Distance-Penalized Active Learning Using Quantile Search

John Lipor[1], Brandon P. Wong[2], Donald Scavia[3], Branko Kerkez[2], and Laura Balzano[1]
[1]Department of Electrical and Computer Engineering,
[2]Department of Civil and Environmental Engineering, [3]School of Natural Resources and Environment
University of Michigan, Ann Arbor
{lipor,bpwong,scavia,bkerkez,girasole}@umich.edu

***Abstract*—Adaptive sampling theory has shown that, with proper assumptions on the signal class, algorithms exist to reconstruct a signal in $\mathbb{R}^d$ with an optimal number of samples. We generalize this problem to the case of spatial signals, where the sampling cost is a function of both the number of samples taken *and* the distance traveled during estimation. This is motivated by our work studying regions of low oxygen concentration in the Great Lakes. We show that for one-dimensional threshold classifiers, a tradeoff between the number of samples taken and distance traveled can be achieved using a generalization of binary search, which we refer to as *quantile search*. We characterize both the estimation error after a fixed number of samples and the distance traveled in the noiseless case, as well as the estimation error in the case of noisy measurements. We illustrate our results in both simulations and experiments and show that our method outperforms existing algorithms in a large range of sampling scenarios.**

***Index Terms*—Active learning, adaptive sampling, autonomous systems, mobile sensors, path planning.**

I. Introduction

Intelligently sampling signals of interest has been a fundamental topic in the signal processing community for many years, the most recent advances in this area being compressed sensing [1] and active learning [2]. In these and other scenarios, the goal is typically to recover a signal from a given class (e.g., bandlimited signals or the Bayes decision boundary for $0/1$ signals) using as few samples as possible. However, in the modeling of spatial phenomena, such as oxygen concentration in lakes, the sampling cost is a function of both the number of samples required *and* the cost to travel to the sample locations. Therefore, the design of provably efficient algorithms to detect spatial phenomena is an important open problem and is the topic of this paper.

Consider our motivating problem, in which we wish to estimate the boundary of a hypoxic region (i.e., a region of oxygen concentration below 2.0 ppm [3]) in the central basin of Lake Erie using an autonomous watercraft with a speed ranging from 0.5-4 m/s. Fig. 1 shows an interpolated estimate

This work was supported by NSF ECCS 1342121, F031543-071159-Graduate Research Fellowship Program, and the MCubed Program at the University of Michigan. The authors would like to thank Hye Won Chung for her insights into the information-theoretic connections in this work.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the authors. The material includes proofs of the technical results in this document. Contact lipor@umich.edu for further questions about this work.

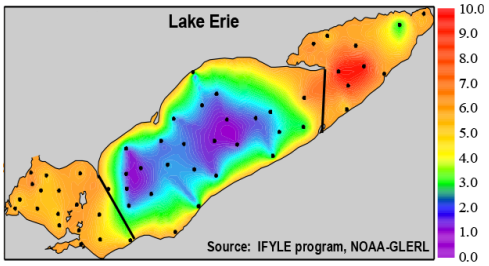


Fig. 1: Dissolved oxygen concentrations in Lake Erie. Points represent sample locations and solid black lines delineate the central basin.

of the oxygen concentration based on a small number of samples taken throughout the lake, where the hypoxic zone is denoted by the dark region (in color: blue/purple region). Oxygen concentration is a strong indicator of the health of the Great Lakes [3] and the spatial extent of such regions is a topic of interest for researchers in the field [4], [5]. We assume the hypoxic region is connected with a smooth boundary and that the boundary remains relatively stationary over the course of a few days. The problem of estimating the boundary can then be viewed as a binary classification problem, in which spatial points receive a label 0 if they are hypoxic and 1 otherwise, and the desired spatial extent corresponds to the Bayes decision boundary. Our goal is to learn the decision boundary in as little *time* as possible.

While the application of optimal active learning algorithms such as [6], [7] minimizes the number of samples required to estimate the boundary, little attention has been given to additional penalties that affect the cost of sampling. In the case of sampling in Lake Erie, the distance traveled between all sampling locations is on the order of hundreds of kilometers, and thus algorithms such as [6], [7], which require a coarse sampling of the entire feature space, are not applicable.

In this paper, we present an active learning algorithm called *quantile search* that achieves a tradeoff between the number of measurements and distance traveled to estimate the change point of a one-dimensional step function. At its two extremes, quantile search minimizes either the number of samples or the distance traveled to estimate the decision boundary, with a tradeoff achieved by varying a search parameter. We derive the expected number of samples required and distance traveled in the noiseless case and bound the number of samples

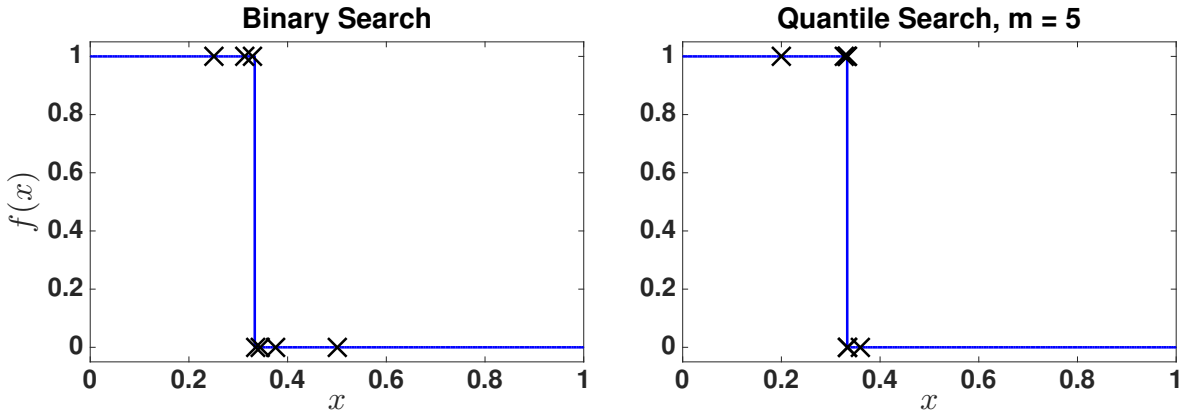


Fig. 2: Example step function with $\theta = 1/3$ with corresponding measurements (marked by an x) taken using binary search (left) and quantile search with $m = 5$ (right).

required in the case of noisy measurements. We also show how a series of one-dimensional estimates can be used to estimate the two-dimensional boundary of interest. This paper is an extension of our previous work [8]. Our contributions beyond [8] are as follows. We provide detailed proofs of our theoretical results in the supplemental material [9]. We present a novel generalization in the case of noisy measurements that, unlike our previous work, is equivalent to the noiseless case when the probability of measurement error is zero. We also provide two algorithmic improvements for the problem of interest and show in simulations that these greatly reduce the required sampling time. Our simulations are more realistic, including real bathymetry data from Lake Erie provided by the National Oceanic and Atmospheric Administration [10]. We also compare the performance of our algorithm to a version of *proactive learning* [11]. Finally, we include results of our experiments performed on Third Sister Lake in Ann Arbor, MI with an autonomous watercraft controlled using a cloud-based architecture.

II. Problem Formulation & Related Work

Determining the spatial extent of the hypoxic region shown in Fig. 1 can be interpreted as learning a two-dimensional Bayes decision boundary. Following [16], we split our two-dimensional problem into several one-dimensional intervals, a process that is described further in Section IV and can be viewed in Fig. 7a-7d. The idea here is that we can carve a two-dimensional boundary fragment (indeed any d-dimensional boundary fragment class) into several one-dimensional interval problems, piecing the solutions together for a full boundary estimate.[1]

Having reduced the problem to several one-dimensional problems, on each interval we must find a threshold beyond which the lake is hypoxic. Define the step function class

$$\mathcal{F} = \{f : [0,1] \to \mathbb{R} | f(x) = \mathbf{1}_{[0,\theta)}(x)\}$$

where $\theta \in [0,1]$ is the change point and $\mathbf{1}_S(x)$ denotes the indicator function, which is 1 on the set S and 0 elsewhere.

[1]As we discuss in Section II-A, this is order-optimal in terms of sample complexity. Our heuristic algorithmic improvements of Section III-C allow us to more intelligently sample from one interval to the next.

An example function belonging to $\mathcal{F}$ with $\theta = 1/3$ is shown in Fig. 2. In contrast to the standard active learning scenario, our goal is to estimate θ while minimizing the total time required for sampling, a function of both the number of samples taken *and* the distance traveled. Denote the observations $\{Y_n\}_{n=1}^N \in \{0,1\}^N$ as samples of an unknown function $f_\theta \in \mathcal{F}$ taken at sample locations on the unit interval $\{X_n\}_{n=1}^N$. With probability p, $0 \le p < 1/2$, we observe an erroneous measurement. Thus

$$Y_n = \begin{cases} f_\theta(X_n) & \text{with probability } 1-p \\ 1 - f_\theta(X_n) & \text{with probability } p \end{cases} = f(X_n) \oplus U_n,$$

where $\oplus$ denotes summation modulo 2, and $U_n \in \{0,1\}$ are Bernoulli random variables with parameter p. While other noise scenarios are common, here we assume the U_n are independent and identically distributed and independent of $\{X_n\}$. This noise scenario is of interest as the motivating data (oxygen concentration) is a thresholded value in $\{0,1\}$, where Gaussian noise results in improper thresholding of the measurements. The extension to nonuniform noise (e.g., a Tsybakov-like noise condition as studied in [7]) remains as a topic for future work.

