Active Sensing and Learning

Rui Castro and Robert Nowak

Mistakes are the portals of discovery — James Joyce

1 Introduction

Consider the problem of estimating a signal from noisy samples. The conventional approach (e.g., Shannon-Nyquist sampling) is to sample at many locations in a non-adaptive and more-or-less uniform manner. For example, in a digital camera we collect samples at locations on a square lattice (each sample being a pixel). Under certain scenarios though there is an extra flexibility, and an alternative and more versatile approach is possible: choose the sample locations 'on-the-fly', depending on the information collected up to that time. This is what we call *adaptive sampling*, as opposed to the conventional approach, referred to here as *passive sampling*. Although intuitively very appealing, adaptive sampling is seldom used because of the difficulty of design and analysis of such feedback techniques, especially in complex settings.

The topic of adaptive sampling, or *active learning* as it is sometimes called, has attracted significant attention from various research communities, in particular in the fields of computer science and statistics. A large body of work exists proposing algorithmic ideas and methods [1, 2, 3, 4, 5], but unfortunately there are few performance guarantees for many of those methods. Further most of those results take place in very special or restricted scenarios (*e.g.*, absence of noise or uncertainty, yielding perfect decisions). Under the adaptive sampling framework there are a few interesting theoretical results, some of which are presented here, namely the pioneering work of [6] regarding the estimation of step functions, that was later rediscovered in [7] using different algorithmic ideas and tools. Building on some of those ideas, the work in [8, 9, 10, 11] provides performance guarantees for function estimation under noisy conditions, for several function classes that are particularly relevant to signal processing and analysis.

In this chapter we provide an introduction to adaptive sampling techniques for signal estimation, both in parametric and non-parametric settings. Note that the scenarios we consider do not have the *Markovian* structure inherent to the Markov Decision Processes (MDPs), that are the topic of many chapters in this book, and that the 'actions' (sample locations, or whether or not to collect a sample) do not affect the environment. Another major difference between the active learning problems considered and the MDPs is that, in the former, the set of possible actions/sample locations is generally uncountable.

We begin this chapter with some concrete applications that motivate active learning problems. Section 2 tackles a simple one-dimensional problem that sheds some light on the potential of the active learning framework. In Section 3 we consider higher dimensional function estimation problems, that are more relevant in practical scenarios. Finally in Section 4 some final thoughts and open questions are presented.

1.1 Some Motivation Examples

When trying to learn a new concept, for example, asking someone to describe a scene or a painting, one usually asks questions in a sequential way, just as playing the twenty questions game. One can start by asking if the scene is outdoors, in case the answer is affirmative if there is sky depicted in the scene, if it is overcast or clear, *et cetera*. Note that a key feature of this scheme is the feedback between the learner and the world. On the other hand, most imaging techniques pursue a completely different approach: all the 'questions' are asked in bulk. A digital scanner will give you the value of every pixel in the scene, regardless of the image, essentially by scanning the entire painting at the finest achievable resolution. If scanning time is a critical asset then this approach can be extremely inefficient. If it is known that the scene in question has some nice properties one can accelerate the scanning process by selectively choosing were to scan. To be more concrete consider an airborne laser scanning sensor. This kind of sensor is able to measure range (distance between sensor and observed object) and maybe some properties of the object (*e.g.*, the type of terrain: vegetation, sand, rock, *et cetera*). Suppose we want to use such a setup to construct a topographic map of a field, possibly an hostile environment (*e.g.*, a battlefield). For a variety of considerations (safety concerns, fuel costs, *et cetera*, we would like to limit the time of flight. Figure 1 illustrates the scenario. In this case the field is reasonably "flat", except for a ledge. Clearly, to estimate the topography of the "flat" regions a low resolution scanning would suffice, but to accurately locate the ledge area a higher resolution scanning is needed. If we are pursuing a non-adaptive sampling approach then we need to scan the entire field at the highest resolution, otherwise the ledge area might be inaccurately estimated. On the other hand, if an adaptive sampling technique is used then we can adapt our resolution to the field (based on our observations), thereby focusing the sampling procedure on the ledge area.

There are also other problems that share common characteristics with adaptive sampling, and that are in a sense dual problems. When the computational power is a critical asset (or the computation time is very expensive) one wants to focus the bulk of computation on tasks that are somehow more important (more rewarding). This is the case with most imaging systems (*e.g.*, satellite imaging): although we may face very few restrictions on the total amount of data collected, the subsequent processing of vast amounts of data can be excessively time consuming, therefore carefully choosing "where to look" during the processing phase can lead to significant computational savings. For example, in [12] some of the ideas/techniques of this chapter are applied to the construction

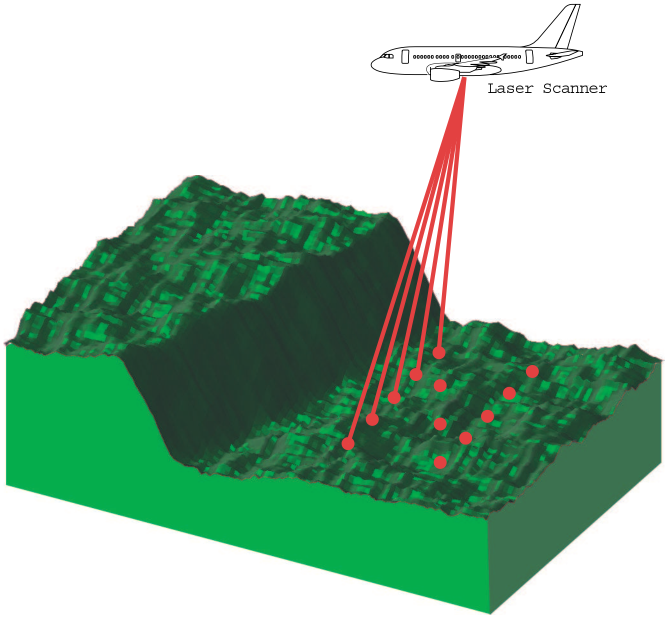

Figure 1: *Airborne range sensor surveying a terrain.*

of fast algorithms for image noise removal.

2 A Simple One-dimensional Problem

In this section we consider the adaptive sampling problem, essentially motivated by the laser scanning scenario. To gain some insight about the power of adaptive sampling we start with a simple, perhaps "toyish" problem. Consider estimating a step function from noisy samples. This problem boils down to locating the step location. Adaptively sampling aims to find this location with a minimal number of strategically placed samples.

Formally, we define the step function class

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

where $\theta \in [0,1]$. This is a parametric class, where each function is characterized by the transition point θ. Given an unknown function $f_\theta \in \mathcal{F}$ our goal is to estimate θ from n point samples. We will work under the following assumptions.

A1.1 - The observations $\{Y_i\}_{i=1}^n$ are point samples of the *unknown* function f_θ, taken at sample locations $\{\boldsymbol{X}_i\}_{i=1}^n$. These are corrupted by noise, that is with probability p we observe a 1 instead of a zero, and vice-versa. Formally

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

where $f_\theta \in \mathcal{F}$, $\oplus$ represents a sumation *modulo* 2 and $U_i \in \{0,1\}$ are Bernoulli random variables, with parameter $0 \leq p < 1/2$, independent and identically distributed (i.i.d.), and independent of $\{X_j\}_{j=1}^n$.

A2.1 - Non-Adaptive Sampling: The sample locations X_i are independent of $\{Y_j\}_{j\neq i}$.

A2.2 - Adaptive Sampling: The sampling location X_i depends only on $\{X_j, Y_j\}_{j<i}$. To be more specific let μ_i be a density defined as

$$\mu_i(X_1, \dots, X_{i-1}, Y_1, \dots, Y_{i-1}).$$

Finally let X_i be a sample taking according to this density. μ_i is called the *sampling strategy*, and completely defines our sampling schedule.

