Toward a Liquid Biopsy: Greedy Approximation Algorithms for Active Sequential Hypothesis Testing

Kyra Gan*, Su Jia*, Andrew Li, Sridhar Tayur
Tepper School of Business, Carnegie Mellon University
{kyragan, sujia, aali1, stayur}@cmu.edu

This paper addresses a set of active learning problems that occur in the development of liquid biopsies via the lens of *active sequential hypothesis testing* (ASHT). In the problem of ASHT, a learner seeks to identify the *true* hypothesis from among a known set of hypotheses. The learner is given a set of actions and knows the random distribution of the outcome of any action under any true hypothesis. Given a target error $\delta > 0$, the goal is to sequentially select the fewest number of actions so as to identify the true hypothesis with probability at least $1 - \delta$. Motivated by applications in which the number of hypotheses or actions is massive (e.g., genomics-based cancer detection), we propose efficient (greedy, in fact) algorithms and provide the first approximation guarantees for ASHT, under two types of adaptivity. Both of our guarantees are independent of the number of actions and logarithmic in the number of hypotheses. We numerically evaluate the performance of our algorithms using both synthetic and real-world DNA mutation data, demonstrating that our algorithms outperform previously proposed heuristic policies by large margins.

Key words: Active Learning, Sequential Hypothesis Testing, Approximation Algorithms, Cancer Detection

1. Introduction

Among the most important open problems in cancer research today is the development of an effective approach for the *detection* of cancer, particularly at its *earliest* stages. Indeed for quite some time now, there has been vast, uncontroverted evidence that early detection substantially enhances the possibility of successful treatment (Etzioni et al. 2003, Cuzick et al. 2014, Jerant et al. 2000). As a few examples, the five-year survival rates after early diagnosis (and treatment) of breast, ovarian, and lung cancers are 90%, 90%, and 70%, respectively, compared to 15%, 5%, and 10% for patients diagnosed at the latest stages (Siegel et al. 2015, Jemal et al. 2010, Ferguson et al. 2000). In short, early detection is a silver bullet.

Unfortunately, although monitoring certain "warning signs" occasionally yields early diagnoses, cancer screening is in general notoriously difficult, and existing approaches fall short. Modern cancer diagnoses are for the most part made via *biopsies*, i.e., the surgical removal

* Equal contribution.

of tissue for testing, and while biopsies are extremely accurate (with respect to identifying cancer in the removed tissue itself), they are too invasive and expensive to be used as a general screening procedure.[1] But even beyond issues like cost and inconvenience to the patient, the use of biopsies for *early* cancer screening is in fact fundamentally impossible for several cancer types, such as lung and pancreatic cancers, which almost never show symptoms until after cancer cells have *metastasized*[2] (Miller et al. 1981, Paez et al. 2004, O'Rourke and Edwards 2000).

Because of these difficulties, there has always existed a dream within the medical community of developing a *liquid biopsy*, i.e., a blood test for cancer. This test would naturally be minimally invasive, and ideally would have the same accuracy as a traditional biopsy. Most importantly, the liquid biopsy would detect cancers at their earliest stages. What is particularly exciting today is that liquid biopsies are no longer just a pipe dream – these tests have been under rapid development over the last few years, largely fueled by advances in technology for collecting data (next-generation DNA sequencing, in particular), and increasing computational and algorithmic power for analyzing this new data. Development of these tests is being undertaken by major academic research labs (Bettegowda et al. 2014, Manterola et al. 2014, Best et al. 2015, Kim et al. 2016, Banerjee et al. 2016, Razavi et al. 2017, Chan et al. 2017, Cohen et al. 2018, Liu et al. 2020) along with a handful of biotechology startups (e.g., Grail, Guardant, Freenome).

1.1. The Genomic Approach and the Value of Adaptivity

Ultimately, this paper addresses a set of *active learning* problems that occur in the development of liquid biopsies. To understand how such problems arise, it may be useful to review, at a high level, the underlying biology here (this subsection can be skipped without loss of continuity). Starting with a basic fact: cancer is caused by mutations in DNA, meaning the DNA within every tumor cell has a *set* of mutations that identifies the cell as tumorous, along with its location in the body (and thus the type of cancer).[3] These mutations are the "signals" that *genomic* liquid biopsies are designed to detect. The reason that these signals

[1] Less invasive screening tools do exist, but by and large none has achieved the requisite accuracy to be adopted by the medical community for general screening.

[2] *Metastasis* refers to the formation of a new cancer "colony" at a separate location in the body. The occurrence of this process is generally used to define the line between early and late stage cancers, as once a primary tumor metastasizes, successful treatment using established therapeutics becomes nearly hopeless.

[3] Identifying the specific mutations which *cause* a particular cancer to form is still an open problem. Fortunately for the purposes of a liquid biopsy, correlation suffices.

are detectable from blood is due to *cell-free DNA*—the DNA of any dying cell is occasionally released into the bloodstream (rather than being destroyed), and so an individual's blood at any moment contains free floating DNA that we can view as having been randomly "sampled" (in a probabilistic sense) from throughout the body.