A. Related Work

A number of active learning algorithms designed to estimate θ exist; however, these algorithms typically assume the sampling cost is due only to the measurements themselves. Most similar to our algorithm is the method of binary bisection and its extensions [7], [12]–[17]. In the noiseless case, binary bisection estimates the change point of a step function on the unit interval by successively halving the space of potential classifiers, termed the *hypothesis space*. An example of this search procedure is shown in the left-hand plot of Fig. 2. A noise-tolerant version of this algorithm was first presented in [12], where measurements are flipped with known probability p. A discretized version of this algorithm was analyzed in [13] and shown to be minimax optimal in [7] under the Tsybakov noise condition. Further, the authors of [7] use the discretized algorithm to show that a series of one-dimensional threshold estimates can be used to estimate functions belonging to the boundary fragment class in d dimensions at a minimax optimal

rate. The original algorithm presented in [12] was recently shown to converge at a geometric rate in [17]. Binary bisection has also been used to obtain optimal rates in optimization [18] and in the noisy 20 questions problem [19]. In [16], the authors give a spatial sampling problem as motivation for the probabilistic binary search (PBS) algorithm. However, a simple analysis shows that in the noiseless case, to estimate the threshold of a step function on the unit interval, binary search travels the entire unit interval. Hence, while the worst-case number of samples required is minimized, the total distance traveled is the worst possible. In the motivating problem given above, the central basin of Lake Erie has a width of roughly 80 km, making this approach prohibitive.

More sophisticated active learning algorithms have been widely studied, achieving optimal rates for piecewise constant functions in [7] and for the linear support vector machine in [20]. In both cases, the algorithm begins by uniformly sampling the entire feature space. Again considering the problem of interest, the central basin of Lake Erie has an area of approximately 14,000 km^2, making this approach infeasible. In contrast, the algorithm studied in [7] was used in [6] to measure the hydrodynamics of Lake Wingra in Madison, WI, which has an area of 1.3 km^2.

Nonuniform sampling costs are studied in [11], [21]–[23]. In [21], the authors use the uncertainty sampling heuristic to determine the most informative points and penalize for spatial costs using the traveling salesman problem with profits. The work of [22] uses both uncertainty and diversity to select points and also penalizes for arbitrary costs. In both cases, the algorithm proceeds in batches, i.e., by iteratively requesting a set of labels and retraining the classifier. This approach suffers the same pitfalls as [6] in that the algorithm can require traversing the entire feature space multiple times. Further, neither algorithm is accompanied by theoretical guarantees. In [23], the authors present and analyze a greedy algorithm for active learning with nonuniform costs. However, in our case the cost associated with each point is the distance from the previous point, so the costs in question are both nonuniform and *dynamic*. A somewhat similar algorithm, known as *proactive learning*, is presented in [11], where the proposed strategy chooses at each round the point maximizing the difference between or ratio of informativeness and cost to label the point. In Section IV, we compare with this algorithm using mutual information as our metric for informativeness.

The problem of sampling spatial phenomena using mobile robots has been studied in signal processing and robotics literature as well. In [24], [25], the authors study the case where the sampling cost is near-zero and show that equispaced parallel lines result in the minimum distance required to reconstruct a variety of practical signals. Mutual information is also used as a metric for informativeness in [26], where the authors impose a Gaussian process model to perform path planning for robots used to track a variety of spatial phenomena. A greedy algorithm is presented with theoretical guarantees based on submodularity [27]. However, the model imposed is not appropriate for determining the boundary of a region of interest. The recent work of [28] considers the measurement cost and travel time to estimate the location of point targets using mobile robots but does not easily extend to the case of estimating the boundary of a region of interest. The algorithm in [29] is similar to the one described in [11], with the main difference being that in early stages the algorithm emphasizes regularity of samples (i.e., encourages early samples to be taken uniformly throughout the feature space).

III. Quantile Search

In this section, we present our algorithm *quantile search*, an extension of binary search and ideas in [13], [16] to penalize both the sample complexity and distance traveled during the estimation procedure. The basic idea behind this algorithm is as follows. We wish to find a tradeoff between the number of samples required and the total distance traveled to achieve a given estimation error for the change point of a step function on the unit interval. As we know, binary bisection minimizes the number of required samples. On the other hand, continuous spatial sampling minimizes the required distance to estimate the threshold. Binary search bisects the feasible interval (hypothesis space) at each step. In contrast, one can think of continuous sampling as dividing the feasible interval into infinitesimal subintervals at each step. With this in mind, a tradeoff becomes clear: one can divide the feasible interval into subintervals of size $1/m$, where m is a real number between 2 and ∞. Intuition would tell us that increasing m would increase the number of samples required but decrease the distance traveled in sampling. In what follows, we show that this intuition is correct in both the noise-free and noisy cases, resulting in two novel search algorithms.

A. Deterministic Quantile Search

We first describe and analyze quantile search in the noise-free case ($p = 0$), here referred to as deterministic quantile search (DQS). To estimate the change point of a step function, deterministic binary bisection travels either forward or backward (depending on the measurement) a fraction 1/2 into the feasible interval. In contrast, the DQS algorithm presented here travels $1/m$ forward or backward, where $m \in [2, \infty)$. While the DQS measurements for $m > 2$ are less informative than in binary bisection, we expect that the distance traveled during the estimation procedure will be reduced, since we can pass the change point by a fraction at most $1/m$. The search procedure for the case of $m = 5$ is shown in the right-hand plot of Fig. 2. Note that in contrast to binary search, quantile search does not overshoot the change point $\theta = 1/3$ by a significant amount. A formal description of the procedure is given in Algorithm 1. In the following subsections, we analyze the expected sample complexity and distance traveled for the algorithm and show the required number of samples increases monotonically with m, and the distance traveled decreases monotonically with m, indicating that the desired tradeoff is achieved.

1) Convergence of Estimation Error: We analyze the expected error after a fixed number of samples for the DQS algorithm. The main result and a sketch of the proof are provided here. An expanded proof can be found in the supplemental material [9].

Algorithm 1 Deterministic Quantile Search (DQS)

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

Theorem 1. *Consider a deterministic quantile search with parameter m and let $\rho = \frac{m-1}{m}$. Begin with a uniform prior on θ. The expected estimation error after n measurements is then*

$$\mathbb{E}[|\hat{\theta}_n - \theta|] = \frac{1}{4}\left[\rho^2 + (1-\rho)^2\right]^n. \tag{1}$$

Proof. (Sketch; see complete proof in [9]) The proof proceeds from the law of total expectation. Let $Z_n = |\hat{\theta}_n - \theta|$. The first measurement is taken at $1/m$, and hence the expected error can be calculated when $\theta \leq 1/m$ and $\theta > 1/m$.

$$\begin{aligned}\mathbb{E}[Z_1] &= \mathbb{E}\left[Z_1 | \theta \leq \frac{1}{m}\right]\mathbb{P}\left(\theta \leq \frac{1}{m}\right) + \\ &\quad \mathbb{E}\left[Z_1 | \theta > \frac{1}{m}\right]\mathbb{P}\left(\theta > \frac{1}{m}\right) \\ &= \frac{1}{4}\left[(1-\rho)^2 + \rho^2\right].\end{aligned}$$

Similarly, after the second measurement is taken, there are four intervals, two which partition the interval $[0, 1/m]$, and two which partition $(1/m, 1]$. These result in four monomials of degree 4, one of which is $(1-\rho)^4$, one which is ρ^4, and two which are $(1-\rho)^2\rho^2$. The basic idea is that each "parent" interval integrates to $(1-\rho)^i\rho^j$ and in the next step gives birth to two "child" intervals, one evaluating to $(1-\rho)^{i+1}\rho^j$ and the other $(1-\rho)^i\rho^{j+1}$. The proof of the theorem then follows by induction. □

Consider the above result when $m = 2$. In this case, the error becomes $\mathbb{E}[|\hat{\theta}_n - \theta|] = 2^{-(n+2)}$. Comparing to the worst case, we see that the average case sample complexity is exactly one sample better than the worst case, matching the well-known theory of binary search. In Section IV we confirm this result through simulation.

2) Distance Traveled: Next, we analyze the expected distance traveled by the DQS algorithm in order to converge to the true θ. The proof is similar to that of the previous theorem in that it follows by the law of total expectation. After each sample, we analyze the expected distance given that the true θ lies in a given interval. The result and a proof sketch are given below, with the full proof included in the supplemental material [9].