Under the non-adaptive sampling scenario (A2.1) the sample locations do not depend in any way on our observations, therefore the collection of sample points $\{X_i\}_{i=1}^n$ can be chosen before any observations are collected. On the other hand, the adaptive sampling scenario (A2.2) allows for the i^{th} sample location to be chosen using all the information collected up to that point (the previous $i-1$ samples). In either case, our goal is to construct an estimate $\widehat{\theta}_n$ that is "close" to θ, using n samples, where close means that $\sup_{\theta\in[0,1]} |\widehat{\theta}_n - \theta|$ is small.

Consider first the case when there is no noise, that is, $p = 0$. Under the non-adaptive scenario (A2.1) the best we can hope to do is

$$\sup_{\theta\in[0,1]} |\widehat{\theta}_n - \theta| \leq \frac{1}{2(n+1)}.$$

This is achieved distributing the sample locations on a uniform grid over the interval $[0, 1]$,

$$\{X_i\}_{i=1}^n = \left\{\frac{1}{n+1}, \frac{2}{n+1}, \cdots, \frac{n}{n+1}\right\}. \tag{1}$$

Any sample arrangement is going to induce a partition of the interval into $n+1$ intervals (unless there are overlapping samples), therefore we can only decide if the true parameter θ is inside one of these intervals. The performance is limited by the maximum size of these intervals (*i.e.*, the bias of any estimate is limited by the length of these intervals). Clearly the proposed sampling strategy is optimal for passive samples, since any other arrangement of the sample locations (even a randomized one) will lead to a possible degradation in performance.

Now suppose we are working under the adaptive scenario (A2.2). In this situation one can focus on θ much more effectively, using binary bisection: start by taking the first sample at $X_1 = 1/2$. Since there is no noise our observation is simply $Y_1 = f(X_1)$. If $Y_1 = 0$ then we know that $\theta \in [0, 1/2]$ and if $Y_1 = 1$ then $\theta \in (1/2, 1]$. We choose X_2 accordingly: If $Y_1 = 0$ then take $X_2 = 1/4$

and if $Y_1 = 1$ take $X_2 = 3/4$. We proceed according to this technique, always bisecting the set of possibilities. It is easy to see that

$$\sup_{\theta \in [0,1]} |\widehat{\theta}_n - \theta| \leq 2^{-(n+1)}.$$

This is clearly the best one can hope for with this measurement scheme: each measurement provides one bit of information, and with n bits we can encode the value of θ only up to the above accuracy.

If there is noise (*i.e.*, $p > 0$) the techniques one would use to estimate θ have to be modified appropriately. If we are working in the non-adaptive setup (A2.1) there is no reason to change the sampling scheme. We already know that our performance is going to be limited by $1/(2(n+1))$, because of our sampling strategy. To perform the estimation we can use the Maximum Likelihood Estimator (MLE). Define

$$S_n(\theta) = \sum_{i:\boldsymbol{X}_i < \theta} Y_i + \sum_{i:\boldsymbol{X}_i \geq \theta} 1 - Y_i.$$

The MLE estimator of θ is given by

$$\begin{aligned} \widehat{\theta}_n &\equiv \arg\min_{\theta \in [0,1)} \left\{ (1-p)^{n-S_n(\theta)}\ p^{S_n(\theta)} \right\} \\ &= \arg\max_{\theta \in [0,1)} S_n(\theta). \end{aligned}$$

Clearly this optimization has more than one solution, since the value of the likelihood is the same for all $\theta \in (X_i, X_{i+1}]$. For our purposes one can reduce the search to the midpoints of these intervals (*i.e.*, $\hat{\theta} \in \left\{ \frac{1}{2(n+1)}, \frac{3}{2(n+1)}, \cdots, \frac{2n-1}{2(n+1)} \right\}$). It can be shown that this estimator performs optimally, in the sense that

$$\sup_{\theta \in [0,1]} E[|\widehat{\theta}_n - \theta|] \leq C(p) \frac{1}{n+1},$$

where $C(p)$ is an increasing function of p. The derivation of the above result is not at all trivial, and can be accomplished using the oracle bounds presented in [13]. Note that the expected error of this estimator behaves like $1/n$, the same behavior one has when there is no noise present, therefore the maximum likelihood estimator is optimal in this case (up to a constant factor).

If we are working under the adaptive framework, dealing with noise makes things significantly more complicated, in part because our decisions about the sampling depend on all the observations made in the past, which are noisy and therefore unreliable. Nevertheless there is a probabilistic bisection method, proposed in [14], that is suitable for this purpose. The key idea stems from Bayesian estimation. Suppose that we have a prior probability density function $P_0(x)$ on the unknown parameter θ, namely that θ is uniformly distributed over the interval $[0,1]$ (that is $P_0(x) = 1$ for all $x \in [0,1]$). To make the exposition clear assume a particular situation, namely that $\theta = 1/4$. Like before, we start by taking a measurement at $X_1 = 1/2$. With probability $1-p$ we correctly observe a zero, and with probability p we incorrectly observe a one. Suppose a zero was observed. Given these facts we can compute the posterior density simply by applying Bayes rule. In this case we would get that

$$P_1(x|X_1, Y_1) = \left\{ \begin{array}{ll} 2(1-p) & \text{, if } x \leq 1/2, \\ 2p & \text{, if } x > 1/2, \end{array} \right. .$$

The next step is to choose the sample location X_2. We choose X_2 so that is *bisects* the posterior distribution, that is, we take X_2 such that $\Pr_{\theta \sim P_1(\cdot)}(\theta > X_2|X_1, Y_1) = \Pr_{\theta \sim P_1(\cdot)}(\theta < X_2|X_1, Y_1)$. In other words X_2 is just the median of the posterior distribution. If our model is correct, the probability of the event $\{\theta < X_2\}$ is identical to the probability of the event $\{\theta > X_2\}$, and therefore sampling Y_2 at $\boldsymbol{X}_2$ is most informative. We continue iterating this procedure until we have collected n samples. The estimate $\hat{\theta}_n$ is defined as the median of the final posterior distribution. Figure 3 illustrates the procedure and the algorithmic details are described in Figure 2. Note that if $p = 0$ then probabilistic bisection is simply the binary bisection described above.

The above algorithm seems to work extremely well in practice, but it is hard to analyze and there are few theoretical guarantees for it, especially pertaining error rates of convergence. In [6] a similar algorithm was proposed. Albeit its operation is slightly more complicated, it is easier to analyze. That algorithm (which we denote by BZ) uses essentially the same ideas, but enforces a parametric structure for the posterior. Also, in the application of the Bayes rule we use α instead of p, where $0 < \alpha < p$. The algorithm is detailed in Figure 4.

Pertaining the BZ algorithm we have the following remarkable result.

Theorem 1. *Under the assumptions (A1.1) and (A2.2) the Burnashev-Zigangirov algorithm (Figure 2) satisfies*

$$\sup_{\theta \in [0,1]} \Pr(|\widehat{\theta}_n - \theta| > \Delta) \leq \frac{1-\Delta}{\Delta} \left(\frac{1-p}{2(1-\alpha)} + \frac{p}{2\alpha} \right)^n .$$

Taking $\Delta^{-1} = \left(\frac{1-p}{2(1-\alpha)} + \frac{p}{2\alpha} \right)^{-n/2}$ *yields a bound on the expected error*

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

Finally, taking $\alpha = \sqrt{p}/(\sqrt{p}+\sqrt{q})$ *minimizes the right hand side of these bounds, yielding*

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

Remarks: The above theorem shows that, even under noisy assumptions, there is a dramatic improvement in performance if one allows adaptive strategies

Initialization: Define the prior probability density function as $P_0 : [0,1] \to \mathbb{R}$, $P_0(x) = 1$ for all $x \in [0,1]$.

1 - **Sample Selection after i samples were collected:** Define X_{i+1} to be the median of the posterior P_i. That is $X_{i+1} \in [0,1]$ satisfies

$$\int_0^{X_{i+1}} P_i(x)\mathrm{d}x = 1/2.$$

2 - **Noisy Observation:** Observe $Y_{i+1} = f(X_{i+1}) \oplus U_{i+1}$.