Thus comes the main idea: if an individual has a tumor, some portion of their cell-free DNA will contain mutations which signal the existence of that tumor. So performing the liquid biopsy simply involves extracting cell-free DNA (a relatively easy task), and sequencing it in search of these mutations. There is no purely biological reason why this approach should fail. Instead, the constraint that we face today is the *cost*—human DNA consists of three billion *addresses*, but the cost of DNA sequencing means that any reasonably-priced test can only include approximately 10^4 of those addresses.[4] So the challenge is to design a liquid biopsy using just a *panel* of 10^4 pre-identified addresses.

To date, one of the two most successful prototypes of a liquid biopsy is from Cohen et al. (2018).[5] To select the panel of DNA addresses for their liquid biopsy, they used the publicly-available dataset—*Catalogue of Somatic Mutations in Cancer* (COSMIC, Tate et al. 2019) which contains complete DNA sequences from thousands of tumor cells, and this allows one to "simulate" the accuracy of different combinations of addresses subject to any budget constraint and select the most accurate one.[6] Cohen et al. (2018) did exactly this analysis, which is reproduced here as the grey-colored curves in Figure 1. Each of those curves shows how the proportion of detected cancer patients increases with the number of addresses, for eight different cancer types.

Now the problem of identifying the subset of addresses, subject to cardinality constraint, that maximizes the number of cancer patients detected (within the COSMIC dataset) is a well-defined optimization problem (a maximum coverage problem, in fact), and while this application can become quite large in practice (technically, each of the three billion DNA

[4] The back-of-the-envelope calculation works as follows: modern DNA sequencing costs approximately $\$10^{-6}$ (USD) per address. But because tumorous DNA only makes up a tiny proportion (about one in ten-thousand) of a cancer patient's cell-free DNA, each address used in a liquid biopsy needs to be sequenced 10^4 times (to avoid false negatives). Finally, a screening test should cost at most $\$10^2$, so $\$10^2$ / ($\$10^{-2}$ per address) $= 10^4$ addresses. In the adaptive problem that we will introduce later, we further relax the assumption that each address needs to be sequenced 10^4 times. Instead, the number of repetition will be determined by our algorithms.

[5] The other is Liu et al. (2020).

[6] Readers with experience in machine learning might interpret this entire procedure as training a classifier, with COSMIC as the training set, and be concerned about over-fitting and generalization error. While we view this as orthogonal to the problem we seek to address, it is worth noting that Cohen et al. (2018) actually observed *higher* detection rates (i.e., accuracy) in practice than those predicted by COSMIC.

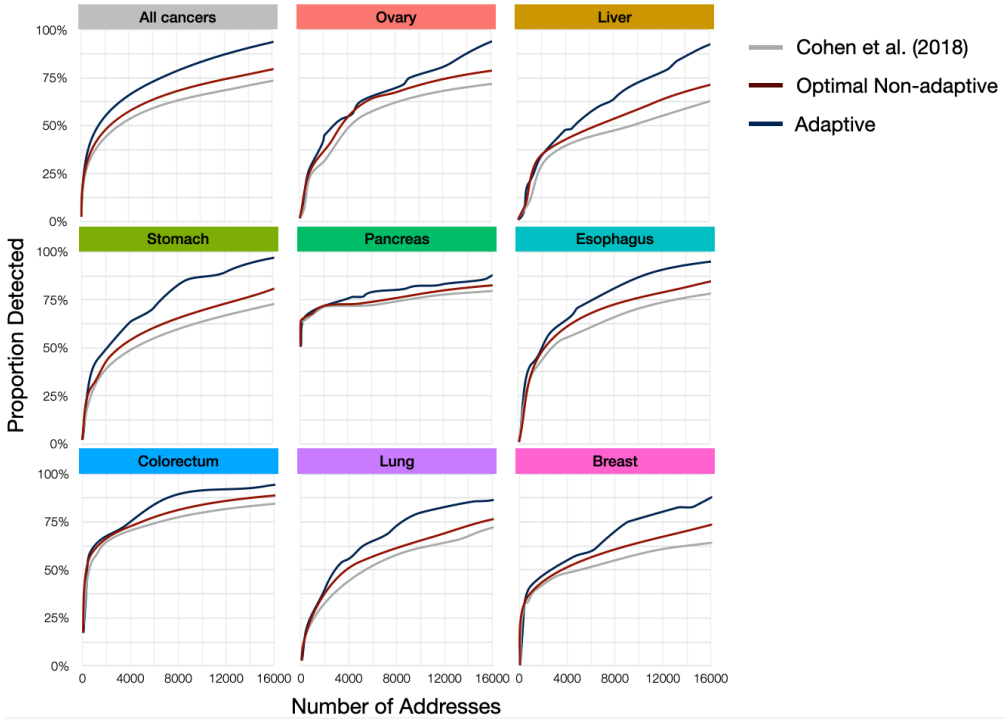


Figure 1 Comparison of (non-adaptive) genomic panels from Cohen et al. (2018) with *optimal* non-adaptive panels, and adaptive panels constructed using our greedy adaptive algorithm. For each approach, detection rate (on the COSMIC dataset) is plotted as a function of panel size. Results are reported for eight cancer types, in combination and individually. Figure format adapted from Cohen et al. (2018).

addresses is a "set," and there are tens of thousands of samples in COSMIC to "cover"), this particular instance can be solved to optimality with an off-the-shelf integer programming solver. That is precisely what we have done and represented as the red curves in Figure 1, and we can see that the red curves indeed dominate the grey curves of Cohen et al. (2018).