Theorem 2. *Let D denote a random variable representing the distance traveled during a deterministic quantile search with parameter m. Begin with a uniform prior on θ. Then*

$$\mathbb{E}[D] = \frac{m}{2m-2}. \tag{2}$$

Proof. (Sketch, see full proof in [9]) We first consider the expected distance traveled before the algorithm reaches a point $x_1 > \theta$. Let D_1 be a random variable denoting this distance. Once the algorithm passes this point, it moves in the reverse direction until reaching $x_2 < \theta$, moving a distance D_2. This process repeats until convergence. Let D_n be a random variable denoting the distance required to move to the right of θ for the $\lceil \frac{n}{2} \rceil$th time when n is odd, and to the left of θ for the $\frac{n}{2}$th time when n is even. In this case, we have that

$$\mathbb{E}[D] = \sum_{n=1}^{\infty}\mathbb{E}[D_n]. \tag{3}$$

First, we would like to find $\mathbb{E}[D_1]$. Let A_i denote the interval $\left[\frac{1}{m}\sum_{p=0}^{i-1}\left(\frac{m-1}{m}\right)^p, \frac{1}{m}\sum_{p=0}^{i}\left(\frac{m-1}{m}\right)^p\right)$, where $A_0 = \left[0, \frac{1}{m}\right)$, so that the A_i's form a partition of the unit interval whose endpoints are possible values of the sample locations X_j. Now note that

$$\mathbb{E}[D_1] = \sum_{i=0}^{\infty}\mathbb{E}[D_1 | \theta \in A_i]\mathbb{P}(\theta \in A_i).$$

Then since we assume θ is distributed uniformly over the unit interval,

$$\begin{aligned}\mathbb{P}(\theta \in A_i) &= \frac{1}{m}\sum_{p=0}^{i}\left(\frac{m-1}{m}\right)^p - \frac{1}{m}\sum_{p=0}^{i-1}\left(\frac{m-1}{m}\right)^p \\ &= \frac{1}{m}\left(\frac{m-1}{m}\right)^i.\end{aligned}$$

Next, note that

$$\begin{aligned}\mathbb{E}[D_1 | \theta \in A_i] &= \frac{1}{m}\sum_{p=0}^{i}\left(\frac{m-1}{m}\right)^p \\ &= 1 - \left(\frac{m-1}{m}\right)^{i+1}.\end{aligned}$$

Thus we have

$$\begin{aligned}\mathbb{E}[D_1] &= \sum_{i=0}^{\infty}\mathbb{E}[D_1 | \theta \in A_i]\mathbb{P}(\theta \in A_i) \\ &= \sum_{i=0}^{\infty}\left[1 - \left(\frac{m-1}{m}\right)^{i+1}\right]\left[\frac{1}{m}\left(\frac{m-1}{m}\right)^i\right] \\ &= \frac{m}{2m-1}.\end{aligned}$$

The proof proceeds by rewriting the above in terms of $\rho = (m-1)/m$ and then calculating $\mathbb{E}[D_n]$. This is done by dividing each A_i into subintervals which form partitions of A_i. By induction we get

$$\mathbb{E}[D_n] = \frac{m}{(2m-1)^n}, \tag{4}$$

and the result then follows from the infinite sum of (3). □

3) Sampling Time: Using the above results, we wish to find the optimal tradeoff for a given set of sampling parameters. Let γ be the time required to take one sample and η be the time required to travel one unit of distance. The total sampling time T is then

$$T = \gamma N + \eta D, \tag{5}$$

where N denotes the number of samples required. Given a fixed sampling time and desired error, (5) can be used to estimate the sample budget N. However, this approach differs from our goal of minimizing the total sampling time. Alternatively, the average value of N can be estimated numerically and used to optimize the expected value of T. We show examples of this approach in Section IV for both the deterministic and probabilistic versions of quantile search.

As a final note, one may wonder about the relation to what is known as m-ary search [30]. In contrast to quantile search, m-ary search is tree-based. To make the difference clear, consider an example with $\theta = 3/8$ and let $m = 4$. In this case, both algorithms take their first sample at $X = 1/4$. However, after measuring $Y = 1$, quantile search takes its second measurement at $X = 7/16$, while m-ary search proceeds to $X = 1/2$. One may then expect that both algorithms would achieve the desired tradeoff, with m-ary search using fewer samples and more distance for the same value of m. We focus on quantile search for two reasons. First, quantile search does not require m to be an integer and therefore gives more flexibility in the resulting tradeoff. Second, quantile search as described is the natural generalization of PBS and lends itself to the analysis of [13], [16] in the case where the measurements are noisy. A comparison to noisy m-ary search is a topic for future work.

B. Probabilistic Quantile Search

In this section, we extend the idea behind Section III-A to the case where measurements may be noisy (i.e., $p \geq 0$). In [13], the authors present an algorithm referred to in the literature as *probabilistic binary search* (PBS). The basic idea behind this algorithm is to perform Bayesian updating in order to maintain a posterior distribution on θ given the measurements and locations. Rather than bisecting the interval at each step, the algorithm bisects the posterior distribution. This process is then iterated until convergence and has been shown to achieve optimal sample complexity throughout the literature [7], [15]. We now extend this idea using the quantile methodology of the previous section, resulting in what we term *probabilistic quantile search* (PQS).

The idea behind PQS is straightforward. Starting with a uniform prior, the first sample is taken at $X_1 = 1/m$. The posterior density $\pi_n(x)$ is then updated as described below, and $\hat{\theta}_n$ is chosen as the median of this distribution. The algorithm proceeds by taking samples X_n such that

$$\int_0^{X_{n+1}} \pi_n(x)dx = \frac{1}{m}.$$

For $m = 2$, the above denotes the median of the posterior distribution and reduces to PBS, while in general this denotes sampling at the m-quantile of the posterior. A formal description is given in Algorithm 2.

Algorithm 2 Probabilistic Quantile Search (PQS)

1: **Input:** search parameter m, probability of error p
2: **Initialize:** $\pi_0(x) = 1$ for all $x \in [0,1]$, $n \leftarrow 0$
3: **while** not converged **do**
4: choose X_{n+1} such that $\int_0^{X_{n+1}} \pi_n(x)dx = \frac{1}{m}$
5: $Y_{n+1} \leftarrow f(X_{n+1}) \oplus U_{n+1}$, where $U_{n+1} \sim$ Ber(p)
6: **if** $Y_{n+1} = 0$ **then**
7:

$$\pi_{n+1}(x) = \begin{cases} (1-p)\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x \leq X_{n+1} \\ p\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x > X_{n+1} \end{cases}$$

8: **else**
9:

$$\pi_{n+1}(x) = \begin{cases} p\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x \leq X_{n+1} \\ (1-p)\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x > X_{n+1} \end{cases}$$

10: **end if**
11: $n \leftarrow n+1$
12: **end while**
13: estimate $\hat{\theta}_n$ such that $\int_0^{\hat{\theta}_n} \pi_n(x) = 1/2$

We derived the update for PQS in our previous work [8], and it can be seen in steps 7 and 9 of Algorithm 2. Here we derive a more general version of the update that will be referred to in Section IV-B. Begin with the first sample. We have $\pi_0(x) = 1$ for all x and wish to find $\pi_1(x)$. Let $f_1(x|X_1, Y_1)$ be the conditional density of θ given X_1, Y_1. Applying Bayes rule, the posterior becomes:

$$f_1(x|X_1, Y_1) = \frac{\mathbb{P}(X_1, Y_1|\theta = x)\pi_0(x)}{\mathbb{P}(X_1, Y_1)}$$

For illustration, consider the case where $\theta = 0$. We now take the first measurement at $X_1 = \phi$ (note $\phi = 1/m$ for PQS). Then

$$\mathbb{P}\left(X_1 = \phi, Y_1 = 0|\theta = 0\right) = 1 - p$$

and

$$\mathbb{P}\left(X_1 = \phi, Y_1 = 1|\theta = 0\right) = p.$$

In fact, this holds for any $\theta < \phi$. Now examine the denominator:

$$\begin{aligned} \mathbb{P}(X_1 = \phi, Y_1 = 0) &= \phi(1-p) + (1-\phi)p \\ &:= \phi * p, \end{aligned}$$

We then update the posterior distribution to be

$$\pi_1(x) = \begin{cases} \frac{(1-p)}{\phi * p} & x \leq \phi \\ \frac{p}{\phi * p} & x > \phi. \end{cases}$$

The equivalent posterior density can be found for when $Y_1 = 1$. The process of making an observation and updating the prior is then repeated, yielding general formula for the posterior update. When $Y_{n+1} = 0$, we have

$$\pi_{n+1}(x) = \begin{cases} \frac{(1-p)}{\phi * p}\pi_n(x) & x \leq X_{n+1} \\ \frac{p}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

Similarly, for $Y_{n+1} = 1$, we have

$$\pi_{n+1}(x) = \begin{cases} \frac{p}{\phi * p}\pi_n(x) & x \leq X_{n+1} \\ \frac{(1-p)}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

1) Convergence of Estimation Error: Analysis of the above algorithm has proven difficult since its inception in 1974, with a first proof of a geometric rate of convergence appearing only recently in [17]. Instead, the authors and those following use a discretized version involving minor modifications. We follow this strategy, with the discretized algorithm given in the supplemental material [9]. In this case, the unit interval is divided into bins of size Δ, such that $\Delta^{-1} \in \mathbb{N}$. The posterior distribution is parameterized, and a parameter α is used instead of p in the Bayesian update, where $0 < p < \alpha$. The analysis of rate of convergence then centers around the increasing probability that at least half of the mass of $\pi_n(x)$ lies in the correct bin. A formal description of the algorithm can be found in the supplemental material [9]. Given this discretized version of PQS, we arrive at the following result.

