median TruVaR cost in 85, 83, and 75 of 100 trials for lengthscales of 0.3, 0.6, and 0.9, respectively.

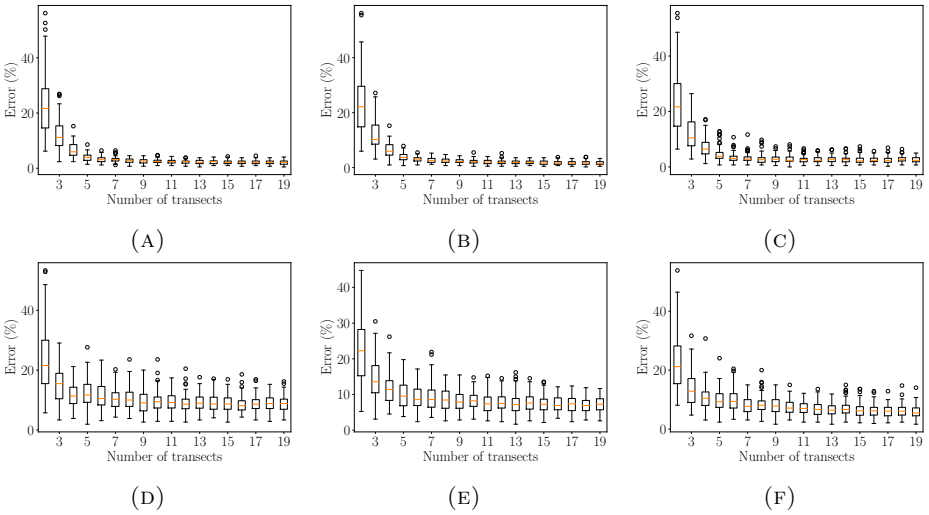


FIGURE 8. Error in estimating super-level set as a function of number of transects for PFHS on synthetic GP data considered in Sec. 4.2.1. Stopping error per transect is fixed to 0.05. Top row: Increasing travel time for measurement noise level $\sigma^2 = 0.01$. Bottom row: Increasing travel time for measurement noise level $\sigma^2 = 0.2$.

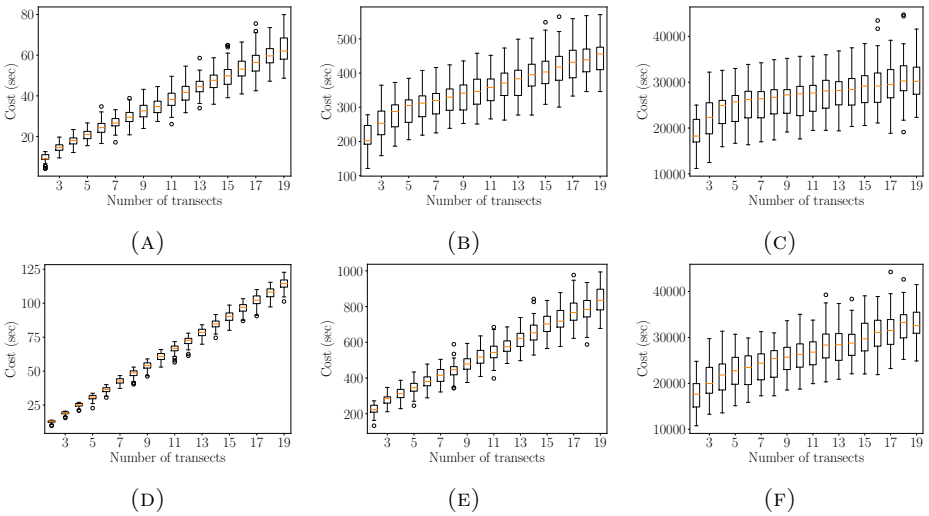


FIGURE 9. Sampling cost as a function of number of transects for PFHS on synthetic GP data considered in Sec. 4.2.1. Stopping error per transect is fixed to 0.05. Top row: Increasing travel time for measurement noise level $\sigma^2 = 0.01$. Bottom row: Increasing travel time for measurement noise level $\sigma^2 = 0.2$.