After a perhaps long-winded introduction, it is at *this* point that the problem we seek to address has finally appeared. Reviewing Figure 1 again, an unfortunate observation is that even the optimal panels (the red curves) are insufficient for a practical liquid biopsy—in visual terms, the curves do not reach far enough into the top-left area (representing low cost and high accuracy). Now advances in DNA sequencing technology may eventually solve this problem (by further reducing sequencing costs), but the purpose of this paper is to study a more immediate solution: *adaptive testing*. We use the term "adaptive" to mean that the test for an individual can operate over multiple rounds, with the choice of addresses in each round being made using the results of prior rounds (the tests used by Cohen et al. (2018), along with our "optimal" panels, were non-adaptive). The problem of identifying the optimal adaptive test can similarly be formalized (as we will do), though that problem almost

certainly does not admit a solution via computational brute force. Instead, we will analyze fast approximation algorithms, whose practical value is partly demonstrated by the blue curves in Figure 1.

1.2. The Problem and Our Contributions

At this point, it is worth abstracting away the application, because the natural model for this is a well-studied one. Consider the problem of learning the *true* hypothesis from among a (potentially large) set of candidate hypotheses H. Assume that the learner is given a (potentially large) set of actions A, and knows the distribution of the noisy outcome of each action, under each potential hypothesis. In the context of liquid biopsies, the candidate hypotheses are different types of cancers, and the actions correspond to sequencing individual DNA addresses (actually combinations of addresses, as we will discuss later on). The learner incurs a fixed cost each time an action is selected, and seeks to identify the true hypothesis with sufficient confidence, at minimum total cost. Finally, and most importantly, the learner is allowed to select actions *adaptively*.

This well-studied problem is referred to as *active sequential hypothesis testing*, and as we will describe momentarily, there exists a broad set of results that tightly characterizes the optimal achievable cost under various notions of adaptivity. Unfortunately, the corresponding optimal policies are typically only characterized as the optimal policy to a Markov decision process (MDP)—thus, they remain computationally hard to compute when one requires a policy in practice. This deficiency becomes particularly apparent in modern applications where both the set of hypotheses and set of actions may be large: our own application has tens of hypotheses and *billions* of tests at full scale. Thus motivated, *our primary contribution is the first approximation algorithms for ASHT*.

We study ASHT under two notions of adaptivity: *partial* and *full*, where partial adaptivity requires the sequence of actions to be decided upfront (with adaptively chosen stopping time), and full adaptivity allows the choice of action to depend on previous outcomes. For both problems, we propose *greedy* algorithms that run in $O(|A||H|)$ time, and prove that their expected costs are upper bounded by a non-trivial multiplicative factor of the corresponding optimal costs. Most notably, these approximation guarantees are *independent* of $|A|$ (contrast this with the trivially-achievable guarantee of $O(|A|)$) and *logarithmic* in $|H|$ (the optimal cost itself is often $\Omega(|H|)$).

	Noise	Approx. Ratio	Objective	Adaptivity Type
Kosaraju et al. (1999)	No	Yes	Both	Full
Chakaravarthy et al. (2009)	No	Yes	Both	Full
Nowak (2009)	Yes	No	Worst-case	Full
Naghshvar and Javidi (2013)	Yes	No	Average	Both
Im et al. (2016)	No	Yes	Both	Partial
Jia et al. (2019)	Semi*	No	Both	Both
This Work	Yes	Yes	Both	Both

Table 1 **Summary of related work. **Semi*** **refers to a restrictive special case. Approx. stands for approximation.**

Our theoretical results rely on drawing connections to two existing problems: *submodular function ranking* (SFR, see Azar and Gamzu 2011) and the *optimal decision tree* (ODT) problem (Laurent and Rivest 1976). These connections allow us to tackle what is arguably the primary challenge in achieving approximation results for ASHT, which is its inherent *combinatorial* nature. We will argue that existing heuristics from statistical learning fail precisely because they disregard this combinatorial difficulty—indeed, they largely amount to solving the completely *non-adaptive* version of the problem. At the same time, existing results for SFR and ODT fail to account for *noise* in a manner that would map directly to ASHT—this extension is among our contributions.

Finally, we performed a set of large-scale experiments, including the ones that were built on the same setup of Cohen et al. (2018). These experiments demonstrate that, in both the partial and fully adaptive settings, our greedy algorithms (a) scale to the size of real-world problems, and (b) outperform existing benchmarks for ASHT in practice. In the setting of liquid biopsies, our results suggest that adaptive testing is sufficient to the achieve the remaining accuracy needed to bring about a practical cancer screening test.

1.3. Related Work

Our work is closely related to three streams of research. Table 1 highlights the key differences between our contributions and those of the most relevant previous works.

Hypothesis Testing and Asymptotic Performance In the classical binary sequential hypothesis testing problem, a decision maker is provided with one action whose outcome is stochastic (Wald 1945, Armitage 1950, Lorden 1977), and the goal is to use the minimum

expected number of samples to identify the true hypothesis subject to some given error probability. The ASHT problem, first studied in Chernoff (1959), generalizes this problem to multiple actions. Most related to our work is Naghshvar and Javidi (2013), who formulated a similar problem as an MDP. We will postpone describing and contrasting their work until the experiments section.