3 - **Update posterior:** Update the posterior function. This is simply the application of Bayes rule. If $Y_{i+1} = 0$ then

$$P_{i+1}(x) = \left\{ \begin{array}{ll} 2(1-p)P_i(x) & \text{if } x \leq X_{i+1} \\ 2pP_i(x) & \text{if } x > X_{i+1} \end{array} \right. .$$

If $Y_{i+1} = 1$ then

$$P_{i+1}(x) = \left\{ \begin{array}{ll} 2pP_i(x) & \text{if } x \leq X_{i+1} \\ 2(1-p)P_i(x) & \text{if } x > X_{i+1} \end{array} \right. .$$

4 - **Final estimate:** Repeat steps 1,2 and 3 until n samples are collected. The estimate $\hat{\theta}_n$ is defined as the median of the final posterior distribution, that is, $\widehat{\theta}_n$ is such that

$$\int_0^{\widehat{\theta}_n} P_n(x)\mathrm{d}x = 1/2.$$

Figure 2: The probabilistic bisection algorithm.

(as opposed to passive strategies). Although the bounds display the exponential error decay behavior, also present in the noiseless scenario, the exponent depends on the noise parameter p, and it is clearly not optimal, since when $p \approx 0$ we would expect to obtain approximately the noiseless error bounds (*i.e.*, $\mathbb{E}[|\widehat{\theta}_n - \theta|] \sim (1/2)^n$). Instead a weaker bound is attained, $\mathbb{E}[|\widehat{\theta}_n - \theta|] \sim (\sqrt{2}/2)^n$. It is possible to improve on this bounds by modifying the bounding strategy (as done in [6]). Finally, we note that although this result was derived for a particular noise model the result is applicable to other noise models. This can be done either by processing the observations, using a thresholding operator, or by modifying the likelihood structure (according to the noise model) in the proof of the Theorem).

The proof of Theorem 1 is extremely elegant and is presented below. The ideas in the proof can be used in various other contexts where feedback is present.

Proof of Theorem 1. For the proof we rely on the notation in the algorithm given in Figure 4. In particular recall that the unit interval is divided into subintervals of width Δ, $a_i(j)$ denotes the posterior probability that the changepoint θ is located in the i-th subinterval after the j-th sample, and $\widehat{\theta}_j$ denotes the median of the posterior after j samples.

Our first step is to construct an upper bound for the probability $\Pr(|\widehat{\theta}_n - \theta| > \Delta)$. Let θ be fixed, but arbitrary, and define $k(\theta)$ to be the index of the bin I_i

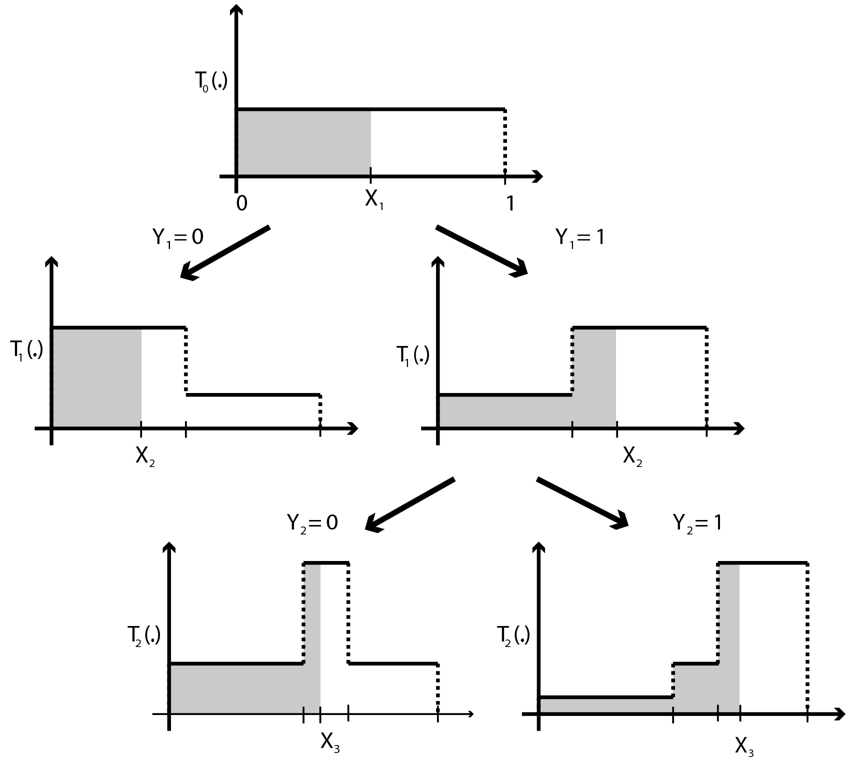

Figure 3: *Illustration of the probabilistic bisection strategy. The shaded areas correspond to* $1/2$ *of the probability mass of the posterior densities.*

containing θ, that is $\theta \in I_{k(\theta)}$. Define

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

and

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

The reasoning behind these definitions is made clear later. For now, notice that $M_\theta(j)$ is a decreasing function of $a_{k(\theta)}(j)$.

After n observations our estimate of θ is the median of the posterior density P_n, which means that $\widehat{\theta}_n \in I_{k(n)}$. Taking that into account we conclude that

$$\begin{aligned} \Pr(|\widehat{\theta}_n - \theta| > \Delta) &\leq \Pr(a_{k(\theta)}(j) < 1/2) \\ &= \Pr(M_\theta(n) > 1) \\ &\leq \mathbb{E}[M_\theta(n)], \end{aligned}$$

Initialization: Let $\Delta > 0$ be such that $\Delta^{-1} \in \mathbb{N}$. Define the posterior after j measurements as $P_j : [0,1] \to \mathbb{R}$,

$$P_j(x) = \Delta^{-1} \sum_{i=2}^{\Delta^{-1}} a_i(j) \mathbf{1}_{I_i}(x),$$

where $I_1 = [0, \Delta]$ and $I_i = (\Delta(i-1), \Delta i]$, for $i \in \{2, \ldots, \Delta^{-1}\}$. Notice that the collection $\{I_i\}$ is a partition of the interval $[0,1]$. We initialize this posterior by taking $a_i(0) = \Delta$. Note that the posterior is completely characterized by $\boldsymbol{a(j)} = \{a_1(j), \ldots, a_{\Delta^{-1}}(j)\}$, and that $\sum_{i=1}^{\Delta^{-1}} a_i(j) = 1$.

1 - Sample Selection: To preserve the parametric structure of the pseudo-posterior we need to take samples at the interval subdivision points. Define $k(j)$ such that

$$\sum_{i=1}^{k(j)-1} a_i(j) \leq 1/2, \quad \sum_{i=1}^{k(j)} a_i(j) > 1/2.$$

Note that $k(j) \in \{1, \ldots, \Delta^{-1}\}$. Select X_{j+1} among $\{\Delta(k(j)-1), \Delta k(j)\}$ by flipping a coin, choosing the first point with probability $\pi_1(j)$ and the second point with probability $\pi_2(j) = 1 - \pi_1(j)$, where $\pi_1(j) = \tau_2(j)/(\tau_1(j) + \tau_2(j))$ and

$$\tau_1(j) = \sum_{i=k(j)}^{\Delta^{-1}} a_i(j) - \sum_{i=1}^{k(j)-1} a_i(j),$$

and

$$\tau_2(j) = \sum_{i=1}^{k(j)} a_i(j) - \sum_{i=k(j)+1}^{\Delta^{-1}} a_i(j).$$

2 - Noisy Observation: Observe $Y_{j+1} = f(X_{j+1}) \oplus U_{j+1}$.

