Quantile Search: A Distance-Penalized Active Learning Algorithm for Spatial Sampling

John Lipor[1], Laura Balzano[1], Branko Kerkez[2], and Don Scavia[3]
[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,girasole,bkerkez,scavia}@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 when the cost of sampling is not only the number of samples but also the distance traveled between samples. 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 number of samples and distance traveled can be achieved using a generalization of binary search, which we refer to as quantile search. We derive the expected total sampling time for noiseless measurements and the expected number of samples for an extension to the noisy case. We illustrate our results in simulations relevant to our sampling application.**

I. Introduction

In many signal reconstruction problems, the cost associated with taking a sample is in some way dependent on the distance between samples. For example, consider sampling the Great Lakes with a small autonomous watercraft in order to reconstruct a spatial signal. Lakes such as Lake Erie span an area on the order of thousands of kilometers; hence the time it takes for the boat to reach the next sample location is a major part of the sampling cost.

In the traditional or *passive* sampling scenario, the sample locations (or sample times) are chosen a priori, often on a uniform grid. Contrasting this is the field of *active sampling* or *active learning* wherein sample locations may be chosen as a function of all previous locations and their corresponding measurements. Active learning theory has shown that binary search guarantees optimal sampling rates [1], but this theory ignores any sampling cost other than the number of samples taken. Our problem of interest is one in spatial sampling where the spatial region of interest is relatively large, and the total sampling time is a function of both the number of samples required *and* the distance traveled throughout the sampling procedure.

Our contribution is an algorithm, termed *quantile search*, that provides a tradeoff between the number of samples required and the distance traveled in finding the change point of a one-dimensional threshold classifier. At its two extremes, quantile search minimizes either the number of samples or the distance traveled, with a tradeoff achieved by varying a search parameter. We derive and analyze this algorithm in the noise-free and noisy cases and study it in simulations with an application to environmental monitoring, the study of oxygen levels in the Great Lakes. The reconstruction of this signal is of interest to water quality experts in order to ensure the balance and quality of water in this region [2]; we show an estimate interpolated from historical measurements in Figure 1.

This work was supported by NSF ECCS 1342121 and F031543-071159-Graduate Research Fellowship Program.

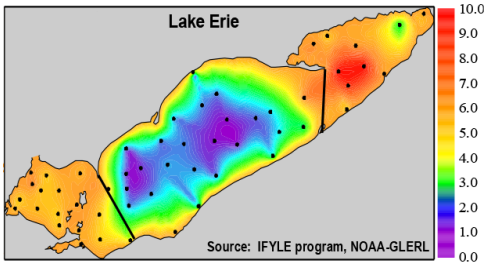

Fig. 1: Dissolved oxygen concentrations in Lake Erie. Points represent sample locations and solid black lines delineate the central basin. Source: http://www.glerl.noaa.gov/res/waterQuality/

Where as binary search takes samples at the bisection of a probability distribution, or the 2-quantile, quantile search instead samples at the first m-quantile. For a preview of our results, suppose that we wish to estimate a threshold θ on a one dimensional interval. For noiseless samples, our results show that the expected estimation error after n measurements is

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

which generalizes to binary search $\mathbb{E}[|\hat{\theta}_n - \theta|] = 2^{-(n+2)}$. The expected distance traveled is

$$\mathbb{E}[D] = \frac{m}{2m-2}, \qquad (2)$$

which also generalizes to binary search where $\mathbb{E}[D] = 1$. Therefore, larger values of m allow for a tradeoff between number of samples and distance traveled.

The remainder of this paper is organized as follows. In Section II, we motivate and formulate the problem of interest. In Section III we place this work into the context of the current literature on active learning. Section IV describes

the quantile search algorithm for both the noise-free and noisy cases. In Section V we verify the theory presented via simulation and apply the algorithms described to a problem of interest. Conclusions and directions of future research are given in Section VI.