Active Learning and Sample Complexity In active learning, the learner is given access to a pool of unlabeled samples (cheaply obtainable) and is allowed to request the label of any sample (expensive) from that pool. The goal is to learn an accurate classifier while requesting as few labels as possible. Some nice surveys include Hanneke et al. (2014) and Settles (2009). Our model extends the classical discrete active learning model Dasgupta (2005) in which outcomes are noiseless (deterministic) for any pair of hypothesis and unlabeled sample. When outcomes are noisy, the majority of provable guarantees are provided via sample complexity using minimax analysis techniques. Castro and Nowak (2007) showed tight minimax classification error rates for a broad class of distributions. Other sample complexity results on noisy active learning include Wang and Singh (2016), Nowak (2009), Balcan et al. (2006), Awasthi et al. (2017), Hanneke and Yang (2015).

Approximation Algorithms for Decision Trees Nearly all optimal approximation algorithms for minimizing cover time are known in the noiseless setting (Kosaraju et al. 1999, Adler and Heeringa 2008, Arkin et al. 1998). When the outcome is stochastic, Golovin and Krause (2011) proposed a framework for analyzing algorithms in the active learning setting under the *adaptive submodularity* assumption, with the goal of maximizing the information gained with a prescribed budget. However, their assumption does not hold for many natural setups including ASHT. Chen et al. (2015) considered a variant using ideas from the submodular max-coverage problem without the adaptive submodularity assumption, and provided a constant factor approximation to the problem using ideas from the submodular max-coverage problem. Other works based on submodular function covering include Navidi et al. (2020), Guillory and Bilmes (2011), Krause et al. (2008). Jia et al. (2019) provided approximation ratios under the constraint that the algorithm may only terminate when it is completely confident about the outcome.

2. Model

In this section, we formally introduce the active sequential hypothesis testing problem; the mapping of this generic problem to the liquid biopsy application is described in detail in

Section 6. Let H be a finite set of *hypotheses*, among which exactly one is the (unknown) *true* hypothesis that we seek to identify. We denote this true hypothesis h^*. In this paper, we study the *Bayesian* setting, wherein h^* is drawn from a known prior distribution π over the entire hypothesis set H.

Let A be the set of available *actions*, and let $\mathcal{D}$ be a given family of parametrized distributions that encode the *outcome distributions*, i.e., the distributions from which the outcome is drawn when an action is chosen. For ease of exposition, we parameterize the family of distributions $\mathcal{D}$ by $\Theta \subseteq \mathbb{R}$, i.e., $\mathcal{D} = \{D_\theta\}_{\theta \in \Theta}$. Thus, selecting, or "playing", an action yields a random *outcome* drawn independently from a distribution within the given family $\mathcal{D} = \{D_\theta\}_{\theta \in \Theta}$. In addition, we are given a function μ that maps each pair of action and hypothesis to some $\theta \in \Theta$, i.e., $\mu : H \times A \to \Theta$, such that if $h \in H$ is the true underlying hypothesis and we select action $a \in A$ to play, then we observe a random outcome that is drawn independently from distribution $D_{\mu(h,a)}$. Note that in this *noisy* setting, an action can (and often should) be played multiple times (in the same way that a DNA address should be sequenced multiple times since each strand of cell-free DNA is effectively sampled randomly from throughout the body).

An *instance* of the active sequential hypothesis testing problem is then fully specified by a tuple: $(H, A, \pi, \mu, \mathcal{D})$. The goal is to sequentially select actions to identify the true hypothesis with "sufficiently high" confidence, at minimal expected cost, where cost is measured as the number of actions, and the expectation is with respect to the Bayesian prior and the random outcomes. The notion of *sufficiently high* confidence is encoded by a parameter $\delta \in (0, 1)$, and requires that under any true hypothesis $h \in H$, the probability of erroneously identifying a different hypothesis is at most δ. An algorithm is said to have achieved δ-**PAC-error** if it identifies the true hypothesis with this notion of sufficiently high confidence.

In this paper, we focus on two important families of the outcome distributions, D_θ's: the Bernoulli distribution $\mathrm{Ber}(\theta)$, and the Gaussian distribution $N(\theta, \sigma^2)$. In the latter, the variance σ^2 is a known constant (with respect to θ).[7] By re-scaling, without loss of generality we may assume $\sigma^2 = 1$. To state our guarantees, we require two additional assumptions. The first assumption is needed for relating the sub-gaussian norm to the KL-divergence, in the partially adaptive setting. It ensures that the parameterization Θ is a meaningful one, in the

[7] Sub-Gaussianity with similar control over the sub-Gaussian norm would suffice.

sense that if two parameters $\theta, \theta' \in \Theta$ are far apart, then the corresponding distributions D_θ and $D_{\theta'}$ are also "far" apart, (i.e., their KL divergence is "far" apart). Note that Assumption 1 is satisfied for the Bernoulli distribution $\text{Ber}(\theta)$ when $\theta \in [\theta_{\min}, \theta_{\max}]$ for some constants $0 < \theta_{\min} < \theta_{\max} < 1$, and for the Gaussian distribution $N(\theta, 1)$ where θ lies in some bounded subset of $\mathbb{R}$.