3 - Update posterior: Update the posterior function after collecting the measurement Y_{j+1}, through the application of Bayes rule, under the assumption that the Bernoulli random variables U_{j+1} have parameter $0 \leq \alpha < 1/2$. Let $\beta = 1 - \alpha$. Note that $X_{j+1} = \Delta k$, $k \in \mathbb{N}$ and define

$$\tau = \sum_{i=1}^{k} a_i(j) - \sum_{i=k+1}^{\Delta^{-1}} a_i(j).$$

If $i \leq k$ we have

$$a_i(j+1) = \begin{cases} \frac{2\beta}{1+\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 0 \\ \frac{2\alpha}{1-\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 1 \end{cases},$$

and if $i > k$ we have

$$a_i(j+1) = \begin{cases} \frac{2\alpha}{1+\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 0 \\ \frac{2\beta}{1-\tau(\beta-\alpha)} & \text{if } Y_{j+1} = 1 \end{cases},$$

4 - Final estimate: Repeat steps 1,2 and 3 until n samples are collected. The estimate $\hat{\theta}_n$ is defined as the median of the final posterior distribution, that is, $\widehat{\theta}_n$ is such that

$$\int_0^{\widehat{\theta}_n} P_n(x) \mathrm{d}x = 1/2.$$

Figure 4: The Burnashev-Zigangirov (BZ) algorithm.

where the last step follows from Markov's inequality. The definition of $M_\theta(j)$ above is meant to get more leverage out of Markov's inequality, in a similar spirit of Chernoff bounding techniques. Using the definition of $N_\theta(j)$ and some conditioning we get

$$\begin{aligned}
\mathbb{E}[M_\theta(n)] &= \mathbb{E}[M_\theta(n-1)N_\theta(n)] \\
&= \mathbb{E}\left[\mathbb{E}[M_\theta(n-1)N_\theta(n)|\boldsymbol{a}(n-1)]\right] \\
&= \mathbb{E}\left[M_\theta(n-1)E[N_\theta(n)|\boldsymbol{a}(n-1)]\right] \\
&\ \vdots \\
&= M_\theta(0)E\left[E[N_\theta(1)|\boldsymbol{a}(0)]\cdots E[N_\theta(n)|\boldsymbol{a}(n-1)]\right] \\
&\le M_\theta(0)\left\{\max_{j\in\{0,\ldots,n-1\}}\max_{\boldsymbol{a}(j)} E[N_\theta(j+1)|\boldsymbol{a}(j)]\right\}^n . \qquad (2)
\end{aligned}$$

The rest of the proof consists of showing that $E[N_\theta(j+1)|\boldsymbol{a_j}] \le 1-\epsilon$, for some $\epsilon > 0$. Before proceeding we make some remarks about the above technique. Note that $M_\theta(j)$ measures how much mass is on the bin containing θ (if $M_\theta(j) = 0$ all the mass in our posterior is in the bin containing θ, the least error scenario). The ratio $N_\theta(j)$ is a measure of the improvement (in terms of concentrating the posterior around the bin containing θ) by sampling at $\boldsymbol{X}_j$ and observing Y_j. This is strictly less than one when an improvement is made. The bound (2) above is therefore only useful if, no matter what happened in the past, a measurement made with the proposed algorithm always leads on average to a performance improvement. This is the case with a variety of other useful myopic algorithms.

To study $E[N_\theta(j+1)|\boldsymbol{a}(j)]$ we are going to consider three particular cases: (i) $k(j) = k(\theta)$; (ii) $k(j) > k(\theta)$; and (iii) $k(j) < k(\theta)$. Let $\beta = 1-\alpha$ and $q = 1-p$. After tedious but straightforward algebra we conclude that

$$N_\theta(j+1) = \begin{cases} \frac{1+(\beta-\alpha)x}{2\beta} & \text{, with probability } q \\ \frac{1-(\beta-\alpha)x}{2\alpha} & \text{, with probability } p \end{cases} ,$$

where we have for the three different cases

i)

$$x = \begin{cases} \frac{\tau_1(j)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{, if } X_{j+1} = \Delta(k(j)-1) \\ \frac{\tau_2(j)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{, if } X_{j+1} = \Delta k(j) \end{cases}$$

ii)

$$x = \begin{cases} -\frac{\tau_1(j)+a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{, if } X_{j+1} = \Delta(k(j)-1) \\ \frac{\tau_2(j)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{, if } X_{j+1} = \Delta k(j) \end{cases}$$

iii)

$$x = \begin{cases} \frac{\tau_1(j)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{, if } X_{j+1} = \Delta(k(j)-1) \\ -\frac{\tau_2(j)+a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{, if } X_{j+1} = \Delta k(j) \end{cases}$$

Note that $0 \leq \tau_1(j) \leq 1$ and $0 < \tau_2(j) \leq 1$, therefore $|x| \leq 1$. To ease the notation define

$$\begin{aligned} g(x) &= \frac{q}{2\beta}(1 + (\beta - \alpha)x) + \frac{p}{2\alpha}(1 - (\beta - \alpha)x) \\ &= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)x. \end{aligned}$$

It can be easily checked that $g(x)$ is an increasing function as long as $0 < \alpha < p$. Using this definition we have

i)

$$\begin{aligned} &E[N_\theta(j+1)|\boldsymbol{a}(j)] \\ &= \pi_1(j)g\left(\frac{\tau_1(j) - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) + \pi_2(j)g\left(\frac{\tau_2(j) - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) \end{aligned}$$

ii)

$$\begin{aligned} &E[N_\theta(j+1)|\boldsymbol{a}(j)] \\ &= \pi_1(j)g\left(-\frac{\tau_1(j) + a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) + \pi_2(j)g\left(\frac{\tau_2(j) - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) \end{aligned}$$

iii)

$$\begin{aligned} &E[N_\theta(j+1)|\boldsymbol{a}(j)] \\ &= \pi_1(j)g\left(\frac{\tau_1(j) - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) + \pi_2(j)g\left(-\frac{\tau_2(j) + a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) \end{aligned}$$

Consider first cases (ii) and (iii). Note that $(\tau - a)/(1 - a) \leq \tau$ and $-(\tau + a)/(1 - a) < -\tau$ for all $0 < a < 1$. Therefore, for case (ii) we have

$$\begin{aligned} &E[N_\theta(j+1)|\boldsymbol{a}(j)] \leq \pi_1(j)g(-\tau_1(j)) + \pi_2(j)g(\tau_2(j)) \\ &= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)(-\pi_1(j)\tau_1 + \pi_2(j)\tau_2) \\ &= \frac{q}{2\beta} + \frac{p}{2\alpha}. \end{aligned}$$

Analogously, for case (iii)

$$\begin{aligned} &E[N_\theta(j+1)|\boldsymbol{a}(j)] \leq \pi_1(j)g(\tau_1(j)) + \pi_2(j)g(-\tau_2(j)) \\ &= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)(\pi_1(j)\tau_1 - \pi_2(j)\tau_2) \\ &= \frac{q}{2\beta} + \frac{p}{2\alpha}. \end{aligned}$$

Finally, for case (i) a we need to proceed in a slightly different way. Begin by noticing that $\tau_1(j) + \tau_2(j) = 2a_{k(j)}(j) = 2a_{k(\theta)}(j)$. Then

$$\begin{aligned}
&E[N_\theta(j+1)|\boldsymbol{a}(j)] \\
&= \pi_1(j) g\left(\frac{\tau_1(j) - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) + \pi_2(j) g\left(-\frac{\tau_1(j) - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\right) \\
&= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\frac{\tau_1 - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}(\pi_1(j) - \pi_2(j)) \\
&= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\frac{\tau_1 - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\ \frac{\tau_2(j) + \tau_1(j)}{\tau_1(j) + \tau_2(j)} \\
&= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\frac{\tau_1 - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}\ \frac{2a_{k(\theta)}(j) - 2\tau_1(j)}{\tau_1(j) + \tau_2(j)} \\
&\le \frac{q}{2\beta} + \frac{p}{2\alpha}
\end{aligned}$$

Plugging in the above results into (2) yields

$$\Pr(|\widehat{\theta}_n - \theta| > \Delta) \le \frac{1-\Delta}{\Delta}\left(\frac{q}{2\beta} + \frac{p}{2\alpha}\right)^n,$$

since $M_\theta(0) = (1-\Delta)/\Delta$.