II. Motivation and Problem Formulation

The problem of interest for this work is the initial motivating problem described in the Introduction, where we wish to determine hypoxic regions in the Great Lakes. The spatial extent of low oxygen concentration or *hypoxia* is a strong indicator of the health of the lakes [3]. Studies performed by the National Oceanic and Atmospheric Administration on Lake Erie utilize a number of measurement stations on the order of tens, as well as hydrodynamic models, in order to provide a warning system for severe hypoxia in the central basin. We wish to determine the spatial extent of the hypoxic region in the central basin of Lake Erie, a phenomenon that occurs yearly in the late summer months. A region is deemed hypoxic if the dissolved oxygen concentration is less than 0.2 ppm [3]. Further, we assume the hypoxic zone is one connected region with a smooth boundary. The problem can then be considered a binary classification problem, with spatial points receiving a label 0 if they are hypoxic and 1 otherwise, where the desired spatial extent corresponds to the Bayes decision boundary.

To perform sampling, an autonomous watercraft with a speed ranging from 0.5-4 m/s will be used. Further, since the maximum spatial extent may occur at one of many depths, a depth profile must be taken at each point, increasing the time per sample significantly. Finally, the hypoxic zone can only be considered approximately stationary for a period of time less than one week. We therefore seek to estimate the decision boundary while simultaneously minimizing the total time spent sampling.

Following [4], we split our problem into several one dimensional intervals, a process that is described further in Section V. In 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. In contrast to the 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 during sampling. Denote the observations $\{Y_i\}_{i=1}^n \in \{0,1\}^n$ as samples of an unknown function $f_\theta \in \mathcal{F}$ taken at sample locations on the unit interval $\{X_i\}_{i=1}^n$. With probability p, $0 \leq p < 1/2$, we observe an erroneous measurement. Thus

$$Y_i = \begin{cases} f_\theta(X_i) & \text{with probability } 1-p \\ 1 - f_\theta(X_i) & \text{with probability } p \end{cases} = f(X_i) \oplus U_i,$$

where $\oplus$ denotes summation modulo 2, and $U_i \in \{0,1\}$ are Bernoulli random variables with parameter p. While other noise scenarios are common, here we assume the U_i are independent and identically distributed and independent of $\{X_i\}$. This noise scenario is of interest as the motivating data (hypoxia) 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 [17]) remains as a topic for future work.

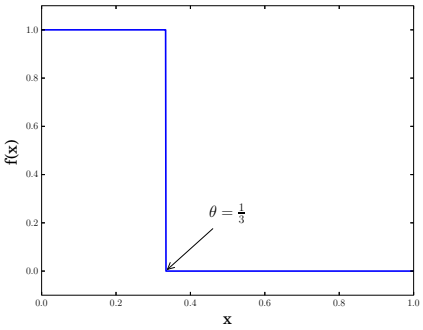

Fig. 2: Threshold classifier in one dimension with change point $\theta = \frac{1}{3}$.

See Figure 2 for an illustration of $f_\theta(X)$. As stated previously, our goal is to estimate θ using as little time as possible. To this end, the sample location at step n is chosen as a function of all previous sample locations and measurements as well as the noise.

III. Related Work

A number of active learning algorithms exist to estimate θ while minimizing the number of samples taken. Binary search or binary bisection has been studied on this problem and on many other active learning problems. Consider binary bisection applied to the signal in Figure 2, where $\theta = 1/3$. Initially, the hypothesis space is the unit interval. The binary bisection algorithm takes its first measurement at $X_1 = 1/2$, measuring $Y_1 = 0$, indicating that θ lies to the left of 1/2, and thus reducing the hypothesis space to the interval $[0, 1/2]$. Continuing in this manner, the algorithm reduces the hypothesis space by half after each measurement, achieving what has been shown to be an optimal rate of convergence [1]. In fact, compared to its passive counterpart (in this case, sample locations distributed uniformly throughout the interval), binary bisection has been shown to achieve an exponential speedup [1], [5]. A large body of work has answered questions such as what type of improvement is attainable in higher dimensions [5], [6], for more complex decision boundaries and under measurement noise [4], [7]–[9], and whether algorithms exist that achieve such improvements [10]–[13].