Theorem 3. *Under the assumptions given in Section II, the discretized PQS algorithm satisfies*

$$\sup_{\theta \in [0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq 2\left(\frac{m-1}{m} + \frac{2\sqrt{p(1-p)}}{m}\right)^{n/2}. \quad (6)$$

The proof can be found in the supplemental material [9]. In the case where $m = 2$, the above result matches that of [13], [16] as desired. One important fact to note is that in contrast to the deterministic case, the result here is an upper bound on the number of samples required for convergence as opposed to an expected value. As this seems to be the case for all analyses of similar algorithms [13], [16], [17], we instead rely on Monte Carlo simulations to choose the optimal value of m. Finally, the bound here is loose. For clarity, consider the case where $p = 0$ and $m = 2$. Then the above becomes

$$\sup_{\theta \in [0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq 2\left(\frac{1}{2}\right)^{n/2}.$$

As noted in [7], we can see by inspection that

$$\sup_{\theta \in [0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq \left(\frac{1}{2}\right)^{n+1},$$

indicating that we lose a factor of about $n/2$, even for the PBS algorithm bound in [7]. However, in [7], the authors use this result when $m = 2$ to show rate optimality of the PBS algorithm. This fact suggests that despite the discrepancy, the result of Thm 3 may still be useful in proving some sort of optimality for the PQS algorithm.

While the rate of convergence for PQS can be derived using standard techniques, the expected distance or a useful bound on the distance is more difficult. The technique used in Section III-A becomes intractable as the values of X_n are no longer deterministic. The approach of examining the posterior distribution after each step and calculating the possible locations has been examined, but at the nth measurement, there are 2^{n-1} possible distributions. Further, PQS as described above has the undesirable property that it does not always travel toward the median of the distribution—a problem we overcome in the next section—and hence the distance traveled is higher than strictly necessary, making analysis of its distance properties of minimal practical importance.

2) Truncated PQS: Probabilistic quantile search as presented in Algorithm 2 is not a strict generalization of DQS in the sense that the two algorithms are not equivalent in the noiseless case. Moreover, in some cases, PQS will choose a sample location farther away from the current location than the median. This choice is suboptimal, as the median of the posterior is the most informative point (in an information-theoretic sense), and hence traveling farther to obtain less information is contrary to our overall goal. For these reasons, we propose the following variant of PQS, which has a sample complexity and distance traveled no worse than the PQS algorithm in Algorithm 2. The algorithm satisfies the statement of Thm. 3 (see [9]), and we show the improved performance in terms of both distance and sample complexity in Section IV. Instead of taking a sample at the m-quantile of the posterior, we instead truncate the posterior distribution in such a way to maintain the median as well as guarantee that the m-quantile of this truncated posterior is moving our sampling location towards the median of the posterior (the most informative point). We refer to this algorithm as Truncated PQS (TPQS).

Truncated PQS begins by sampling at the m-quantile as in PQS. For subsequent samples, we first define

$$\chi = \min\left\{\int_0^{X_n} \pi_n(x)dx, \int_{X_n}^1 \pi_n(x)dx\right\},$$

the probability in the tail of the distribution that would possibly cause us to move away from the median point. We then define the truncated distribution to be the normalized form of

$$\tilde{\pi}_n(x) = \begin{cases} 0, & \int_0^x \pi_n(z)dz \leq \chi \\ 0, & \int_x^1 \pi_n(z)dz \leq \chi \\ \pi_n(x), & \text{otherwise} \end{cases}.$$

Finally, the sample location is chosen as

$$X_{n+1} = \underset{X \in \{\tilde{X}_0, \tilde{X}_1\}}{\arg\min} |X_n - X|,$$

where

$$\int_0^{\tilde{X}_0} \tilde{\pi}(x)dx = \frac{1}{m} \quad \text{and} \quad \int_{\tilde{X}_1}^1 \tilde{\pi}(x)dx = \frac{m-1}{m}.$$

Analogous to traveling "forward" or "backward" in DQS, this process guarantees that we always choose sample locations that are in the direction of the median of the posterior. This fact ensures that the information gain is at least that of the PQS algorithm, while choosing the nearer of the two locations results in a distance no greater than that of PQS. Note that we continue to use $\pi_n(x)$ as the posterior distribution of θ and update this distribution according to Algorithm 2, i.e., we only use $\tilde{\pi}_n(x)$ when choosing the sample locations. Further, for the case of $m = 2$, this generalization and PQS are equivalent, both resulting in the PBS algorithm.

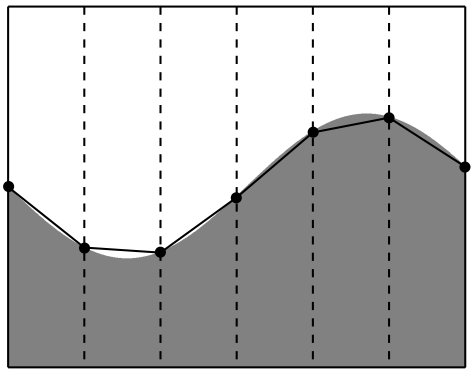


Fig. 3: Example of set belonging to boundary fragment class and piecewise linear estimation of boundary.

3) Stopping Criterion: Previous work on PBS centers around the case where there is a fixed sample budget, avoiding the need for a stopping criterion for this algorithm. However in our application, while we need to further reduce sampling resources, we only stop sampling once we have reached a desired accuracy. In this case, one natural choice of stopping criteria for PBS would be to stop when the distance between successive samples is smaller than some predetermined value. However, in the case of PQS with high m, the step size may be very small from the start, resulting in early termination. In the case of DQS, the width of the feasible interval provides a direct measure of the absolute error in estimating θ. While there is no such width in the case of PQS, the certainty in our estimate of θ is quantified via the posterior distribution $\pi_n(x)$, which is discretized in our implementation. In light of this, we terminate PQS (or its generalized version) when there exists an x_i such that $\pi_n(x_i) \geq 0.9$.

C. Algorithmic Improvements

In this section, we describe two heuristics that can be used to further reduce the sampling time. These heuristics are appropriate in the case where the decision boundary is smooth in some sense and is estimated using a series of successive quantile searches. Consider the boundary fragment class on $[0,1]^d$ defined informally in [7] as the collection of sets in which the Bayes decision boundary is a Hölder smooth function of the first $d-1$ coordinates. In $[0,1]^2$, this implies that the boundary is crossed at most one time when traveling on a path along the second coordinate. The boundary can be estimated by dividing the problem into strips along the first dimension, estimating the change point of each strip, and estimating the boundary as a piecewise linear function of the estimates, as shown in Fig. 3. For simplicity, we motivate the heuristics in this section by restricting f to the class of Lipschitz functions (a subset of Hölder smooth functions). Recall that a function $f : [0,1]^d \to \mathbb{R}$ is said to be Lipschitz with constant $L \geq 0$ if for all $x_1 \neq x_2$

$$|f(x_1) - f(x_2)| \leq L \|x_1 - x_2\|.$$

Returning to Fig. 3, we see that a great deal of time would be wasted by returning to the origin after estimating the boundary at each strip. In this section, we leverage the assumed smoothness to intelligently initialize quantile search, resulting in significantly reduced sampling times, as shown in the simulations.

1) Initialization Using Previous Estimate: Assume we split the region of interest into K strips, each of which is a step function on the unit interval whose change point we wish to estimate. Let the true change point of the kth strip be θ^k and the estimate be $\hat{\theta}^k$. The smoothness assumption implies that θ^{k+1} is not "too far" from θ^k. For example, if f is Lipschitz with constant L and two successive strips are located at x_k and x_{k+1}, we know that $|\theta^k - \theta^{k+1}|/|x_k - x_{k+1}| \leq L$. For this reason, our first proposed improvement is to let the first sample location of the $k+1$st strip X_0 be the previous estimate $\hat{\theta}^k$. Note that if we further assume a uniform prior on the subinterval allowed by the smoothness assumption, we are choosing our first sample as the minimum absolute error estimate, i.e., the median of the distribution. For later reference, we refer to this initialization as Improvement 1 (I-1). We show in Section IV that this simple heuristic dramatically reduces the required sampling time of our algorithm.