To get a bound on the expected error one proceeds by integration

$$\begin{aligned}
\mathbb{E}[|\widehat{\theta}_n - \theta|] &= \int_0^\infty \Pr(|\widehat{\theta}_n - \theta| > t)\mathrm{d}t \\
&= \int_0^\Delta \Pr(|\widehat{\theta}_n - \theta| > t)\mathrm{d}t + \int_\Delta^1 \Pr(|\widehat{\theta}_n - \theta| > t)\mathrm{d}t \\
&\le \Delta + (1-\Delta)\Pr(|\widehat{\theta}_n - \theta| > \Delta) \\
&\le \Delta + \frac{(1-\Delta)^2}{\Delta}\left(\frac{q}{2\beta} + \frac{p}{2\alpha}\right)^n.
\end{aligned}$$

Choosing Δ as in the statement of the theorem yields the desired result, concluding the proof. □

3 Beyond 1d - Piecewise Constant Function Estimation

In this section we consider again the adaptive sampling scenario, but now in a higher dimensional setting. The one-dimensional setup in the previous section provided us with some insight about the possibilities of active learning, but it is quite restrictive: (i) the function is known up to the location of the step, that is, we know the function takes the values 0 and 1. (ii) One dimensional piecewise functions are extremely simple - they form a parametric class. Nevertheless even this simple type of problem can arise in some practical applications [15]. The kinds of functions we are going to consider next are generally higher dimensional, as in the case of laser field scanning, where the field can be described by a two dimensional function. Also the only prior knowledge we have about these functions is that they are piecewise "smooth", that is, these are composed of smooth regions (where the function varies slowly) separated by low dimensional boundary regions (where the function might change abruptly). One expects active learning to be advantageous for such function classes, since the complexity of such functions is concentrated around the boundary. Pin-pointing the boundary requires many more samples than estimation of the smooth regions so using the active learning paradigm it might be possible to focus most of the samples where they are needed: "near" the boundary. To make the description and discussion simpler we will consider solely piecewise constant functions, whose definition follows.

Definition 1. *A function $f : [0,1]^d \to \mathbb{R}$ is* ***piecewise constant*** *if it is locally constant*[1] *at any point $\boldsymbol{x} \in [0,1]^d \backslash B(f)$, where $B(f) \subseteq [0,1]^d$ is a set with upper box-counting dimension at most $d-1$. Furthermore let f be uniformly bounded on $[0,1]^d$ (that is, $|f(\boldsymbol{x})| \leq M$, $\forall \boldsymbol{x} \in [0,1]^d$) and let $B(f)$ satisfy $N(r) \leq \beta r^{-(d-1)}$ for all $r > 0$, where $\beta > 0$ is a constant and $N(r)$ is the minimal number of closed balls of diameter r covering $B(f)$. The set of all piecewise constant functions f satisfying the above conditions is denoted by $PC(\beta, M)$.*

The concept of box-counting dimension is closely related to topological dimension. The condition on the number of covering balls is essentially a measure of the $d-1$-dimensional volume of the boundary set. Example of such a functions are depicted in Figures 5-7. Note that this class of functions is non-parametric. These functions can provide a simple imaging model in various applications, for example in medical imaging, where one observes various homogeneous regions of tissue of differing densities.

In the rest of the paper we are also going to consider a slightly modified observation model, namely the observations are going to be samples of the function corrupted with additive white Gaussian noise.

A1.2 - The observations $\{Y_i\}_{i=1}^n$ are given by

$$Y_i = f(X_i) + W_i,$$

where $f \in \mathrm{PC}(\beta, M)$ and W_i are i.i.d. Gaussian random variables with zero mean and variance σ^2, and independent of $\{X_j\}_{j=1}^n$.

Under this framework we are mainly interested in answering two questions:

Q1 - What are the limitations of active learning, that is, what is the best performance one can hope to achieve?

[1] A function $f : [0,1]^d \to \mathbb{R}$ is locally constant at a point $\boldsymbol{x} \in [0,1]^d$ if

$$\exists \epsilon > 0 \ : \forall \boldsymbol{y} \in [0,1]^d : \quad \|\boldsymbol{x} - \boldsymbol{y}\| < \epsilon \ \Rightarrow \ f(\boldsymbol{y}) = f(\boldsymbol{x}).$$

Q2 - Can a simple algorithm be devised such that the performance improves on the performance of the best passive learning algorithm?

Before attempting to answer these two questions it is important to know what are the limitations when the passive framework (A2.1) is considered. In [16] the following minimax lower bound is presented.

$$\inf_{\hat{f}_n, S_n} \sup_{f \in \mathrm{PC}(\beta,M)} \mathbb{E}[\|\hat{f}_n - f\|^2] \geq c n^{-\frac{1}{d}}, \tag{3}$$

where $c > 0$ and the infimum is taken with respect to every possible estimator and sample arrangement S_n.

There exist practical passive learning strategies that can nearly achieve the above performance bound. Tree-structured estimators based on *Recursive Dyadic Partitions* (RDPs) are an example of such a learning strategy [17]. These estimators are constructed as follows: (i) Divide $[0,1]^d$ into 2^d equal sized hypercubes. (ii) Repeat this process again on each hypercube. Repeating this process $\log_2 m$ times gives rise to a partition of the unit hypercube into m^d hypercubes of identical size. This process can be represented as a 2^d-ary tree structure (where a leaf of the tree corresponds to a partition cell). Pruning this tree gives rise to an RDP with non-uniform resolution. Let Π denote the class of all possible pruned RDPs. The estimators we consider consist of a stair function supported over a RDP, that is, associated with each element in the partition there is a constant value. Let π be an RDP; the estimators built over this RDP have the form $\tilde{f}^{(\pi)}(\boldsymbol{x}) \equiv \sum_{A \in \pi} c_A \mathbf{1}\{x \in A\}$.

Since the location of the boundary is *a priori* unknown it is natural to distribute the sample points uniformly over the unit cube. There are various ways of doing this; for example, the points can be placed deterministically over a lattice, or randomly sampled from a uniform distribution. We will use the latter strategy. Let $\{\boldsymbol{X}_i\}_{i=1}^n$ be i.i.d. uniform over $[0,1]^d$ and define the *complexity regularized estimator* as

$$\hat{f}_n \equiv \arg \min_{\tilde{f}^{(\pi)}:\pi \in \Pi} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\tilde{f}^{(\pi)}(\boldsymbol{X}_i) - Y_i \right)^2 + \lambda \frac{\log n}{n} |\pi| \right\}, \tag{4}$$

where $|\pi|$ denotes the number of cells of π and $\lambda > 0$. The above optimization can be solved efficiently in $O(n)$ operations using a bottom-up tree pruning algorithm [18, 17].

The performance of the estimator in (4) can be assessed using bounding techniques in the spirit of [19, 17]. From that analysis we conclude that

$$\sup_{f \in \mathrm{PC}(\beta,M)} \mathbb{E}_f[\|\hat{f}_n - f\|^2] \leq C \left(\frac{n}{\log n} \right)^{-\frac{1}{d}}, \tag{5}$$

where the constant factor $C \equiv C(\beta, M, \sigma^2) > 0$. This shows that, up to a logarithmic factor, the rate in (3) is the optimal rate of convergence for passive strategies. A complete derivation of the above result is available in [20].