Extensions of the binary bisection algorithm continue to be an important topic of research in active learning and related fields. In [14], the author presents a modified version of binary search that can handle noisy measurements, referred to as probabilistic binary search (PBS). A discretized version of this algorithm was presented in [7] and analyzed, yielding

an upper bound on the expected rate of convergence of the algorithm. This discretized algorithm, referred to as the B-Z algorithm after its authors, has been analyzed from an information-theoretic perspective [15], [16], as well as under Tsybakov's noise condition [17]. Further, in [17], the authors show that for the boundary fragment class in dimension $d \geq 2$, the B-Z algorithm can be used to perform a series of one-dimensional searches, achieving an optimal rate of convergence up to a logarithmic factor. The algorithm has also been used more recently in [18] to demonstrate the relationship between active learning and stochastic convex optimization.

As we noted before, however, binary search travels on average the entire length of the interval under consideration in order to estimate the threshold. In the case of Lake Erie, one interval corresponds to approximately 58 km, making the use of binary search undesirable.

A number of active learning algorithms exist aside from binary search, many of which have been shown to be optimal under various boundary and noise conditions [4], [6], [8], [10]–[13]. The work of [6], [10]–[13] presents active linear classifiers that are increasingly adaptive to noise conditions and computationally feasible. In [13], the authors present an algorithm that makes use of empirical risk minimization of convex losses. Another approach relies on the use of recursive dyadic partitions (RDPs) [19], in which the d-dimensional unit hypercube is recursively divided into 2^d equal sized hypercubes, forming a tree that is then pruned to give rise to an optimal estimator. An active learning algorithm based on this idea is shown to be nearly optimal in [8], and this algorithm is used in a lake sampling context in [9]. While these algorithms provide optimal sample rates, they both begin by sampling uniformly at random throughout the entire feature space. This would require traversing the entire lake before proceeding to locations where the boundary is likely to lie, rendering these algorithms infeasible.

A small amount of literature exists in which the cost of sampling is generalized beyond the number of samples [20]–[22]. The problems considered in [20], [21] are similar to the setting here, in that the distance traveled to obtain samples is taken into account. In [20], the authors employ a traveling salesman with profits model to minimize the distance traveled while obtaining labels for points of maximum uncertainty. The authors of [21] minimize an objective function that takes into account both label uncertainty and label cost. In both cases, the samples are taken batch-wise before being labeled by a human expert. As every individual sample is informative, we wish to take advantage of all available data by choosing our sample locations on a sample-by-sample basis, rather than updating after each batch. Further, neither algorithm is accompanied by theoretical analysis. In [22], the authors investigate performance bounds for active learning in the case where label costs are nonuniform. The quantile search algorithms presented here extend this idea to the case where label costs are both nonuniform and dynamic.

As a final note, one may wonder about the relation to what is known as m-ary search [24]. In contrast to quantile search, m-ary search is tree-based and thus requires integer values of m. In contrast, quantile search does not require m to be an integer and therefore gives more flexibility in the resulting tradeoff. Further, quantile search as described is the natural generalization of PBS and lends itself to the analysis of [4], [7] in the case where the measurements are noisy. A comparison to noisy m-ary search is a topic for future work.

IV. Quantile Search

In this section, we extend the ideas of [4], [7] to account for a distance penalization in addition to sample complexity, resulting in what we refer to as *quantile search*. 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 ranges 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) and given as Algorithm 1. In the following subsections, we analyze the expected sample complexity and distance traveled for the algorithm and show how these results can be used to minimize the total sampling time. Further, the results show that 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) Rate of Convergence: Recalling from Section II, our goal is to estimate the threshold θ of a one-dimensional threshold classifier. We analyze the expected error of our estimate $\hat{\theta}_n$ 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 [23].