Similarly, for a fixed lengthscale of $l = 0.3$, the PFHS cost is lower than the median TruVaR cost in 85, 86, and 98 of 100 trials for measurement noise levels of 0.01, 0.1,

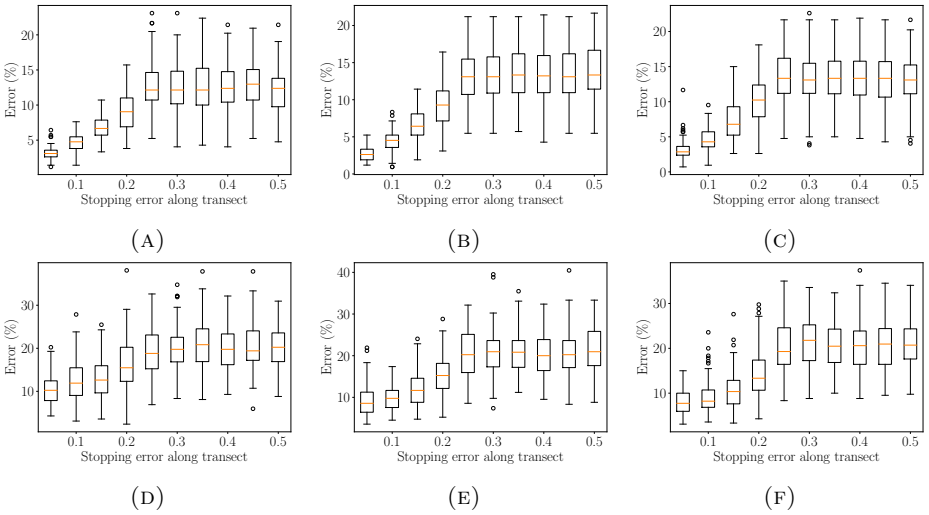


FIGURE 10. Error in estimating super-level set as a function of transect stopping error for PFHS on synthetic GP data considered in Sec. 4.2.1. Number of transects is fixed to 7. Top row: Increasing travel time for measurement noise level $\sigma^2 = 0.01$. Bottom row: Increasing travel time for measurement noise level $\sigma^2 = 0.2$.

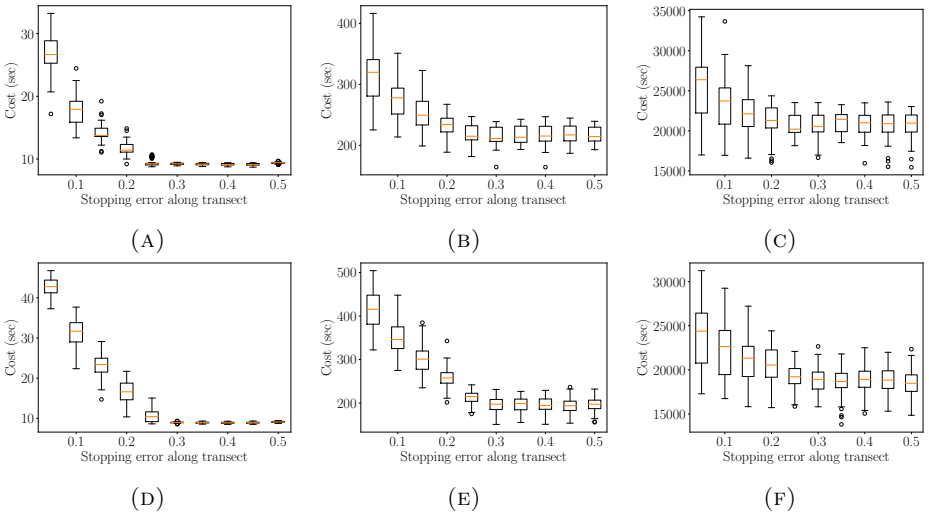


FIGURE 11. Sampling cost as a function of transect stopping error for PFHS on synthetic GP data considered in Sec. 4.2.1. Number of transects is fixed to 7. Top row: Increasing travel time for measurement noise level $\sigma^2 = 0.01$. Bottom row: Increasing travel time for measurement noise level $\sigma^2 = 0.2$.

and 0.2, respectively. Hence, although PFHS has greater variation, it still maintains a significant performance advantage over TruVaR in terms of sampling cost.