We now turn our attention to the active learning framework (A2.2). To address question (Q1) we will consider a subclass of the piecewise constant functions defined above, called the boundary fragments. Let $g : [0,1]^{d-1} \to [0,1]$ be a Lipshitz function, that is

$$|g(\boldsymbol{x}) - g(\boldsymbol{z})| \leq \|\boldsymbol{x} - \boldsymbol{z}\|, \; \forall \; \boldsymbol{x}, \boldsymbol{z} \in [0,1]^{d-1}.$$

Define

$$G = \{(\boldsymbol{x}, y) : 0 \leq y \leq g(\boldsymbol{x}), \; (\boldsymbol{x}, y) \in [0,1]^d\}, \tag{6}$$

and let $f : [0,1]^d \to \mathbb{R}$ be defined as $f(\boldsymbol{x}) = M\mathbf{1}_G(\boldsymbol{x})$. The class of all functions of this form is called the *boundary fragment* class (usually taking $M = 1$), denoted by $\mathrm{BF}(M)$. An example of a boundary fragment function is depicted in Figure 5(a). It is straightforward to to show that $\mathrm{BF}(M) \subseteq \mathrm{PC}(\beta, M)$, for a suitable constant β. In [8] it was shown that under (A1.2) and A(2.2) we have the lower bound

$$\inf_{\hat{f}_n, S_n} \sup_{f \in \mathrm{BF}(M)} \mathbb{E}[\|\hat{f}_n - f\|^2] \geq cn^{-\frac{1}{d-1}},$$

for n large enough, where $c \equiv c(M, \sigma^2) > 0$, and the infimum is taken with respect to every possible estimator and sampling strategy. Since $\mathrm{BF}(M) \subseteq \mathrm{PC}(\beta, M)$ it follows that

$$\inf_{\hat{f}_n, S_n} \sup_{f \in \mathrm{PC}(\beta, M)} \mathbb{E}[\|\hat{f}_n - f\|^2] \geq cn^{-\frac{1}{d-1}}, \tag{7}$$

The above results are restricted to $d \geq 2$. Note that the error rates for the adaptive sampling framework display very significant improvement. For example, for $d = 2$ the passive learning error rate is $O(1/\sqrt{(}n))$, which is significantly slower that the active learning error rate of $O(1/n)$. Note that, for this two-dimensional case, the active learning rate coincides with the classical parametric rate, although this class of functions is non-parametric. In [8] an algorithm capable of achieving the above rate for the boundary fragment class is also presented. This algorithm takes advantage of the very special functional form of the boundary fragment functions. The algorithmic idea is very simply: begin by dividing the unit hypercube into $O(n/\log(n))$ “strips” and perform a one-dimensional change-point estimation in each of the strips (using the BZ algorithm with $\log(n)$ samples). This process is illustrated in Figure 5(b).

Unfortunately, the boundary fragment class is very restrictive and impractical for most applications. Recall that boundary fragments consist of only two regions, separated by a boundary that is a function of the first $d-1$ coordinates. For a general piecewise constant function the boundaries oriented arbitrarily and generally are not aligned with any coordinate axis such that they can be described in a functional way. The class $\mathrm{PC}(\beta, M)$ is much larger and more general, so the algorithmic ideas that work for boundary fragments can no longer be used. A completely different approach is required, using radically different tools.

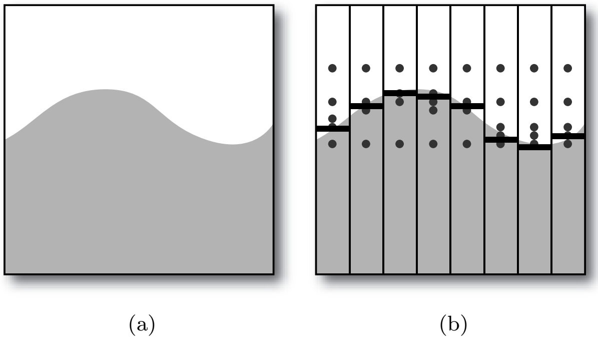

Figure 5: (a) Illustration of a boundary fragment function for $d = 2$. (b) Adaptive sampling for boundary fragments. In each vertical stripe one uses the BZ algorithm to estimate a step function. The final estimate is a piecewise constant function whose boundary is a stair function.

We now attempt to answer question (Q2), by proposing an active learning scheme for the piecewise constant class. The scheme is a two-step approach motivated by the tree-structured estimators for passive learning described above. Although the ideas and intuition behind the approach are quite simple, the formal analysis of the method is significantly difficult and cumbersome, therefore the focus of the presentation is on the algorithm and sketch of the proofs, deferring the details to the references. The main idea is to devise a strategy that uses the first sampling step to find advantageous locations for new samples, to be collected at the second step. More precisely in the first step, called the *preview step*, a rough estimator of f is constructed using $n/2$ samples (assume for simplicity that n is even), distributed uniformly over $[0,1]^d$. In the second step, called the *refinement step*, we select $n/2$ samples near the perceived location of the boundaries (estimated in the preview step) separating constant regions. At the end of this process we will have half the samples concentrated in the perceived vicinity of the boundary set $B(f)$. Since accurately estimating f near $B(f)$ is key to obtaining faster rates, the strategy described seems quite sensible. However, it is *critical* that the preview step is able to detect the boundary with very high probability. If part of the boundary is missed, then the error incurred is going to propagate into the final estimate, ultimately degrading the performance. Conversely, if too many regions are (incorrectly) detected as boundary locations in the preview step, then the second step will distribute samples too liberally and no gains will be achieved. Therefore extreme care must be taken to accurately detect the boundary in the preview step, as described below.

Preview: The goal of this stage is to provide a coarse estimate of the location of $B(f)$. Specifically, collect $n' \equiv n/2$ samples at points distributed uniformly over $[0,1]^d$. Next proceed by using the passive learning algorithm described before, but restrict the estimator to RDPs with leafs at a maximum depth of $J = \frac{d-1}{(d-1)^2+d} \log(n'/\log(n'))$. This ensures that, on average, every

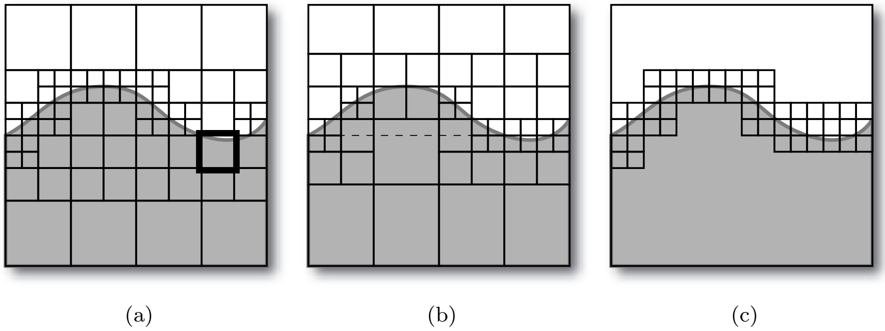

Figure 6: Illustration of the shifted RDP construction for $d = 2$: (a) RDP used obtained from the preview step with the regular RDPs. The highlighted cell intersects the boundary but it was pruned, since the pruning does not incur in severe error. (b) Shifted RDP, Obtained from the preview step over vertically shifted RDPs . In this case there is no pruning, since it would cause a large error. (c) These are the cells that are going to be refined in the refinement stage.

element of the RDP contains many sample points; therefore we obtain a low variance estimate, although the estimator bias is going to be large. In other words, we obtain a very “stable” coarse estimate of f, where stable means that the estimator does not change much for different realizations of the data. The justification for the particular value of J arises in the formal analysis of the method.