Assumption 1 *There exist two constants $C_1, C_2 > 0$ such that for any $\theta, \theta' \in \Theta$, we have*

$$C_1 \cdot \text{KL}(D_\theta, D_{\theta'}) \leq (\theta - \theta')^2 \leq C_2 \cdot \text{KL}(D_\theta, D_{\theta'}),$$

where $\text{KL}(\cdot, \cdot)$ is the Kullback-Leibler divergence.

Our second major assumption simply ensures the existence of a valid algorithm by ensuring that every hypothesis is distinguishable via some action:

Assumption 2 (Validity) *For any pair of distinct hypotheses $g, h \in H$, there exists an action $a \in A$ with $\mu(g, a) \neq \mu(h, a)$.*

Note that in Assumption 2, we do not preclude the possibility that for a given action a, there exist (potentially many) pairs of hypotheses g and h such that the outcome distributions are the same, i.e., $\mu(g, a) = \mu(h, a)$. In fact, eliminating such possibilities would effectively wash out any meaningful combinatorial dimension to this problem. On the other hand, any approximation guarantee should be parameterized by some notion of *separation* (when it exists). For any two hypotheses $g, h \in H$ and any action $a \in A$, we define the distance between these two hypotheses under action a as $d(g, h; a) := \text{KL}\left(D_{\mu(g,a)}, D_{\mu(h,a)}\right)$. Let $s > 0$ be some positive constant. The following is the notion of separation, *s-separability*, that we use throughout the paper:

Definition 1 (s-separated instance) *An ASHT instance is said to be* **s-separated**, *if for any $a \in A$ and $g, h \in H$, $d(g, h; a)$ is either 0 or at least s.*

Note that in real-world applications, the parameters s could be arbitrarily small, and we introduce the notation of s-separability for the sake of proofs. We will show in Section 6 how our algorithms can easily be modified to handle small s values. In this paper, we will study two classes of algorithms that differ in the extent to which adaptivity is allowed.

Definition 2 *A* **fully adaptive** *algorithm is a decision tree,*[8] *each of whose interior nodes is labeled with some action, and each of whose edges corresponds to an outcome. Each leaf is labeled with a hypothesis, corresponding to the output when the algorithm terminates.*

Definition 3 *A* **partially adaptive** *algorithm* (σ, T) *is specified by a fixed sequence of actions* $\sigma = (\sigma_1, \sigma_2, ...)$*, with each* $\sigma_i \in A$*, and a stopping time* T*, such that the event* $\{T = t\}$ *is independent of the outcomes of actions* $\sigma_{t+1}, \sigma_{t+2}, \ldots$*, under any true hypothesis* $h^* \in H$ *and for any* $t \geq 1$*. (At the stopping time, the choice of which hypothesis to identify is trivial in our Bayesian setting—it is simply the one with the highest "posterior" probability.)*

Note that a partially adaptive algorithm can be viewed as a special type of fully adaptive algorithm: it is a decision tree with the additional restriction that the actions at each depth are the same. Therefore, a fully adaptive algorithm may be far cheaper than any partially adaptive algorithm. However, there are many scenarios (e.g., content recommendation and web search (Azar et al. 2009)) where it is desirable to fix the sequence of actions in advance. Furthermore, in many problems the theoretical analysis of partially adaptive algorithms turns out to be challenging (e.g., Kamath and Tzamos 2019, Chawla et al. 2019).

Thus, given an ASHT instance, there are two problems that we will consider, depending on whether the algorithms are partially or fully adaptive. In both cases, our goal is to design fast approximation algorithms—ones that are computable in polynomial[9] time and that are guaranteed to incur expected costs at most within a multiplicative factor of the optimum. In the coming sections, we will describe our algorithms and approximation guarantees. Before moving on to this, it is worth noting that our problem setup is extremely generic and captures a number of well-known problems related to decision-making for learning including best-arm identification for multi-armed bandits (Bubeck et al. 2009, Even-Dar et al. 2002, Mannor and Tsitsiklis 2004), group testing (Du et al. 2000), and causal inference (Gan et al. 2020), just to name a few.

3. Our Approximation Guarantees

In this section, we will state our approximation guarantees. We will define the corresponding greedy algorithms in the next two sections. Let $\mathrm{OPT}_\delta^{\mathrm{PA}}$ and $\mathrm{OPT}_\delta^{\mathrm{FA}}$ denote the minimal

[8] By approximating D_θ's with discrete distributions, we may assume each node has a finite number of children.