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

$$\mathbb{E}[|\hat{\theta}_n - \theta|] = \frac{1}{4}\left[\rho^2 + (1-\rho)^2\right]^n. \tag{3}$$

Proof: (Sketch) The proof proceeds from the law of total expectation. Let $Z_n = |\hat{\theta}_n - \theta|$. The first measurement is

Algorithm 1 Deterministic Quantile Search (DQS)

input search parameter m, sample budget N
initialize $X_0 \leftarrow 0$, $Y_0 \leftarrow 1$, $n \leftarrow 1$, $a \leftarrow 0$, $b \leftarrow 1$
while $n \leq N$ **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

taken at $1/m$, and hence the expected error can be calculated when $\theta \leq 1/m$ and $\theta > 1/m$.

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

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

Consider the result of Theorem 1 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 V 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 section 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 [23].

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

$$\mathbb{E}[D] = \frac{m}{2m-2}. \tag{4}$$

Proof: (Sketch) The DQS algorithm for our class of functions chooses samples moving further into the interval until it makes a zero measurement. It then turns back and takes samples in the opposite direction until a one is measured. This behavior allows us to analyze the total distance by splitting it up into stages—first the expected distance traveled before the algorithm reaches a point $x_1 > \theta$, and then $x_2 < \theta$, etc. 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

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

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

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

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

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

Next, note that

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

Thus we have

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

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 that form partitions of A_i. The result then follows by induction on $\mathbb{E}[D_n]$. ■

3) Sampling Time: Given the results in Theorems 1 and 2, we can now choose the optimal value of m to minimize the expected sampling time. First, let N be a random variable denoting the number of samples required to achieve an error ε and denote its expectation

$$\mathbb{E}[N] = \frac{\log(4\varepsilon)}{\log\left(\left(\frac{m-1}{m}\right)^2 + \frac{1}{m^2}\right)} \equiv n',$$

which is found by solving (3) for n. Now let T be a random variable denoting the total time required to achieve an error ε. Then the expected value is

$$\begin{aligned}\mathbb{E}[T] &= \gamma\mathbb{E}[N] + \eta\mathbb{E}[D] \\ &= \gamma n' + \eta\left(\frac{m}{2m-2} - \frac{1}{(2m-2)(2m-1)^{n'}}\right) \\ &\approx \gamma n' + \eta\frac{m}{2m-2},\end{aligned} \tag{5}$$

Algorithm 2 Probabilistic Quantile Search (PQS)

input search parameter m, sample budget N, probability of error p
initialize $\pi_0(x) = 1$ for all $x \in [0,1]$, $n \leftarrow 0$
while $n \leq N$ **do**
  choose X_{n+1} such that $\int_0^{X_{n+1}} \pi_n(x)dx = \frac{1}{m}$
  observe $Y_{n+1} \leftarrow f(X_{n+1}) \oplus U_{n+1}$, where $U_{n+1} \sim$ Bern(p)
  if $Y_{n+1} = 0$ **then**

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

  else

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

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

where γ denotes the time required to take one sample, and η denotes the time required to travel one unit of distance. Finding the optimal m is difficult. However, (5) can be used to solve for m numerically using standard techniques. We show examples of such values in Section V.

B. Probabilistic Quantile Search

We now extend the idea behind Section IV-A to the case where measurements may be noisy (*i.e.,* $p \geq 0$). In [7], the authors present the *probabilistic binary search* (PBS) algorithm. 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 [16], [17]. We now extend this idea to achieve a tradeoff between sample complexity and distance traveled.

We refer to the noise-tolerant algorithm as *probabilistic quantile search* (PQS), which proceeds as follows. 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

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

i.e., X_n is the first m-quantile of π_{n-1}. For $m = 2$, the above denotes the median of the posterior distribution and reduces to PBS. A formal description is given in Algorithm 2.