The above strategy ensures that most of the time, leafs that intersect the boundary are at the maximum allowed depth (because otherwise the estimator would incur too much empirical error) and leafs away from the boundary are at shallower depths. Therefore we can “detect” the rough location of the boundary just by looking at the deepest leafs. Unfortunately, if the set $B(f)$ is somewhat aligned with the dyadic splits of the RDP, leafs intersecting the boundary can be pruned without incurring a large error. This is illustrated in Figure 7(b); the cell with the arrow was pruned and contains a piece of the boundary, but the error incurred by pruning is small since that region is mostly a constant region. However, worst-case analysis reveals that the squared bias induced by these small volumes can add up, precluding the desired improved performance. A way of mitigating this issue is to consider multiple RDP-based estimators, each one using RDPs appropriately shifted. We use $d + 1$ estimators in the preview step: one on the initial uniform partition, and d over partitions whose dyadic splits have been translated by 2^{-J} in each one of the d coordinates. Any leaf that is at the maximum depth on any of the $d + 1$ RDPs pruned in the preview step indicates the highly probable presence of a boundary, and will be refined in the next stage. This shifting strategy is illustrated in Figure 6

Refinement: With high probability, the boundary is contained in the leafs at the maximum depth. In the refinement step we collect additional $n/2$ samples

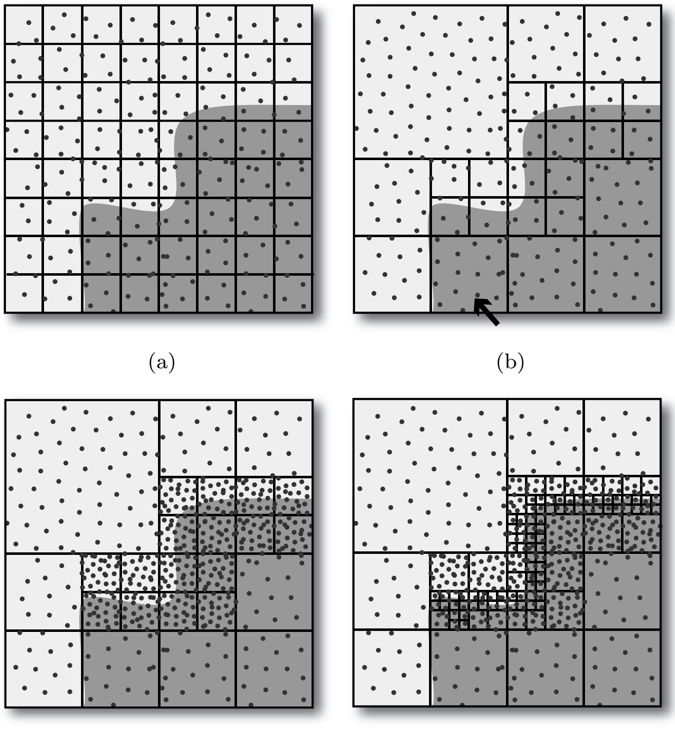

Figure 7: The two step procedure for $d = 2$: (a) Initial unpruned RDP and $n/2$ samples. (b) Preview step RDP. Note that the cell with the arrow was pruned, but it contains a part of the boundary. (c) Additional sampling for the refinement step. (d) Refinement step.

on the corresponding partition cells, using these to obtain a refined estimate of the function f by applying again an RDP-based estimator. This produces a higher resolution estimate in the vicinity of the boundary set $B(f)$, yielding a better performance than the passive learning technique.

The final estimator is constructed assembling the estimate "away" from the boundary obtained in the preview step with the estimate in the vicinity of the boundary obtained in the refinement step.

To formally show that this algorithm attains the faster rates we desire we have to consider a further technical assumption, namely that the boundary set is "cusp-free"[2]. This condition is rather technical, but it is not very restrictive, and encompasses many interesting situations, including of course, boundary fragments. This condition seems to be necessary for the algorithm to perform well, and it is not simply an artifact of the proof. For a more detailed explanation see [20]. Under this condition we have the following theorem.

[2] A cusp-free boundary cannot have the behavior you observe in the graph of $|x|^{1/2}$ at the origin. Less "aggressive" kinks are allowed, such as in the graph of $|x|$.

Theorem 2. *Under the active learning scenario (A1.2) and (A2.2) we have, for $d \geq 2$ and functions f whose boundary is cusp-free,*

$$\mathbb{E}\left[\|\hat{f}_n - f\|^2\right] \quad \leq C\left(\frac{n}{\log n}\right)^{-\frac{1}{d-1+1/d}}, \tag{8}$$

where $C > 0$.

This bound improves on (5), demonstrating that this technique performs better than the best possible passive learning estimator. The proof of Theorem 2 is quite involved and is presented in detail in [20]. Below we present only a sketch of the proof.

A couple of remarks are important at this point. Instead of a two-step procedure one can iterate this algorithm, performing multiple steps (*e.g.*, for a three-step approach replace the refinement step with the two-step approach described above). Doing so can further improve the performance. One can show that the expected error will decay like $n^{-1/(d-1+\epsilon)}$, with $\epsilon > 0$, given a sufficiently large number of steps. Therefore we can get rates arbitrarily close to the lower bound rates in (7). These multi-step methods start to lack practical usefulness if the number of steps is too large, since the practical benefits appear only for a large number of samples. The two step procedure on the other hand displays excellent performance under practical scenarios as seen in [21].

Proof of Theorem 2. The main idea behind the proof is to decompose the error of the estimator for three different cases: (i) the error incurred during the preview stage in regions "away" from the boundary; (ii) the error incurred by not detecting a piece of the boundary (and therefore not performing the refinement step in that area); (iii) the error remaining in the refinement region at the end of the process. By restricting the maximum depth of the trees in the preview stage we can control the type-(i) error, ensuring that it does not exceed the error rate in (8). We start by defining f_J, a coarse approximation of f up to resolution J. Consider the partition of $[0,1]^d$ into 2^{dJ} identical hypercubes and denote this partition by π_J. Note that this partition can also be described by a RDP, where all the leafs are at depth J. Define $f_J : [0,1]^d \to \mathbb{R}$ as

$$f_J(\boldsymbol{x}) = \frac{1}{2^{dJ}} \sum_{A \in \pi_j} \left(\int_A f(\boldsymbol{t})\mathrm{d}t\right) \mathbf{1}_A(\boldsymbol{x}).$$

Note that f_J is identical to f "away" from the boundary, but in the vicinity of the boundary there is some averaging. We have the following key Lemma.

Lemma 1. *Let $\hat{f}^p$ be the complexity regularized estimator of the preview step, using $n' = n/2$ samples. Then*

$$\mathbb{E}[\|\hat{f}^p - f_J\|] \leq C\frac{2^{(d-1)J}\log(n')}{n'},$$

for a suitable $C > 0$ and all $n' > 2$.

Lemma 1 characterizes the behavior of the final estimate "away" from the boundary, since f and f_J are identical in that region. So the above error bound controls the type-(i) error.

Type-(ii) error corresponds to the situations when a part of the boundary was not detected in the preview step. This can happen because of the inherent randomness of the noise and sampling distribution, or because the boundary is somewhat aligned with the dyadic splits, like in Figure 7(b). The latter can be a problem and this is why one needs to perform $d + 1$ preview estimates over shifted partitions. If the boundary is cusp-free then it is guaranteed that one of those preview estimators is going to "feel" the boundary since it is not aligned with the corresponding partition. A piece of the boundary region is not refined if it is not detected in *all* the shifted partition estimators. The worst-case error can be shown not to exceed $C2^{(d-1)J}\log(n')/n'$, for some $C > 0$, therefore failure to detect the boundary has the same contribution for the total error as the type-(i) error. The proof of this fact is detailed in [20].

Finally, analysis of type-(iii) error is relatively easy. Nonetheless one needs to make sure that the size of the region that needs to be refined is not too large, since in that case the density of samples in the refinement step might be not be sufficient to improve on passive methods. In other words, one needs to make sure that in the preview step not much more than the boundary region is detected. Let $\mathcal{R}$ be the set of cells that need to be re-sampled in the refinement step. One has to guarantee that, with high probability the number of boundary cells detected in the preview step (denoted by $|\mathcal{R}|$) is on the order of $2^{(d-1)J}$. The following Lemma [20] provides an affirmative answer.

Lemma 2. *Let $|\mathcal{R}|$ be the number of cells detected in the preview step, that possibly contain the boundary (and therefore are going to be re-sampled in the refinement step). Then*

$$\Pr(|\mathcal{R}| > C2^{(d-1)J}) \leq 1/n,$$

for $C > 0$ and n sufficiently large.