[9] Throughout this paper, *polynomial time* refers to polynomial in $(|H|, |A|, s^{-1}, \delta^{-1})$

expected cost of any partially adaptive and fully adaptive algorithm that achieves δ-PAC-error, respectively. Theorem 1 summarizes the approximation guarantee for our greedy partially adaptive algorithm:

Theorem 1 *Given an s-separated instance and any $\delta \in (0, 1/2)$, there exists a polynomial-time partially adaptive algorithm that achieves δ-PAC-error with expected cost*

$$O\left(s^{-1}\left(1+\log_{1/\delta}|H|\right)\log\left(s^{-1}|H|\log\delta^{-1}\right)\right)\mathrm{OPT}_{\delta}^{\mathrm{PA}}.$$

To help parse this result, if δ is on the order of $|H|^{-c}$ for some constant c, then the approximation factor becomes $s^{-1}(\log s^{-1} + \log|H| + c\log\log|H|)$. Theorem 2 summarizes the approximation guarantee for our greedy fully adaptive algorithm:

Theorem 2 *Given an s-separated instance and any $\delta \in (0, 1/2)$, there exists a polynomial-time fully adaptive algorithm that achieves δ-PAC-error with expected cost*

$$O\left(s^{-1}\log\left(|H|\delta^{-1}\right)\log|H|\right)\mathrm{OPT}_{\delta}^{\mathrm{FA}}.$$

A few observations might clarify the significance of these approximation guarantees:

1. Dependence on action space: Both guarantees are independent of the number of actions $|A|$. This is extremely important since, as described in the Introduction, there exist many applications where the the action space is massive. Moreover, since an approximation factor of $O(|A|)$ is always trivially achievable (by cycling through the actions), instances where $|A|$ is large are arguably the most interesting problems.
2. Dependence on $|H|, \delta$ and s: For fixed s and δ, these are the first polylog-approximations for both partially and fully adaptive versions. Further, for the partially adaptive version, the dependence of the approximation factor on δ is $O(\log\log\delta^{-1})$ when δ^{-1} is polynomial in $|H|$, improving upon the naive dependence $O(\log\delta^{-1})$. This is crucial since δ is often needed to be tiny in practice.
3. Greedy runtime: While we have only stated in our formal results that our approximation algorithms can be computed in $\mathrm{poly}(|A|, |H|)$ time, the actual time is more attractive: $O(|A||H|)$ for selecting each action. In contrast, the heuristic that we will compare against in the experiments requires solving multiple $\Omega(|A||H|^2)$-sized linear programs.

Despite their similar appearances, Theorems 1 and 2 rely on fundamentally different algorithmic techniques and thus require different analyses. In Section 4, we propose an algorithm inspired by the *submodular function ranking* problem, which greedily chooses a sequence of actions according to a carefully chosen "greedy score." We then sketch the proof of Theorem 1. In Section 5, we introduce our fully adaptive algorithm and sketch the proof of Theorem 2.

Finally, by proving a structural lemma (in Appendix D), we extend the above results to a special case of the **total-error** version (i.e., averaging the error over the prior π) where the prior distribution is uniform. With *δ-total-error* formally defined in Appendix D:

Theorem 3 *Given an s-separated instance with uniform prior π and any $\delta \in (0, 1/4)$, for both the partially and fully adaptive versions, there exist polynomial-time δ-total-error algorithms with expected cost $O\left(s^{-1}\left(1+|H|\delta^2\right)\log\left(|H|\delta^{-1}\right)\log|H|\right)$ times the optimum.*

4. Partially Adaptive Algorithm

This section describes our algorithm and guarantee for the partially adaptive problem. We first review necessary background from a related problem, and then state our algorithm (Algorithm 1). Finally, we sketch the proof of the following more general version of Theorem 1 (complete proof in Appendix B):

Proposition 1 *Let $\delta \in (0, \frac{1}{4}]$ and consider finding the optimal δ-PAC error algorithm. Given any boosting intensity $\alpha \geq 1$ and coverage saturation threshold $B \in (0, \frac{1}{2}\log\delta^{-1}]$, $\mathrm{RnB}(B, \alpha)$ (as defined in Algorithm 1) produces a partially adaptive algorithm with error $|H|\exp\left(-\Omega\left(\alpha B\right)\right)$ and expected cost $O\left(\alpha s^{-1}\log\left(|H|Bs^{-1}\right)\right)\mathrm{OPT}_{\delta}^{\mathrm{PA}}$.*

By setting $\alpha = 1 + \log_{\delta^{-1}}|H|$ and $B = \frac{1}{2}\log\delta^{-1}$, we immediately obtain Theorem 1.

4.1. Background: Submodular Function Ranking

In the SFR problem, we are given a ground set U of N *elements*, a family $\mathcal{F}$ of non-decreasing submodular functions $f : 2^U \to [0, 1]$ with $f(U)$ equaling 1 for every $f \in \mathcal{F}$, and a weight function $w : \mathcal{F} \to \mathbb{R}^+$. For any permutation $\sigma = (u_1, ..., u_N)$ of U, the *cover time* of f is defined as $\mathrm{CT}(f, \sigma) = \min\{t : f(\{u_1, ..., u_t\}) = 1\}$. The goal is to find a permutation σ of U with minimal weighted *cover time*, $\sum_{f\in\mathcal{F}} w(f)\cdot\mathrm{CT}(f, \sigma)$.