2) Nonuniform Priors: Our second proposed algorithmic improvement involves assigning a nonuniform prior when beginning the search. Similar to the previous improvement, we utilize the function smoothness to assign lower starting probabilities to points unlikely to lie near the decision boundary. Letting $\hat{\theta}^k$ again be the boundary estimate at the kth strip, we assign a nonuniform prior whose mean is centered around $\hat{\theta}^k$. We propose the use of either a piecewise uniform or a Gaussian kernel function and refer to these as I-2.1 and I-2.2, respectively. Let the strip width $|x_k - x_{k+1}| = W$. We assign the prior probability for the $k+1$st strip to be either

$$\pi_0(x) = \begin{cases} c_1, & \left|x - \hat{\theta}^k\right| \leq LW \\ c_2, & \left|x - \hat{\theta}^k\right| > LW \end{cases} \qquad (\mathrm{I}-2.1),$$

where $c_1 > c_2$, or

$$\pi_0(x) = c_3 \exp\left(-\frac{(x-\hat{\theta}^k)^2}{2(LW)^2}\right) \qquad (\mathrm{I}-2.2),$$

where c_3 is a normalization constant so that the prior sums to 1. We discuss the choice of L and W in Section IV.

IV. Simulations & Experiments

In this section, we show the efficacy of our algorithm through simulations. We first verify the theoretical guarantees provided in Section III and then compare the performance of PQS with the generalized version, which we refer to as TPQS. Next, we compare our method to proactive learning from [11]. We then show how a series of one-dimensional searches can be used to estimate the boundary of a two-dimensional hypoxic region in Lake Erie. We conclude with experimental results from Third Sister Lake in Ann Arbor, MI.

A. Verification of Algorithms

In this section, we verify through simulation the theoretical rate of convergence and distance traveled derived in Section III-A. Further, we present simulated results for the

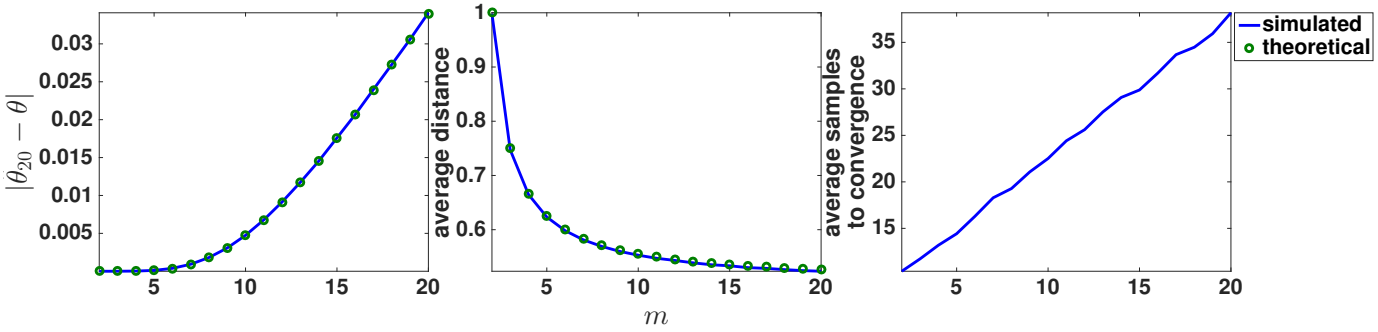


Fig. 4: Simulated and theoretical values for DQS. Left-to-right: expected error after 20 samples, distance traveled before convergence to an estimation error less than 1×10^{-4}, simulated average samples required to converge to the same error.

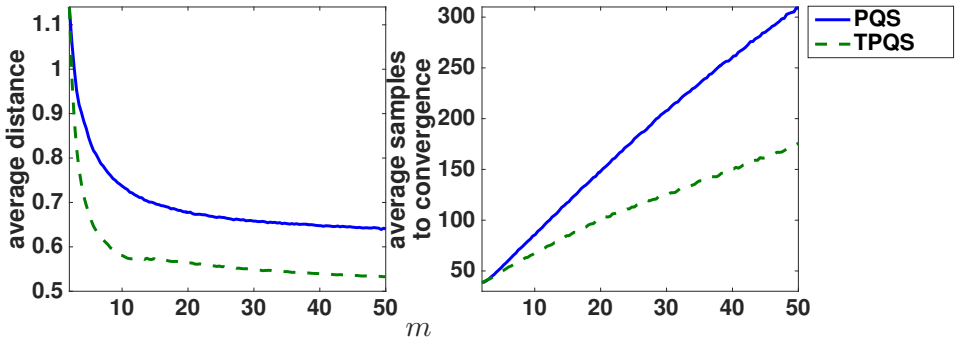


Fig. 5: Average simulated values for PQS and TPQS. Left-to-right: distance traveled during estimation and number of samples required to converge.

PQS and TPQS algorithms and show that the desired tradeoff is achieved by both algorithms, with TPQS achieving better overall performance.

We first simulate the the DQS algorithm over a range of m from 2 to 20, where θ is swept over a 1000-point grid on the unit interval. The resulting average error after 20 samples is shown in the left-most plot of Fig. 4, while the average distance before convergence to an error of $\varepsilon = 1 \times 10^{-4}$ is shown in the middle plot of the same figure. The figures show the theoretical values for expected error and distance match the simulated values. The right-most plot of Fig. 4 shows the number of samples required to converge to the same error. From the figures, our intuition is confirmed; the number of samples required is monotonically increasing in m, while the distance traveled is monotonically decreasing. This indicates that DQS achieves the desired tradeoff in the noise-free case.

Next, we simulate the PQS and TPQS algorithms with error probability $p = 0.1$ over a range of m from 2 to 50, where θ ranges over a 100-point grid on the unit interval with 100 random instances run for each value of θ. The left-hand plot of Fig. 5 shows the average number of samples required to converge to a mass of at least 0.9 at a single point, as described in Section III-B3. As in the deterministic case, the required number of samples increases monotonically with m. The right-hand plot of Fig. 5 shows the average distance traveled before converging to the same error value. Again, the distance decreases monotonically with m, indicating that the algorithm achieves the desired tradeoff in the noisy case. Further, we see that TPQS outperforms PQS both in terms of samples required and distance traveled. Because of this, we consider only TPQS in all remaining simulations.

B. Application of Proactive Learning

The most competitive algorithm to quantile search is that of [11] applied to our problem. Of the scenarios explored in [11], the most relevant is Scenario 3, in which a non-uniform cost is charged for each label. The authors propose choosing each sample location to maximize the utility $U(X)$ at each round, where utility is defined as the difference between the value of the sample at X and the cost of taking that sample. The authors alternatively define utility as the ratio of value to cost, lending to a more natural interpretation that is similar to [29]. However, we found this version to result in poor performance, and hence we rely on the first approach. For comparison purposes, we maintain an estimate of the posterior distribution of θ as in quantile search. We define the value of a point X as the mutual information $I(\theta; X, Y)$ [31]. Note that in the noiseless case, Y is a deterministic function of θ, and hence mutual information is a misnomer. In this case, we still consider the reduction in entropy of θ given the measurement Y taken at point X. The relation of binary search to communicating a noisy sequence of bits over a binary symmetric channel has been well-studied [12]. In the noiseless case, we have

$$H(\theta) - H(\theta|X, Y) = H_b(X),$$

where $H(\cdot)$ denotes the differential entropy and $H_b(\cdot)$ is the entropy of a Bernoulli random variable with corresponding probability X.

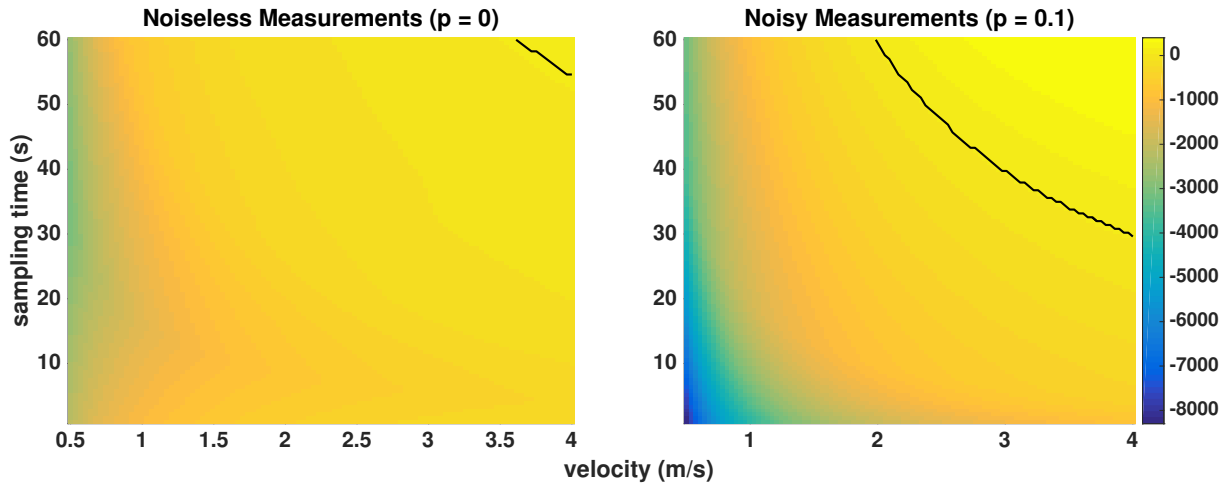


Fig. 6: Difference in sampling time between quantile search and proactive learning under a variety of practical sampling regimes for both noiseless (left) and noisy (right) measurements with $p = 0.1$. Quantile search results in less required time for all points "southwest" of the black line.