In the regions that are going to be refined, that is, the regions in $\mathcal{R}$, we are going to collect further samples and apply the passive estimation strategy described in (4). As shown in the Lemma 2 we can assume that there are $O(2^{(d-1)J})$ elements in $\mathcal{R}$ (with high probability). We collect a total of $L \equiv n/(2|\mathcal{R}|)$ samples in each element of $\mathcal{R}$. The error incurred by $\hat{f}_r$, the refinement estimator, over each one of the elements of $\mathcal{R}$ is upper-bounded by

$$C\left(\frac{\log L}{L}\right)^{1/d} 2^{-dJ},$$

where $C > 0$ comes from (5), and 2^{-dJ} is just the volume of each element of $\mathcal{R}$. Therefore overall error contribution of the refinement step is upper-bounded by

$$C\left(\frac{\log L}{L}\right)^{1/d} 2^{-dJ}|\mathcal{R}|.$$

To compute the total error incurred by $\hat{f}_{\text{active}}$, our proposed active learning estimator, we just have to sum the contributions of (i), (ii) and (iii), and therefore we get

$$\mathbb{E}\left[\|\hat{f}_{\text{active}} - f\|^2\right] \leq C \left(\frac{\log L}{L}\right)^{1/d} 2^{-dJ}|\mathcal{R}| + C' \frac{2^{(d-1)J}\log n}{n},$$

with $C, C' > 0$. Assuming now that $|\mathcal{R}| = O(2^{(d-1)J})$ we can balance the two terms in the above expression by choosing

$$J = \left\lceil \frac{d-1}{(d-1)^2 + d} \log(n/\log(n)) \right\rceil,$$

yielding the desired result. □

4 Final Remarks and Open Questions

The results presented in this chapter show that for certain scenarios active learning attains provable gains over the classical passive approaches, even if observation uncertainty is present. Active learning is intuitively appealing, and finds applications for many practical problems, for example in imaging techniques, some of them described here. Despite these benefits, the analysis of such active methods is quite challenging due to the existence of feedback in the measurement process. This creates statistical dependence in the observations (recall that now the sample locations are coupled with all the observations made in the past), precluding the use of the usual analysis tools, such as concentration inequalities and laws of large numbers, that require independence (or *quasi* independence) to be applicable. The piecewise constant function class studied provides a non-trivial canonical example that illustrates under what conditions one might expect the adaptive sampling framework to yield a performance improvement over more traditional passive sampling techniques. The algorithm presented here for actively learning members of the piecewise constant class demonstrates the possibilities of active learning in more general settings. A natural extension of this function class is the piecewise smooth class, whose element are Hölder smooth functions[3] with smoothness parameter α. For this class of functions we conjecture that the best attainable performance under assumptions (A1.2) and (A2.2) is $O(\max\{n^{-1/(d-1)}, n^{-2\alpha/(2\alpha+d)}\})$. This is quite intuitive, since it is known that adaptive sampling is not effective for learning smooth functions [11]. Constructing a two-step algorithm to learn such functions is relatively simple and it has been done in the context of field estimation

[3] A function $f : [0,1]^d \to \mathbb{R}$ is *Hölder smooth* around $\boldsymbol{x}$ if it has continuous partial derivatives up to order $k = \lfloor \alpha \rfloor$ at point $\boldsymbol{x}$ and

$$\exists \epsilon > 0 : \forall\ \boldsymbol{z} \in [0,1]^d : \quad \|\boldsymbol{z} - \boldsymbol{x}\| < \epsilon \ \Rightarrow\ |f(\boldsymbol{z}) - P_{\boldsymbol{x}}(\boldsymbol{z})| \leq L\|\boldsymbol{z} - \boldsymbol{x}\|^{\alpha},$$

where $L, \alpha > 0$, and $P_{\boldsymbol{x}}(\cdot)$ denotes the order k Taylor polynomial approximation of f expanded around $\boldsymbol{x}$, and $k = \lfloor \alpha \rfloor$ is the maximal integer such that $k < \alpha$.

using wireless sensor networks [21]. The key modification is that now one needs to decorate the estimator RDPs with polynomials instead of constants. Despite its simplicity this algorithm is very hard to analyze (to be specific, it is difficult to generalize Lemma 1 in the proof).

Under the selective sampling framework there are even more open problems, that have recently spawned much interest. It is a known fact that most existing active learning algorithms tend to be too greedy: They work well when the number of collected samples/examples is small, but the performance quickly degrades as that number increases, leading to results that are worse than when using classical passive learning techniques. This creates some fertile ground for both practitioners and theoretical researchers, and an interesting interplay between the two.

References

[1] D. Cohn, Z. Ghahramani, and M. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, pp. 129–145, 1996.

[2] D. J. C. Mackay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, pp. 698–714, 1991.

[3] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Information, prediction, and query by committee," *Proc. Advances in Neural Information Processing Systems*, 1993.

[4] K. Sung and P. Niyogi, "Active learning for function approximation," *Proc. Advances in Neural Information Processing Systems*, vol. 7, 1995.

[5] G. Blanchard and D. Geman, "Hierarchical testing designs for pattern recognition." to appear in Annals of Statistics, 2005.

[6] M. V. Burnashev and K. S. Zigangirov, "An interval estimation problem for controlled observations," *Problems in Information Transmission*, vol. 10, pp. 223–231, 1974. (Translated from *Problemy Peredachi Informatsii,* 10(3):51–61, July-September, 1974. Original article submitted June 25, 1973).

[7] P. Hall and I. Molchanov, "Sequential methods for design-adaptive estimation of discontinuities in regression curves and surfaces," *The Annals of Statistics*, vol. 31, no. 3, pp. 921–941, 2003.

[8] A. P. Korostelev, "On minimax rates of convergence in image models under sequential design," *Statistics & Probability Letters*, vol. 43, pp. 369–375, 1999.

[9] A. Korostelev and J.-C. Kim, "Rates of convergence for the sup-norm risk in image models under sequential designs," *Statistics & probability Letters*, vol. 46, pp. 391–399, 2000.

[10] G. Golubev and B. Levit, “Sequential recovery of analytic periodic edges in the binary image models,” *Mathematical Methods of Statistics*, no. 12, pp. 95–115, 2003.

[11] R. Castro, R. Willett, and R. Nowak, “Faster rates in regression via active learning,” in *Proceedings of Neural Information Processing Systems (NIPS)*, 2005.

[12] R. Castro, R. Willett, and R. Nowak, “Coarse-to-fine manifold learning,” in *Proceedings of the International Conference on Accoustics, Speech and Signal Processing (ICASSP)*, (May, Montreal, Canada), 2004.

[13] E. D. Kolaczyk and R. D. Nowak, “Multiscale likelihood analysis and complexity penalized estimation,” *Annals of Statistics*, vol. 32, no. 2, pp. 500–527, 2004.

[14] M. Horstein, “Sequential decoding using noiseless feedback,” *IEEE Trans. Info. Theory*, vol. 9, no. 3, pp. 136–143, 1963.

[15] A. Singh, R. Nowak, and P. Ramanathan, “Active learning for adaptive mobile sensing networks,” in *Proceedings of 5th International Conference on Information Processing in Sensor Networks (IPSN ’06)*, (Nashville, TN 2006), April 19-21, 2006.

[16] A. Korostelev and A. Tsybakov, *Minimax Theory of Image Reconstruction.* Springer Lecture Notes in Statistics, 1993.

[17] R. Nowak, U. Mitra, and R. Willett, “Estimating inhomogeneous fields using wireless sensor networks,” *IEEE Journal on Selected Areas in Communication*, vol. 22, no. 6, pp. 999–1006, 2004.

[18] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Trees.* Belmont, CA: Wadsworth, 1983.

[19] A. R. Barron, “Complexity regularization with application to artificial neural networks,” in *Nonparametric Functional Estimation and Related Topics*, pp. 561–576, Kluwer Academic Publishers, 1991.

[20] R. Castro, R. Willett, and R. Nowak, “Faster rates in regression via active learning,” tech. rep., University of Wisconsin, Madison, October 2005. ECE-05-3 Technical Report.

[21] R. Willett, A. Martin, and R. Nowak, “Backcasting: Adaptive sampling for sensor networks,” in *Proc. Information Processing in Sensor Networks*, (26-27 April, Berkeley, CA, USA), 2004.