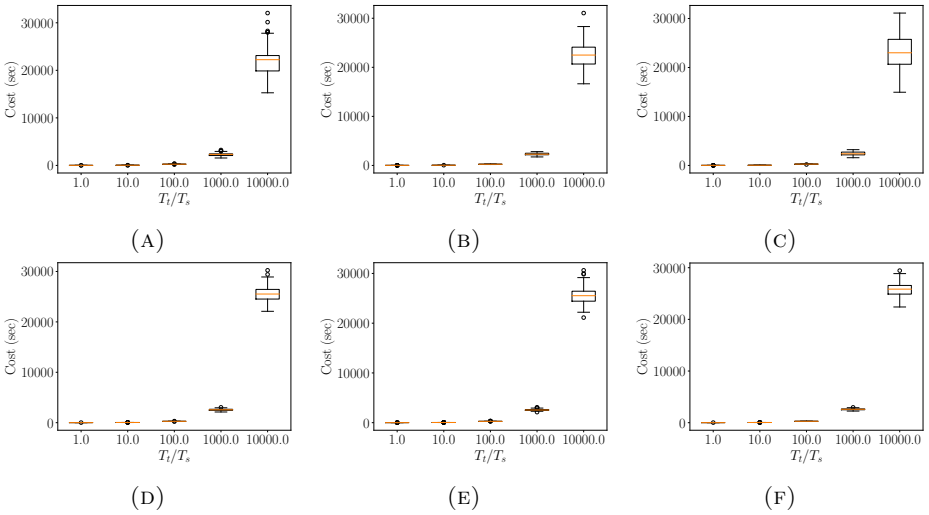


FIGURE 12. Sampling cost as a function of ratio T_t/T_s on synthetic GP test data considered in Sec. 4.2.1. The measurement noise level is fixed to $\sigma^2 = 0.01$. Top row: PFHS for lengthscales $l = 0.3$, 0.6, and 0.9. Bottom row: TruVaR for lengthscales $l = 0.3$, 0.6, and 0.9.

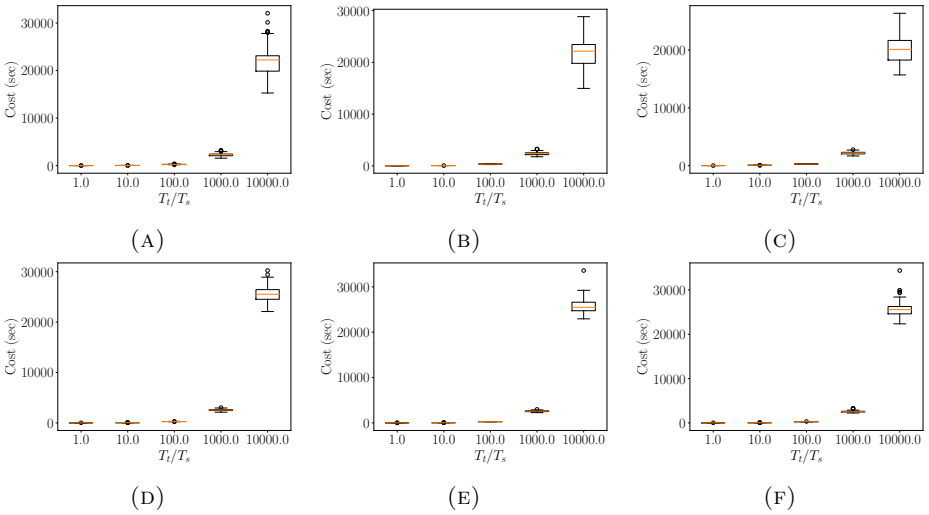


FIGURE 13. Sampling cost as a function of ratio T_t/T_s on synthetic GP test data considered in Sec. 4.2.1. The boundary lengthscale is fixed to $l = 0.3$. Top row: PFHS for measurement noise levels $\sigma^2 = 0.01$, 0.1, and 0.2. Bottom row: TruVaR for measurement noise levels $\sigma^2 = 0.01$, 0.1, and 0.2.

REFERENCES

[1] R. J. Adler and J. E. Taylor, *Random Fields and Geometry*, Springer Science & Business Media, 2009.