The noisy case of Section III-B corresponds to a binary symmetric channel with non-uniform priors [32]. In this case, we have

$$\begin{aligned} I(\theta; X, Y) &= H(\theta) - H(\theta|X, Y) \\ &= H_b(\phi * p) - H_b(p), \end{aligned}$$

where

$$\phi = \int_0^X \pi(x)dx.$$

Note that for $\phi = 1/2$ (i.e., PBS), the mutual information is $1 - H_b(p)$, which is the capacity of a noisy binary symmetric channel. We implement the proactive algorithm from [11], Eqn. (5) with two modifications in order to provide a fair comparison. First, the non-uniform cost in our case is the distance between the current location and the point under consideration, rather than the generic cost described in [11]. Second, we provide a tuning parameter that can be used similarly to m to balance between the number of samples and distance traveled during estimation. Pseudocode for this algorithm is given in Algorithm 3. In both the noiseless and noisy cases, we use the stopping criteria from DQS and PQS, respectively.

To obtain a profile of the performance of proactive learning, we simulate for both noiseless and noisy ($p = 0.1$) measurements, where we range λ over 100 points on the unit interval. We let θ range over a 100-point grid, and 200 random instances are run for each value of θ in the noisy case. To compare with DQS and TPQS, we simulate the average time required to perform sampling on the unit interval under a variety of sampling times and travel times relevant to our problem of sampling in Lake Erie. We let the time per sample η in (5) range from 1-60 s per sample. For travel time, we consider a strip length of 40 km, about half the size of the central basin of Lake Erie, and let the velocity range from 0.5-4 m/s. Fig. 6 shows the difference in sampling time required by quantile search and proactive learning. The boundary of where quantile search outperforms

Algorithm 3 Proactive Learning with Non-Uniform Costs [11] applied to one-dimensional threshold estimation

Input: search parameter $\lambda \in [0, 1]$, probability of error p
Initialize: $\pi_0(x) = 1$ for all $x \in [0, 1]$, $n \leftarrow 0$
while not converged **do**
  $X_{n+1} = \arg\max_x I(\theta; x, Y) - \lambda|X_n - x|$
  $Y_{n+1} \leftarrow f(X_{n+1}) \oplus U_{n+1}$, where $U_{n+1} \sim \text{Ber}(p)$
  $\phi = \int_0^{X_{n+1}} \pi_n(x)dx$
  if $Y_{n+1} = 0$ **then**

$$\pi_{n+1}(x) = \begin{cases} \frac{(1-p)}{\phi * p}\pi_n(x) & x \le X_{n+1} \\ \frac{p}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

  else

$$\pi_{n+1}(x) = \begin{cases} \frac{p}{\phi * p}\pi_n(x) & x \le X_{n+1} \\ \frac{(1-p)}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

  end if
  $n \leftarrow n + 1$
end while
estimate $\hat{\theta}_n$ such that $\int_0^{\hat{\theta}_n} \pi_n(x) = 1/2$

proactive learning is shown in black, so that all points "up and to the right" of the boundary denote sampling regimes in which proactive learning requires less time than quantile search. The figure shows that in the majority of relevant cases, quantile search results in superior performance. However, in the case of large sampling time and high velocity, proactive learning generally performs better. Although not shown, we analyzed figures similar to Figs. 4 and 5 and saw that the number of samples required for proactive learning to converge reduces quickly with λ compared to quantile search, while the distance traveled reduces slowly. Thus, for scenarios in which sampling is significantly more costly than travel, proactive learning may be a more appropriate choice. This is likely due to the fact that proactive learning often takes comparatively large steps early

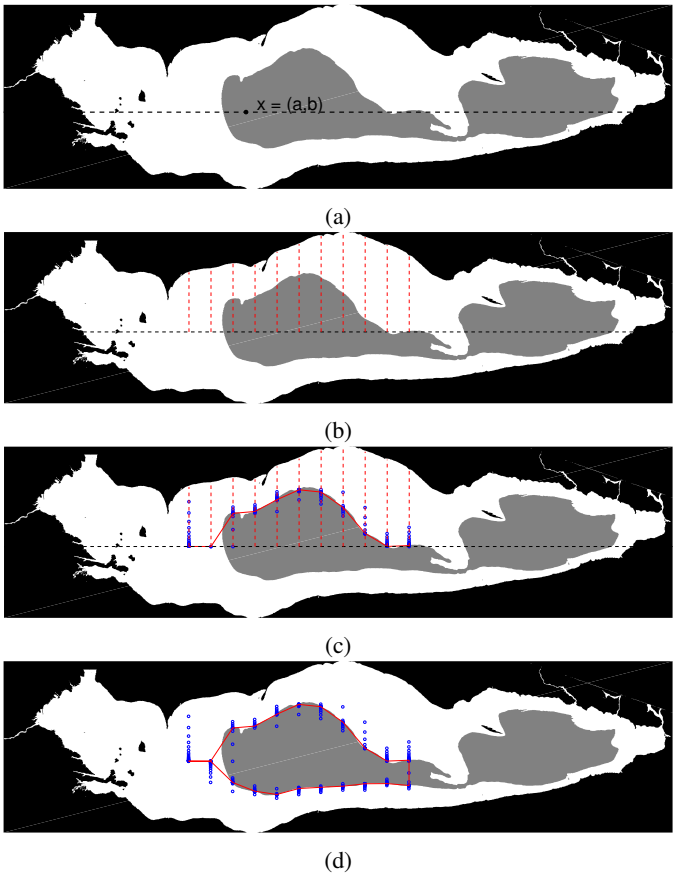


Fig. 7: Proposed sampling procedure for detection of hypoxic region in Lake Erie. (a) Lake Erie with hypoxic region illustrated in gray and split along $x = (a, b)$. (b) Division of top portion into strips. (c) Estimation procedure for top of lake with sample locations shown in blue and estimated boundary in solid red. (d) Final sample locations and estimation of entire boundary.

in the measurement process, and investigating the properties of this algorithm is a topic for our future research.

C. Simulations on Lake Erie

In this section, we apply the quantile search and proactive learning algorithms to the problem of sampling hypoxic regions in Lake Erie. Fig. 7a shows the lake with an example hypoxic zone pictured in gray. In [7], the authors show that for the set of distributions such that the Bayes decision set is a boundary fragment, a variation on PBS can be used to estimate the boundary while achieving optimal rates up to a logarithmic factor. We now describe how the same approach can be used to estimate the hypoxic region in Lake Erie and demonstrate the benefits of our algorithm compared to PBS and proactive learning. The results in this section differ from our previous work [8] in that we consider a more realistic boundary derived from bathymetry data retrieved from [10]. To simulate the boundary of interest, we threshold the bathymetry data at a depth of 21 m and consider anything at a depth of greater than 21 m hypoxic. Although this may not be directly correlated with the hypoxic region, the resulting region is sufficiently irregular to test our algorithm and is visually similar to the regions found in [33]. Further, we previously considered only the time required to estimate the strips (described below) individually, whereas in this work we consider the entire sampling process.

Consider the instance of a hypoxic region shown in Fig. 7a. Using models and measurements from previous years (e.g., historical data from [33]), it is reasonable to assume we can split the lake into intervals so that the boundary does not significantly violate the boundary fragment assumption. Splitting the lake along the line $y = b$ yields the two sets above and below the dashed line in Fig. 7a. Now we can further divide the problem into strips along the first dimension, as shown by the solid red line in Fig. 7b. Along each of these strips, the problem reduces to change point estimation of a one-dimensional threshold classifier as we have studied thus far. After estimating the change point at each strip, the boundary is estimated as a piecewise linear function of the estimates, as shown in Fig. 7c. The same procedure is used for the bottom portion of the lake, with the final estimation shown in Fig. 7d. In all cases, we choose the optimal m by estimating the average number of samples and distance traveled via simulations and note the chosen value in the tables.

We apply this procedure to the hypoxic region shown in Fig. 7a using 11 strips for a variety of values for time per sample and speed of watercraft. To simulate an actual sampling pattern, we proceed counterclockwise through the strips, beginning from the top left, and record the total distance traveled and number of samples taken. We consider several sampling strategies. As a baseline, we use binary bisection with no algorithmic improvements, i.e., quantile search with fixed $m = 2$. We also show DQS with a fixed m chosen to optimize the total sampling time using the average scale factor for the entire lake. Next, we show the sampling time for proactive learning with λ chosen similarly. Finally, we consider these scenarios while employing Improvement 1, where we initialize our search algorithm using the previous boundary estimate. We forego the application of I-2.1 and I-2.2, as they will have minimal impact in the noiseless case. Table I shows the resulting sampling time (in days) required to estimate the boundary of the hypoxic region. When I-1 is not in use, binary search outperforms our algorithm. This is due to the fact that the craft must travel back to the position $1/m$ at each strip, a significant distance when m is small and the boundary estimate is not near the origin. However, this problem is overcome by employing I-1, in which case quantile search requires roughly half the sampling time required by binary search. Further, DQS outperforms proactive learning, even in the scenarios where DQS requires more time on a single strip. In the case of low sampling time and low velocity, we see that DQS significantly outperforms proactive learning, which matches our expectations based on Fig. 6.