The Bayesian update on π_n can be derived as follows. 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:

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

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

$$\Pr\left(X_1 = 1/m, Y_1 = 0|\theta = 0\right) = 1 - p$$

and

$$\Pr\left(X_1 = 1/m, Y_1 = 1|\theta = 0\right) = p.$$

In fact, this holds for any $\theta < 1/m$. Now examine the denominator:

$$\Pr(X_1 = 1/m, Y_1 = 0) = \frac{1 + (m-2)p}{m}.$$

Notice that for $m = 2$ this simplifies to $1/2$, which is the case in [7]. We then update the posterior distribution to be

$$\pi_1(x) = \begin{cases} (1-p)\left(\frac{m}{1+(m-2)p}\right) & x \leq 1/m \\ p\left(\frac{m}{1+(m-2)p}\right) & x > 1/m. \end{cases}$$

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

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

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

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

1) Rate of Convergence: 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 [25]. Instead, the authors of [4], [7], [16], [17] use a discretized version involving minor modifications. 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 [23].

Given a version of PQS discretized in the same way, we arrive at the following result.

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

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

Proof: (Sketch) In the discretized algorithm, $a_i(j)$ denotes the posterior mass at the ith bin after j measurements and $a(j)$ denotes the vectorized collection of all bins after j measurements. Let $k(\theta)$ be the index of the bin containing θ. Note that $a_i(0) = \Delta$ for all i. We define

$$M_\theta(j) = \frac{1 - a_{k(\theta)}(j)}{a_{k(\theta)}(j)},$$

and

$$N_\theta(j+1) = \frac{M_\theta(j+1)}{M_\theta(j)} = \frac{a_{k(\theta)}(j)(1 - a_{k(\theta)}(j+1)}{a_{k(\theta)}(j+1)(1 - a_{k(\theta)}(j))}.$$

As a brief bit of intuition, note that $M_\theta(j)$ is a measure of how much mass is in the bin actually containing θ, while $N_\theta(j+1)$ is a measure of improvement from one iteration to the next and is strictly less than one when an improvement is made [4]. From [4] and following a straightforward application of Markov's inequality and the law of total expectation, we have that

$$\Pr\left(|\widehat{\theta}_n - \theta| > \Delta\right) \leq \mathbb{E}[M_\theta(n)], \tag{7}$$

and

$$\mathbb{E}[M_\theta(n)] \leq M_\theta(0) \left\{ \max_{j \in 0,\ldots,n-1} \max_{a(j)} \mathbb{E}[N_\theta(j+1)|a(j)] \right\}^n.$$

The remainder of the proof consists of bounding $\mathbb{E}[N_\theta(j+1)|a(j)]$ by considering the three cases: (i) $k(j) = k(\theta)$; (ii) $k(j) > k(\theta)$; and (iii) $k(j) < k(\theta)$. In all cases, we show that

$$\mathbb{E}[N_\theta(j+1)|a(j)] \leq \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\left(\frac{m-2}{m}\right),$$

where $q = 1 - p$ and $\beta = 1 - \alpha$. Using (7), we have

$$\Pr\left(|\hat{\theta}_n - \theta| > \Delta\right) \leq \frac{1-\Delta}{\Delta}\left[\frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right) \times (\beta - \alpha)\left(\frac{m-2}{m}\right)\right]^n.$$

Next, we bound the expected error

$$\begin{aligned}\mathbb{E}[|\hat{\theta}_n - \theta|] &= \int_0^\infty \Pr\left(|\hat{\theta}_n - \theta| > t\right) dt \\ &\leq \Delta + \frac{(1-\Delta)^2}{\Delta}\left[\frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\left(\frac{m-2}{m}\right)\right]^n,\end{aligned}$$

and the result follows by minimizing with respect to Δ and α. ∎

The full proof can be found in [23]. In the case where $m = 2$, the above result matches that of [4], [7] 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 [4], [7], [25], 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 [17] and mentioned in Section IV-A,

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

indicating that the bound is quite loose, even for the previously considered PBS algorithm. However, in [17], 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 Theorem 3 may still be useful in proving 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 IV-A becomes intractable as the values of X_i are no longer deterministic given θ. We have considered the approach of examining the posterior distribution after each step and calculating the possible locations, but at the nth measurement, there are 2^{n-1} possible distributions. Determining the expected distance traveled by PQS remains a subject of future work.