[2] D. Azzimonti, D. Ginsbourger, C. Chevalier, J. Bect and Y. Richet, Adaptive design of experiments for conservative estimation of excursion sets, *Technometrics*, **63** (2021), 13-26.
[3] J. Bect, D. Ginsbourger, L. Li, V. Picheny and E. Vazquez, Sequential design of computer experiments for the estimation of a probability of failure, *Statistics and Computing*, **22** (2012), 773-793.
[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control. Vol. 1*, Athena Scientific Belmont, MA, 2005.
[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control. Vol. 2*, Athena Scientific Belmont, MA, 2007.
[6] J. Binney, A. Krause and G. S. Sukhatme, Informative path planning for an autonomous underwater vehicle, in *2010 IEEE International Conference on Robotics and Automation*, 2010, 4791-4796.
[7] I. Bogunovic, J. Scarlett, A. Krause and V. Cevher, Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation, in *Advances in Neural Information Processing Systems*, 2016, 1507-1515.
[8] L. Bottarelli, J. Blum, M. Bicego and A. Farinelli, Path efficient level set estimation for mobile sensors, in *Proceedings of the Symposium on Applied Computing, ACM*, 2017, 262-267.
[9] M. V. Burnashev and K. S. Zigangirov, An interval estimation problem for controlled observations, *Problems in Information Transmission*, **10** (1974), 223-231. Translated from *Problemy Peredachi Informatsii*, **10** (1974), 51-61, https://www.mathnet.ru/links/a55baaa948eda7a4d5f49bee3696975c/ppi1042.pdf
[10] C. J. Cannell and D. J. Stilwell, A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles, in *Proceedings of OCEANS 2005 MTS/IEEE, IEEE*, 2005, 1514-1521.
[11] R. Castro and R. Nowak, Active learning and sampling, in *Foundations and Applications of Sensor Management*, Springer, New York, NY, first ed., 2008, 177-200.
[12] R. M. Castro and R. D. Nowak, Minimax bounds for active learning, *IEEE Trans. Inf. Theory*, **54** (2008), 2339-2353.
[13] N. A. Cruz and A. C. Matos, Adaptive sampling of thermoclines with autonomous underwater vehicles, in *OCEANS 2010 MTS/IEEE SEATTLE, IEEE*, 2010, 1-6.
[14] S. Dasgupta, Analysis of a greedy active learning strategy, *Proc. Advances in Neural Information Processing Systems*, 2005.
[15] P. J. Diggle, J. A. Tawn and R. A. Moyeed, Model-based geostatistics, *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **47** (1998), 299-350.
[16] P. Donmez and J. G. Carbonell, Proactive learning: Cost-sensitive active learning with multiple imperfect oracles, in *Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM*, 2008, 619-628.
[17] C. Duhamel, C. Helbert, M. Munoz Zuniga, C. Prieur and D. Sinoquet, A sur version of the bichon criterion for excursion set estimation, *Statistics and Computing*, **33** (2023), Paper No. 41, 17 pp.
[18] F. Fleuret and D. Geman, Graded learning for object detection, in *Proceedings of the Workshop on Statistical and Computational Theories of Vision of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR/SCTV)*, vol. 2, Citeseer, 1999.
[19] D. Golovin and A. Krause, Adaptive submodularity: A new approach to active learning and stochastic optimization, in *COLT*, 2010, 333-345.
[20] D. Golovin and A. Krause, Adaptive submodularity: Theory and applications in active learning and stochastic optimization, *Journal of Artificial Intelligence Research*, **42** (2011), 427-486.
[21] A. Gotovos, N. Casati, G. Hitz and A. Krause, Active learning for level set estimation, in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.
[22] GPy, GPy: A gaussian process framework in python. https://sheffieldml.github.io/GPy/, since 2012.
[23] A. Guillory and J. Blimes, Average-case active learning with costs, in *Proc. Algorithmic Learning Theory*, 2009, 141-155.
[24] G. Hitz, A. Gotovos, F. Pomerleau, M.-É. Garneau, C. Pradalier, A. Krause and R. Y. Siegwart, Fully autonomous focused exploration for robotic environmental monitoring, in *2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE*, 2014, 2658-2664.