Next, we apply the same procedure to the noisy case with a probability of measurement error $p = 0.1$ averaged over 100 random instances. Due to the performance benefits shown in DQS, we employ I-1 in all sampling scenarios. For both I-2.1 and I-2.2, we choose the strip width W based on the number of strips, which is a function of the desired estimation error. Choosing W small will result in more accurate estimation but

Sampling Scenarios	Sampling Time (s) Velocity (m/s)	60 4	60 0.5	10 4	10 0.5
Sampling Parameters	m λ	9.48 0.17	43.00 0.13	43.00 0.13	43.00 0.10
Base Algorithm without Improvements	Bisection DQS Proactive Learning	2.14 2.62 2.84	15.96 19.97 21.20	2.00 2.52 2.67	15.82 19.44 20.45
I-1	Bisection DQS Proactive Learning	1.99 **1.44** 1.47	14.86 **9.41** 9.99	1.86 **1.19** 1.26	14.73 **9.09** 9.62

TABLE I: Total sampling time (in days) for various search methods under noiseless measurements and a variety of sampling times and velocities. Fastest time for each scenario shown in bold.

Sampling Scenarios	Sampling Time (s) Velocity (m/s)	60 4	60 0.5	10 4	10 0.5
Sampling Parameters	m λ	6.40 0.30	11.17 0.29	10.80 0.29	62.16 0.29
I-1	PBS TPQS Proactive Learning	2.64 1.83 **1.72**	19.00 10.84 11.67	2.38 1.38 1.48	18.61 9.57 11.47
I-1, I-2.1	PBS TPQS Proactive Learning	2.63 1.82 1.73	18.66 10.85 11.69	2.37 1.38 1.47	18.66 9.58 11.44
I-1, I-2.2	PBS TPQS Proactive Learning	2.58 1.83 1.73	18.30 **10.75** 11.77	2.33 **1.37** 1.49	18.11 **9.56** 11.52

TABLE II: Total sampling time (in days) for various search methods under noisy measurements with $p = 0.1$ and a variety of sampling times and velocities. Fastest time for each scenario shown in bold.

require more sampling time. In practice one would estimate L using historical data. We estimate L numerically as

$$\hat{L} = \arg\max_i \frac{|f(x_i) - f(x_i + \delta)|}{\delta},$$

where we choose δ to be $0.1W$ to prevent the value of L from being inflated by a single point in f with high derivative. Note that since we are only using L to generate priors for our search function, even an aggressive choice will not prevent our algorithm from finding the true boundary. In some cases, where the function is very smooth in many places and has high derivative in a few places, a user may wish to choose L smaller than the estimated value to reduce sampling time. In I-2.1, we choose c_1 and c_2 such that the probability within LW of $\hat{\theta}^k$ is 100 times the probability outside this region, i.e., $c_1 = 100c_2$.

Table II shows the resulting sampling times under the various sampling scenarios. The results of the table across all sampling parameters indicate that TPQS with a Gaussian prior is the best sampling strategy in most cases. In the case of a 60 sec sampling time and 4 m/s velocity, proactive learning outperforms our algorithm, which is consistent with the results of Fig. 6. Interestingly, the use of nonuniform priors results in a small benefit in most cases. This is likely due to the fact that the bottom half of the hypoxic region is extremely smooth, and hence our value of L is not aggressive enough for these strips. A better choice may be to choose L separately for the strips on top and bottom of the lake.

D. Experiments on Third Sister Lake

In this section, we present the implementation and performance of the DQS algorithm in the field. The algorithm was tested on a robotic boat that was deployed at Third Sister Lake in Ann Arbor, Michigan. Third Sister Lake is a spring-fed kettle lake with an area of 9.4 acres and a maximum depth of 17 meters that notably exhibits hypoxic conditions on an annual basis [34]. The smaller size and calmer waters of Third Sister Lake posed an ideal test bed for evaluating the algorithm. Because of the high fidelity of the oxygen sensor used, only the DQS algorithm was tested in the field. Further, these experiments are intended as a pilot study to motivate the use of our algorithm in larger bodies of water, such as Lake Erie.

The robotic boat platform [35] features an Android cellular phone for GPS navigation and 3G cellular communications. The prevalent cell coverage at Third Sister Lake enables bi-directional communication with the boat for remotely tracking and delineating the evolution of the hypoxic region in real-time. For a given GPS coordinate, the boat autonomously navigates to the destination to collect a sample. The platform was outfitted with a motorized winch to raise and lower a suite of water quality sensors to measure dissolved oxygen throughout the water column at each sampling location. Due to the low noise level of these sensors, we employed the noiseless version of the algorithm with I-1.

Leveraging the persistent Internet connectivity of the robotic boat, the platform was paired with a web-service-based cyberinfrastructure [36]. This enabled the same script used to develop the algorithm to be tested in the field by modifying the script to open a web connection and directly control the boat. Time-stamped location and measurement data were immediately accessible to the algorithm to direct where the boat should sample next. Taking a web-based approach provides the flexibility more readily interface with any web-enabled robotic

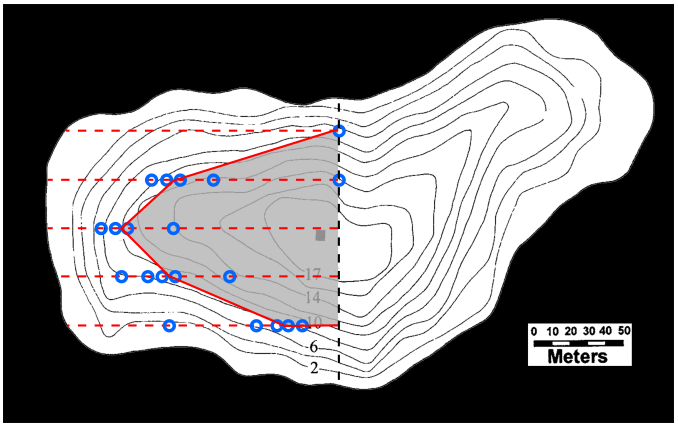


Fig. 8: Delineated hypoxic region on the western half of Third Sister Lake.

boat that may be more suitable for increased winds and waves of more challenging sites.

We present the results from a sampling campaign on November 17, 2015 in Fig. 8. Third Sister Lake was divided into five horizontal strips, along which an average of five samples were taken until GPS precision could no longer distinguish between two locations. The estimated velocity of the robotic boat was 0.1 m/s. Due to the need to lower and raise the winch for each sample location, the average time to collect a sample was 300 s, resulting in an optimal sampling parameter of $m = 2$. We observed that over the course of five hours, the platform successfully identified and delineated the hypoxic zone as directed by the algorithm. In comparison, a uniform sampling at the same resolution would take an estimated 27 hours. The successful results from the experiments on Third Sister Lake demonstrate the potential to extend this algorithm to other lake systems including Lake Erie.

V. Conclusions & Future Work

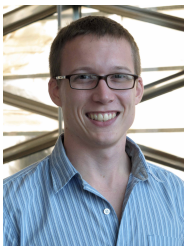
We have presented an active learning algorithm for spatial sampling capable of balancing the number of samples and distance traveled in order to minimize the overall sampling time. To the best of our knowledge, this is the only nonuniformly penalized active learning algorithm accompanied by theoretical guarantees. We have shown how our algorithm can be used to estimate a two-dimensional region of hypoxia under certain smoothness assumptions on the boundary, and empirical results indicate the benefits of quantile search over traditional binary search as well as other active learning methods in the literature.

Several open questions remain. Deriving or bounding the expected distance for the TPQS algorithm is an important next step. The boundary fragment class mentioned here is restrictive [16], and the extension to more general cases would be of interest. The recent work of [37] describes a graph-based algorithm that employs PBS to higher-dimensional nonparametric estimation. Extending this idea to penalize distance traveled is a promising avenue for practical applications of quantile search. Finally, the PQS algorithm requires knowledge of the noise parameter p in order to update the posterior. The algorithms presented in [20], [38] enjoy the property that they are adaptive to unknown noise levels. The development of a noise-adaptive probabilistic search would certainly be of great interest, with potential applications in areas such as stochastic optimization [38] beyond direct applicability to this problem.