V. Simulations

A. Verification of Algorithms

In this section, we verify through simulation the theoretical rate of convergence and distance traveled derived in Section IV-A. Further, we present simulated results for the PQS algorithm and compare with the bound derived in Section IV-B.

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 Fig. 3a, while the average distance before convergence to an error of $\varepsilon = 1 \times 10^{-4}$ is shown in Fig. 3b. The figures show the theoretical values for expected error and distance match the simulated values. Further, our intuition is confirmed; by inverting the error, one can see that the number of samples required to achieve a given error is monotonically increasing in m, while the distance traveled is monotonically decreasing. This indicates that DQS achieves a tradeoff in the noise-free case. Fig. 3c shows a plot of the expected sampling time as a function of m, with the optimal value of m indicated in the case where $\gamma = 60$ s and $\eta = 4$ m/s.

Next, we simulate the PQS algorithm over a range of m from 2 to 20, where θ ranges over a 100 point grid on the unit interval with 500 Monte Carlo trials run for each value of θ. Fig. 4a shows the average number of samples required to converge to an error of 1×10^{-4}. As in the deterministic case, the required number of samples increases monotonically with m. Fig. 4b shows the average distance traveled before converging to the same error value. Again,

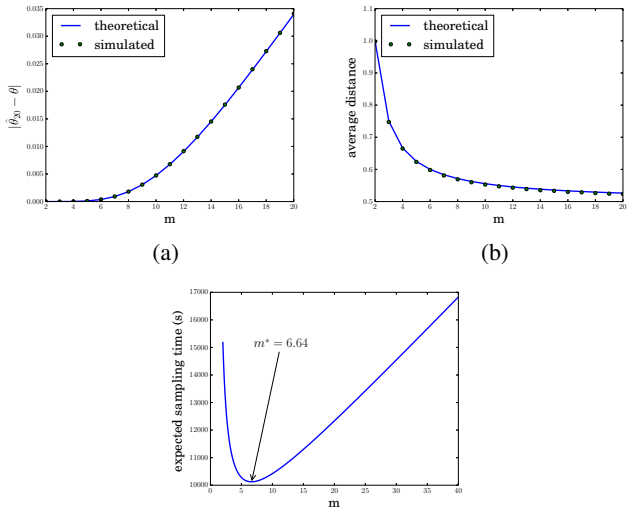

Fig. 3: Simulated and theoretical vales for (a) expected error after 20 samples and (b) distance traveled for DQS algorithm. (c) Optimal value of m for $\gamma = 60$ s and $\eta = 4$ m/s.

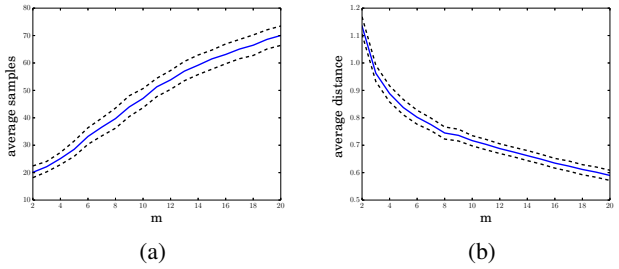

Fig. 4: Simulated average (a) samples required for convergence and (b) distance traveled for PQS algorithm with probability of error $p = 0.1$.

the distance decreases monotonically with m, indicating that the algorithm achieves the desired tradeoff in the noisy case.