A greedy algorithm was proposed in Azar and Gamzu (2011), and we will use this algorithm as an important subroutine in our algorithm. This greedy algorithm constructs a sequence iteratively. At each iteration, we say a function is *covered* if its value on the set of the elements

selected so far is 1, and the function is *uncovered* otherwise. The sequence is initialized to be empty. At each iteration, let S denote the set of elements selected so far. The algorithm selects the element u with the maximal *coverage*, defined as

$$\mathrm{Cov}(u;S) := \sum_{f\in\mathcal{F}:f(S)<1} w(f)\cdot\frac{f(S\cup\{u\})-f(S)}{1-f(S)}.$$

Loosely, the algorithm chooses the element that maximizes the weighted sum of additional immediate proportional coverage. The following approximation guarantee for this algorithm is known to be the best possible among all polynomial-time algorithms:

Theorem 4 (Im et al. 2016) *For any SFR instance, the greedy algorithm described above returns a sequence whose cost is $O(\log \varepsilon^{-1})$ times the optimum, where*

$$\varepsilon := \min\left\{f(S\cup\{u\})-f(S)>0 : S\in 2^U, u\in U, f\in\mathcal{F}\right\}.$$

Challenge: To motivate our algorithm, consider first the following simple idea: "boost" each action, and hence reduce the problem to a deterministic problem P_{det}. Then show that the existing technique (submodular function ranking for partially adaptive and greedy analysis for ODT for fully-adaptive) returns a policy with cost $O(\log|H|)$ times the no-noise optimum, and finally show that this no-noise policy can be converted to a noisy version by losing anther factor of $O(s^{-1}\log(\delta^{-1}|H|))$. This analysis was in fact our first attempt. However, there are at least two issues that one runs into along this path:

1. This analysis only compares the policy's cost with the no-noise optimum, but our focus is the δ-noise optimum. In particular, the simpler analysis implicitly assumes that the δ-noise optimum is at least $\Omega(s^{-1}\log(\delta^{-1}|H|))$ times the no-noise optimum, which is not necessarily true. Moreover, it is challenging to analyze the gap between the no-noise optimum and the δ-noise optimum.
2. The guarantee that results from this simple analysis is *weaker* than ours in terms of δ: it yields a factor of $\log(1/\delta)$, as opposed to the $\log\log(1/\delta)$ in our analysis. This distinction is nontrivial, particularly in applications where the error is required to be exponentially small in $|H|$.

4.2. Rank and Boost (RnB)

Our RnB algorithm (Algorithm 1) circumvents the issues above by drawing a connection between ASHT and SFR. First, we observe that although an action is allowed to be selected multiple times, we may assume each action is selected for at most $M = M(\delta, s, |H|) = O(s^{-1}|H|^2 \log(|H|/\delta))$ times. In fact,

Observation 1 *Let $\widetilde{A}$ be the (multi)-set obtained by creating M copies of each $a \in A$. Then there exists a sequence σ of $|\widetilde{A}|$ actions, s.t. for any true hypothesis $h^* \in H$, h^* has the highest posterior with probability $1-\delta$ after performing all actions in σ.*

Thus, given $\widetilde{A}$, we define $f_h^B : 2^{\widetilde{A}} \to [0,1]$ for any coverage saturation level $B > 0$ and $h \in H$ as $f_h^B(S) = \frac{1}{|H|-1} \sum_{g \in H \setminus \{h\}} \min\{1, \frac{1}{B} \sum_{a \in S} d(g,h;a)\}$. One can verify that f_h^B is monotone and submodular. Our algorithm computes a nearly optimal sequence of actions using the greedy algorithm for SFR, and creates a number of copies for each of them. Then we assign a *timestamp* to each $h \in H$, and scan them one by one, terminating when the likelihood of one hypothesis is dominantly high.

Although a naive implementation of Algorithm 1 yields a running time that is linear in the number of actions, however since $\text{Score}(a; S)$ (Line 6 of Algorithm 1) can be calculated independently for each action a, one could paralyze this calculation for different actions and thus reducing the dependency on $|A|$. The same observation also holds for the rest algorithms to be introduced in the paper.

4.3. Proof Sketch for Proposition 1

We sketch a proof here and defer the details to Appendix B. The error analysis follows from standard concentration bounds, so we focus on the cost analysis. Suppose $\alpha > 0$, $\delta \in (0, 1/4]$, and $B \in (0, (1/2)\log \delta^{-1}]$. Let (σ^*, T^*) be any optimal partially adaptive algorithm, and let (σ, T) be the policy returned by RnB. Our analysis consists of the following steps:

(A) The sequence σ does well in covering the submodular functions, in terms of the total cover time: $\sum_{h \in H} \pi(h) \cdot \text{CT}(f_h^B, \sigma) \le O\left(\log\left(|H| B s^{-1}\right)\right) \sum_{h \in H} \pi(h) \cdot \text{CT}(f_h^B, \sigma^*)$.

(B) The expected stopping time of our algorithm is not too much higher than the cover time of its submodular function: $\mathbb{E}_h[T] \le \alpha \cdot \text{CT}(f_h^B, \sigma)$, $\forall h \in H$.

(C) The expected stopping time in (σ^*, T^*) can be lower bounded in terms of the total cover time: $\mathbb{E}_h[T^*] \ge \Omega(s) \cdot \text{CT}(f_h^B, \sigma^*)$, $\forall h \in H$.