[25] T. N. Hoang, B. K. H. Low, P. Jaillet and M. Kankanhalli, Nonmyopic ε-bayes-optimal active learning of gaussian processes, in *International Conference on Machine Learning*, 2014, 739-747.
[26] M. Horstein, Sequential decoding using noiseless feedback, *IEEE Trans. Inf. Theory*, **9** (1963), 136-143.
[27] Y. Inatsu, M. Karasuyama, K. Inoue and I. Takeuchi, Active learning for level set estimation under cost-dependent input uncertainty, arXiv preprint, `arXiv:1909.06064`, (2019).
[28] B. Jedynak, P. I. Frazier and R. Sznitman, Twenty questions with noise: Bayes optimal policies for entropy loss, *Journal of Applied Probability*, **49** (2012), 114-136.
[29] Z. Jin and A. L. Bertozzi, Environmental boundary tracking and estimation using multiple autonomous vehicles, in *2007 46th IEEE Conference on Decision and Control, IEEE*, 2007, 4918-4923.
[30] R. M. Karp and R. Kleinberg, Noisy binary search and its applications, in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2007, 881-890.
[31] A. P. Korostelev and A. B. Tsybakov, *Minimax Theory of Image Reconstruction*, vol. 82, Springer Science & Business Media, 2012.
[32] D. LeJeune, G. Dasarathy and R. Baraniuk, Thresholding graph bandits with grapl, in *International Conference on Artificial Intelligence and Statistics, PMLR*, 2020, 2476-2485.
[33] J. Lipor and G. Dasarathy, Quantile search with time-varying search parameter, in *2018 52nd Asilomar Conference on Signals, Systems, and Computers, IEEE*, 2018, 1016-1018.
[34] J. Lipor, B. P. Wong, D. Scavia, B. Kerkez and L. Balzano, Distance-penalized active learning using quantile search, *IEEE Transactions on Signal Processing*, **65** (2017), 5453-5465.
[35] K.-C. Ma, L. Liu, H. K. Heidarsson and G. S. Sukhatme, Data-driven learning and planning for environmental sampling, *Journal of Field Robotics*, **35** (2018), 643-661.
[36] D. Marthaler and A. L. Bertozzi, Tracking environmental level sets with autonomous vehicles, in *Recent Developments in Cooperative Control and Optimization*, Springer, 2004, 317-332.
[37] N. H. Motlagh, P. Kortoçi, X. Su, L. Lovén, H. K. Hoel, S. B. Haugsvær, V. Srivastava, C. F. Gulbrandsen, P. Nurmi and S. Tarkoma, Unmanned aerial vehicles for air pollution monitoring: A survey, *IEEE Internet of Things Journal*, (2023).
[38] R. Nowak, Generalized binary search, in *Proc. Allerton Conference on Communication, Control, and Computing*, 2008.
[39] M. B. Or and A. Hassidim, The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well), in *Proc. IEEE Symposium of Foundations of Computer Science*, 2008.
[40] Oregonlive, Portland's air quality was the worst of major cities in the world friday, due to oregon and washington wildfires, *Sept.*, 2020.
[41] A. Pelc, Searching games with errors – fifty years of coping with liars, *Theoretical Computer Science*, **270** (2002), 71-109.
[42] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, vol. 2, MIT Press Cambridge, MA, 2006.
[43] C. Scott and R. D. Nowak, Minimax-optimal classification with dyadic decision trees, *IEEE Trans. Inf. Theory*, **52** (2006), 1335-1353.
[44] Sensirion, Particulate matter sensor SPS30. `https://sensirion.com/`, 2021.
[45] B. Settles, *Active Learning*, Morgan & Claypool, 2012.
[46] A. Singh, R. Nowak and P. Ramanathan, Active learning for adaptive mobile sensing networks, in *Proc. Information Processing in Sensor Networks*, 2006, 60-68.
[47] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer Science & Business Media, 2012.
[48] M. Sullivan, J. Gebbie and J. Lipor, Adaptive sampling for seabed identification from ambient acoustic noise, in *2023 IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE*, 2023.
[49] S. Tschiatschek, A. Singla and A. Krause, Selecting sequences of items via submodular maximization, in *AAAI*, **31** (2017), 2667-2673.
[50] T. Tsiligkaridis, Asynchronous decentralized algorithms for the noisy 20 questions problem, in *2016 IEEE International Symposium on Information Theory (ISIT), IEEE*, 2016, 2699-2703.
[51] U.S. Census Bureau, Camp fire - 2018 california wildfires, *Oct.*, 2020.
[52] U.S. Environmental Protection Agency, Air quality system data mart, *Oct.*, 2020.
[53] R. Waeber, P. I. Frazier and S. G. Henderson, Bisection search with noisy responses, *SIAM Journal on Control and Optimization*, **51** (2013), 2261-2279.

[54] D. Wang, J. Lipor and G. Dasarathy, Distance-penalized active learning via markov decision processes, in *Proc. IEEE Data Science Workshop*, 2019.
[55] R. Willett, R. Nowak and R. M. Castro, Faster rates in regression via active learning, in *Advances in Neural Information Processing Systems*, 2006, 179-186.
[56] D. W. Wong, L. Yuan and S. A. Perlin, Comparison of spatial interpolation methods for the estimation of air quality data, *Journal of Exposure Science & Environmental Epidemiology*, **14** (2004), 404-415.
[57] B. Zhang and G. S. Sukhatme, Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network, in *Proc. IEEE International Conference on Robotics and Automation*, 2007.
[58] Y. Zhou, D. R. Obenour, D. Scavia, T. H. Johengen and A. M. Michalak, Spatial and temporal trends in lake erie hypoxia, 1987-2007, *Environmental Science & Technology*, **47** (2013), 899-905.

Received April 2024; 1st revision October 2024; 2nd revision November 2024; early access November 2024.