B. Application to Spatial Sampling

In this section, we apply the DQS and PQS algorithms to the problem of sampling hypoxic regions in Lake Erie. Fig. 5a shows the lake with an example hypoxic zone pictured in gray. In [17], 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. The authors of [17] informally describe the boundary fragment class on $[0, 1]^d$ 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. For a formal definition, see [17], Defn. 1 and the surrounding text.

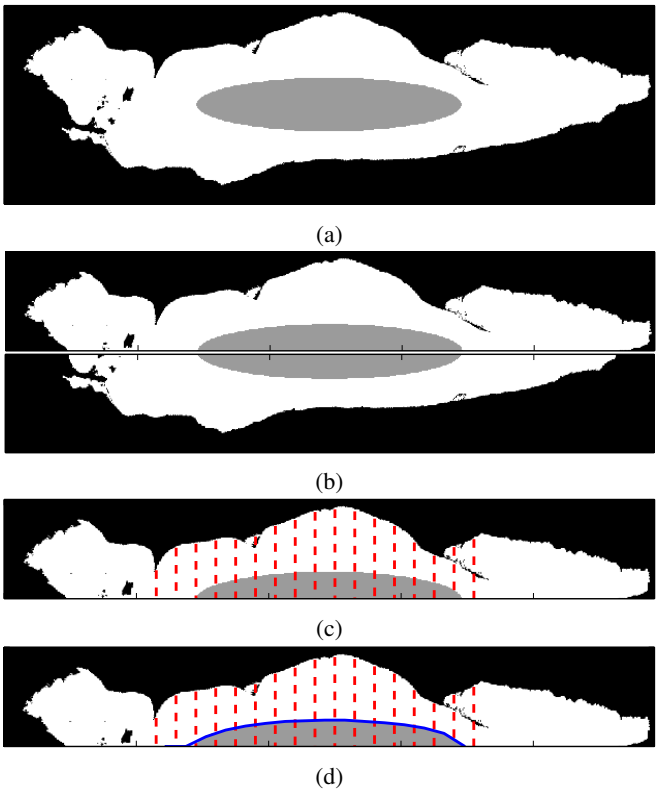

Fig. 5: Example splitting of hypoxic zone in Lake Erie to achieve two boundary fragment sets. (a) Lake erie with hypoxic region shown in gray. (b) Example splitting of hypoxic zone to achieve two boundary fragment sets. (c) Division of boundary fragment into strips. (d) Piecewise linear estimation of boundary.

Given one location $x = (a, b)$ inside the hypoxic zone, the problem can be transformed into two problems, each with a function belonging to the boundary fragment class. Using models and measurements from previous years, this is a reasonable assumption. Splitting the lake along the line $y = b$ yields the two sets shown in Fig. 5b. Now we can further divide the problem into strips along the first dimension, as shown in Fig. 5c. 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. 5d.

We apply this procedure to the hypoxic region shown in Fig. 5a using 16 strips for a variety of values for time per sample and speed of watercraft. These values are used with Theorems 1 and 2 in order to select the optimal value of m. The total time required for both DQS and binary search can be seen in Table I. The results are as expected. A higher sampling time biases the algorithm toward lower values of m in order to minimize the number of samples required, while a lower speed biases the algorithm toward higher values of m in order to minimize the distance traveled. We see that in the case of a low sampling time and low speed, the total time saved compared to binary search is on the order of 60

Sampling Time (s)	Speed (m/s)	m	Total Time (hrs)
60	4	2	62
60	4	6.64	43
60	2	2	123
60	2	8.92	81
10	4	2	61
10	4	14.63	35
10	2	2	122
10	2	20.26	64

TABLE I: Comparison of deterministic quantile search with binary search for a variety of sampling times and speeds.