References

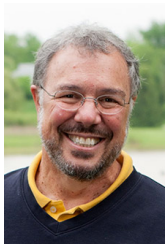
[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[2] B. Settles, *Active Learning*. Morgan & Claypool, 2012.
[3] G. L. E. R. Laboratory, "Lake Erie hypoxia warning system," http://www.glerl.noaa.gov/res/waterQuality/, 2005.
[4] D. Beletsky, D. Schwab, and M. McCormick, "Modeling the 1998-2003 summer circulation and thermal structure in Lake Michigan," *Journal of Geophys. Res.*, vol. 111, 2006.
[5] D. Scavia, J. D. Allan, K. K. Arend, S. Bartell, D. Beletsky, N. S. Bosch, S. B. Brandt, R. D. Briland, I. Daloğlu, J. V. DePinto *et al.*, "Assessing and addressing the re-eutrophication of lake erie: Central basin hypoxia," *Journal of Great Lakes Research*, vol. 40, no. 2, pp. 226–246, 2014.
[6] A. Singh, R. Nowak, and P. Ramanathan, "Active learning for adaptive mobile sensing networks," in *Proc. Information Processing in Sensor Networks*, 2006.
[7] R. Castro and R. Nowak, "Minimax bounds for active learning," *IEEE Trans. Inf. Theory*, vol. 54, pp. 2339–2353, May 2008.
[8] J. Lipor and L. Balzano, "Quantile search: A distance-penalized active learning algorithm for spatial sampling," in *Proc. Allerton Conf. on Communication, Control, and Computing*, 2015.
[9] ——, "Distance-penalized active learning using quantile search," *Supplemental Material*, 2016.
[10] N. Oceanic and A. Administration, "Bathymetry of lake erie & lake saint clair," http://www.ngdc.noaa.gov/mgg/greatlakes/erie.html, 2015.
[11] 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, pp. 619–628.
[12] M. Horstein, "Sequential decoding using noiseless feedback," *IEEE Trans. Inf. Theory*, vol. 9, 1963.
[13] M. V. Burnashev and K. S. Zigangirov, "An interval estimation problem for controlled observations," *Problems in Information Transmission*, vol. 10:223-231, 1974, translated from Problemy Peredachi Informatsii, 10(3):51-61, July-September, 1974.
[14] R. M. Karp and R. Kleinberg, "Noisy binary search and its applications," in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2007.
[15] 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.
[16] R. Castro and R. Nowak, "Active learning and sampling," in *Foundations and Applications of Sensor Management*, 1st ed. New York, NY: Springer, 2008, ch. 8.
[17] R. Waeber, P. I. Frazier, and S. G. Henderson, "Bisection search with noisy responses," *SIAM Journal on Control and Optimization*, vol. 51, pp. 2261–2279, 2013.
[18] A. Ramdas and A. Singh, "Algorithmic connections between active learning and stochastic convex optimization," in *International Conference on Algorithmic Learning Theory*. Springer, 2013, pp. 339–353.
[19] T. Tsiligkaridis, "Asynchronous decentralized algorithms for the noisy 20 questions problem," in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 2699–2703.
[20] Y. Wang and A. Singh, "Noise-adaptive margin-based active learning for multi-dimensional data and lower bounds under tsybakov noise," in *Proc. AAAI Conference on Artificial Intellgence*, 2016.
[21] A. Liu, G. Jun, and J. Ghosh, "Spatially cost-sensitive active learning," in *Proc. SIAM Conf. on Data Mining*, 2009.
[22] B. Demir, L. Minello, and L. Bruzzone, "A cost-sensitive active learning technique for the definition of effective training sets for supervised classifiers," in *Proc. SIAM Conf. on Data Mining*, 2009.
[23] A. Guillory and J. Blimes, "Average-case active learning with costs," in *Proc. Algorithmic Learning Theory*, 2009.
[24] J. Unnikrishnan and M. Vetterli, "Sampling and reconstruction of spatial fields using mobile sensors," *IEEE Trans. Sig. Proc*, vol. 61, pp. 2328–2340, 2013.
[25] ——, "Sampling high-dimensional bandlimited fields on low-dimensional manifolds," *IEEE Trans. Inf. Theory*, vol. 59, pp. 2103–2127, 2013.

[26] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.
[27] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.
[28] H. Bayrum, J. V. Hook, and V. Isler, "Gathering bearing data for target localization," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 369–374, Jan. 2016.
[29] M. Rahimi, M. Hansen, W. J. Kaiser, G. S. Sukhatme, and D. Estrin, "Adaptive sampling for environmental field estimation using robotic sensors," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3692–3698.
[30] B. Schlegel, R. Gemulla, and W. Lehner, "k-ary search on modern processors," in *Proc. Fifth Int. Workshop on Data Management on New Hardware*, 2009.
[31] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
[32] J. Jiang and K. R. Narayanan, "Multilevel coding for channels with non-uniform inputs and rateless transmission over the bsc," in *2006 IEEE International Symposium on Information Theory*. IEEE, 2006, pp. 518–522.
[33] 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*, vol. 47, no. 2, pp. 899–905, 2013.
[34] T. Bridgeman, C. Wallace, G. Carter, R. Carvajal, L. Schiesari, S. Aslam, E. Cloyd, D. Elder, A. Field, K. Schulz *et al.*, "A limnological survey of third sister lake, michigan with historical comparisons," *Lake and Reservoir Management*, vol. 16, no. 4, pp. 253–267, 2000.
[35] A. Valada, P. Velagapudi, B. Kannan, C. Tomaszewski, G. Kantor, and P. Scerri, "Development of a low cost multi-robot autonomous marine surface platform," in *Field and Service Robotics*. Springer, 2014, pp. 643–658.
[36] B. P. Wong and B. Kerkez, "Real-time environmental sensor data: An application to water quality using web services," *Environmental Modelling & Software*, vol. 84, pp. 505–517, 2016.
[37] G. Dasarathy and R. Nowak, "$\mathcal{S}^2$: An efficient graph-based active learning algorithm with application to nonparametric learning," in *Proc. Conf. on Learning Theory*, 2015.
[38] Y. R. Wang and A. Singh, "Algorithmic connections between active learning and stochastic convex optimization," in *Proc. Algorithmic Learning Theory*, 2013.



John Lipor is a Ph.D. candidate in Electrical Engineering at the University of Michigan, degree expected September 2017. He earned his BSEE from the University of Wisconsin and MSEE from the King Abdullah University of Science and Technology (KAUST). John is an awardee of the NSF Graduate Research Fellowship. His research interests include active learning and subspace methods.

Brandon P. Wong is a Ph.D. Candidate at the University of Michigan in Civil and Environmental Engineering working with Professor Branko Kerkez, degree expected December 2017. He obtained a B.S. and M.S. in Civil and Environmental Engineering from UC Berkeley and an M.S. in Electrical Engineering and Computer Science from the University of Michigan. His main research focus is on the development and validation of active, resilient urban water systems for managing stormwater using wireless sensing and control.



Donald Scavia is Professor of Environment and Sustainability and Professor of Environmental Engineering at the University of Michigan (U-M). From 2009-2016, he was the Graham Family Professor of Sustainability, Special Counsel to the U-M President for Sustainability, and Director of the Graham Sustainability Institute. He is a member of the National Academies of Sciences, Engineering, and Medicines Roundtable on Science and Technology for Sustainability. He served previously as Research Associate Dean for the School of Natural Resources and Environment, Director of Michigan Sea Grant, and Director of U-Ms cooperative institute with NOAA. Prior to coming to U-M in 2004, he held positions between 1975 and 2003 as Chief Scientist of NOAA's National Ocean Service, Director of the National Centers for Coastal Ocean Science, and a research scientist at NOAA's Great Lakes Environmental Research Laboratory.

Scavia combines numerical models, empirical analysis, and assessments to improve the predictive understanding of interactions between human activities on land and their impacts on coastal marine and freshwater ecosystems. His research supports integrated assessments that bring together natural and social sciences with practitioners to help shape environmental policy. He has published over 220 articles in the primary literature and books, co-edited two books, and led dozens of interagency scientific assessments and program development plans.

He served on review committees for the National Research Council, USEPA, and NOAA and is on Advisory Boards for the National Wildlife Federation and the Healing our Waters Great Lakes Coalition. He has also served on advisory committees for the Environmental Law and Policy Center, and U-M's Erb Institute for Global Sustainable Enterprise, Risk Science Center, Energy Institute, and the Matthaei Botanical Gardens and Nichols Arboretum. He was Associate Editor for Estuaries and Coasts and Frontiers in Ecology and Environment, and has served on the Boards of Directors for the American Society of Limnology and Oceanography, the International Association for Great Lakes Research, and the Great Lakes Observing System. He received B.S. and M.S. degrees in Environmental Engineering from Rensselaer Polytechnic Institute and a Ph.D. in Environmental Engineering from U-M.

Branko Kerkez is an assistant professor and Berker and Gokyigit Faculty Scholar in the Civil and Environmental Engineering department at the University of Michigan. He obtained his M.S. and Ph.D. in Civil and Environmental Engineering, and an M.S. in Electrical Engineering and Computer Science, all from UC Berkeley. His research interests include water, data and sensors. He is working to enable "smart" water systems, which use sensors and real-time controls to autonomously make decisions. His research projects have spanned wireless sensing of large mountain basins, real-time flood forecasting, large-scale water balance sensor networks, aquatic robotics and the control of urban water systems. He is also the founder of Open-Storm.org, an open source consortium for the sensing and control of water systems.

Laura Balzano Laura Balzano is an assistant professor in Electrical Engineering and Computer Science at the University of Michigan. She is an Intel Early Career Faculty Honor Fellow and received an NSF BRIGE award. She received her Ph.D. in Electrical and Computer Engineering from the University of Wisconsin, Madison along with the ECE Best Dissertation Award. Her main research focus is on optimization for low-rank models, with high-dimensional data that are highly incomplete or corrupted. She works on applications in networks, environmental monitoring, and computer vision.