Sampling Time (s)	Speed (m/s)	m	Total Time (hrs)
60	4	2	65
60	4	6.64	58
60	2	2	129
60	2	8.92	104
10	4	2	64
10	4	14.63	42
10	2	2	127
10	2	20.26	81

TABLE II: Comparison of probabilistic quantile search with binary search for a variety of sampling times and speeds with probability of error $p = 0.1$.

hours. This brings the sampling time from roughly 5 days down to 2-3, decreasing the possibility that the spatial extent of the hypoxic region will change significantly before the measurement is complete. Similar results hold for PQS, as shown in Table II, where we show the sampling times with a probability of measurement error $p = 0.1$ averaged over 100 Monte Carlo simulations. Note that in this case, the chosen value of m may not be optimal.

VI. Conclusions & Future Work

The algorithms described here demonstrate the desired tradeoff between samples required and distance traveled for one-dimensional threshold classifier estimation. We have shown how this idea 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.

Several open questions remain. Deriving or bounding the expected distance for the PQS algorithm is an important next step. Further, while the choice of m shown here is optimal for the DQS algorithm, the question remains whether this algorithm is optimal in some general sense. Beyond this, several interesting generalizations exist. The boundary fragment class mentioned here is restrictive [4], and the extension to more general cases would be of interest. The recent work of [26] 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 [13], [18] 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 [18] beyond direct applicability to this problem.

References

[1] S. Dasgupta, "Analysis of a greedy active learning strategy," in *Proc. Advances in Neural Information Processing Systems*, 2005.

[2] 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.

[3] G. L. E. R. Laboratory, "Lake Erie hypoxia warning system," http://www.glerl.noaa.gov/res/waterQuality/, 2005.

[4] 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.

[5] S. Dasgupta, "Coarse sample complexity bounds for active learning," in *Proc. Advances in Neural Information Processing Systems*, 2005.

[6] S. Hanneke, "A bound on the label complexity of agnostic active learning," in *Proc. ICML*, 2007.

[7] 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.

[8] R. Castro, R. Willett, and R. Nowak, "Faster rates in regression via active learning," in *Proc. Advances in Neural Information Processing Systems*, 2005.

[9] A. Singh, R. Nowak, and P. Ramanathan, "Active learning for adaptive mobile sensing networks," in *Proc. Information Processing in Sensor Networks*, 2006.

[10] M.-F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," in *Proc. International Conference on Machine Learning*, 2006.

[11] S. Dasgupta, D. Hsu, and C. Monteleoni, "A general agnostic active learning algorithm," in *Proc. Advances in Neural Information Processing Systems*, 2007.

[12] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," *Learning Theory*, p. 350, 2007.

[13] Y. Wang and A. Singh, "Noise-adaptive margin-based active learning for multi-dimensional data," *arXiv Preprint*, vol. 1406.5383v1, 2014.

[14] M. Horstein, "Sequential decoding using noiseless feedback," *IEEE Trans. Inf. Theory*, vol. 9, 1963.

[15] R. M. Karp and R. Kleinberg, "Noisy binary search and its applications," in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[16] 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.

[17] R. Castro and R. Nowak, "Minimax bounds for active learning," *IEEE Trans. Inf. Theory*, vol. 54, pp. 2339–2353, May 2008.

[18] Y. R. Wang and A. Singh, "Algorithmic connections between active learning and stochastic convex optimization," in *Proc. Algorithmic Learning Theory*, 2013.

[19] C. Scott and R. Nowak, "Minimax-optimal classification with dyadic decision trees," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1335–1353, Apr. 2006.

[20] A. Liu, G. Jun, and J. Ghosh, "Spatially cost-sensitive active learning," in *Proc. SIAM Conf. on Data Mining*, 2009.

[21] 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.

[22] A. Guillory and J. Blimes, "Average-case active learning with costs," in *Proc. Algorithmic Learning Theory*, 2009.

[23] J. Lipor and L. Balzano, "Quantile search: A distance-penalized active learning algorithm for spatial sampling," *arXiv Preprint*, vol. 1509.08387, 2015.

[24] 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.

[25] 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.

[26] 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.