Algorithm 1 Partially Adaptive Algorithm: RnB(B, α)

Parameters: Coverage saturation level $B > 0$ and boosting intensity $\alpha > 0$.
Input: ASHT instance $(H, A, \pi, \mu, \mathcal{D})$
Initialize: $\sigma \leftarrow \emptyset, \tilde{\sigma} \leftarrow \emptyset$ ▷ Store the selected of actions.
for $t = 1, 2, ..., |\widetilde{A}|$ **do** ▷ **Rank:** Compute a sequence of actions.
  $S \leftarrow \{\sigma(1), ..., \sigma(t-1)\}$. ▷ Actions selected so far.
  for $a \in \widetilde{A}$ **do** ▷ Compute scores for each action.

$$\text{Score}(a; S) \leftarrow \sum_{h: f_h^B(S) < 1} \pi(h) \frac{f_h^B(S \cup \{a\}) - f_h^B(S)}{1 - f_h^B(S)}.$$

    $\sigma(t) \leftarrow \arg\max\{\text{Score}(a; S) : a \in \widetilde{A} \backslash S\}$. ▷ Select the greediest action.
for $t = 1, 2, ..., |\tilde{A}|$: **do** ▷ **Boost:** Repeat each action in σ for α times.
  for $i = 1, 2, ..., \alpha$: **do**
    $\tilde{\sigma}\big(\alpha(t-1) + i\big) \leftarrow \sigma(t)$.
for $t = 1, ..., \alpha|\tilde{A}|$: **do**
  Select action $\tilde{\sigma}(t)$ and observe outcome y_t.
  if $t = \alpha \cdot \text{CT}(f_h^B, \sigma)$ for some $h \in H$: **then** ▷ If t is the *timestamp* for some h.
    for $g \in H \backslash \{h\}$: **do**
      $\Lambda(h, g) \leftarrow \prod_{i=1}^{t} \mathbb{P}_{h,\tilde{\sigma}(i)}(y_i) / \mathbb{P}_{g,\tilde{\sigma}(i)}(y_i)$. ▷ Compute the likelihood ratio.
    if $\log \Lambda(h, g) \geq \alpha B/2$ for all $g \in H \backslash \{h\}$, **then**
      return h. ▷ Hypothesis identified.

Proposition 1 follows by combining the above three steps. In fact,

$$\begin{aligned}
\sum_{h \in H} \pi(h) \cdot \mathbb{E}_h[T] &\leq \alpha \sum_{h \in H} \pi(h) \cdot \text{CT}(f_h^B, \sigma) \\
&\leq O\left(\alpha \log \frac{|H|B}{s}\right) \sum_{h \in H} \pi(h) \cdot \text{CT}(f_h^B, \sigma^*) \\
&\leq O\left(\frac{\alpha}{s} \log \frac{|H|B}{s}\right) \sum_{h \in H} \pi(h) \cdot \mathbb{E}_h[T^*],
\end{aligned}$$

where $\sum_h \pi(h) \cdot \mathbb{E}_h[T]$ is the expected cost of our algorithm, and $\sum_h \pi(h) \cdot \mathbb{E}_h[T^*]$ is the expected cost of the optimal partially adaptive algorithm, $\text{OPT}_\delta^{\text{PA}}$.

At a high level, Step A can be showed by applying Theorem 4 and observing that the marginal positive increment of each f_h^B is $\Omega(s/(|H|B))$. Step B is implied by the correctness of the algorithm. In our key step, Step C, we fix an arbitrary δ-PAC-error partially adaptive algorithm (σ, T) and a hypothesis $h \in H$. Denote CT_h the cover time of f_h^B under permutation σ, with B chosen to be $\frac{1}{2}\log\delta^{-1}$, i.e., $\mathrm{CT}_h = CT(f_h^B, \sigma)$. Our goal is to lower bound $\mathbb{E}_h[T]$ in terms of CT_h. Given any $d_1, ..., d_n$, denote $d^i = \sum_{j=1}^{i} d_j$. To this aim, we first show that for suitable choices of d_i's and t, the solution $z_i = \mathbb{P}_h[T = i]$ is feasible to the following LP:

$$\begin{aligned} LP(d,t): \quad \min_{z} & \sum_{i=1}^{N} i \cdot z_i \\ s.t. & \sum_{i=1}^{N} d^i z_i \geq \sum_{i=1}^{\mathrm{CT}_h - 1} d_i, \\ & \sum_{i=1}^{N} z_i = 1, \\ & z \geq 0. \end{aligned}$$

A feasible solution z can be viewed as a distribution of the stopping time. When $d_i = d(g, h; a_i)$, the first constraint says that the total KL-divergence "collected" at the stopping time has to reach a certain threshold. We show that $z_i = \mathbb{P}_h[T = i]$ is feasible, and the objective value of z is exactly $\mathbb{E}_h[T]$, hence $\mathbb{E}_h[T]$ is upper bounded by the LP-optimum $LP^*(d, t)$. Finally, we lower bound $LP^*(d, \mathrm{CT}_h - 1)$ by $\Omega(s \cdot \mathrm{CT}_h)$. The complete proof of Step C could be found in Appendix B.1.

5. Fully Adaptive Algorithm

In this section, we introduce our greedy fully adaptive algorithm. For ease of presentation, we only consider the scenario where the prior π is uniform over all hypotheses in this work. However, note that our guarantees hold for general priors. Our analysis is based on a reduction to the classical ODT problem.

5.1. Background: Optimal Decision Trees

In the ODT problem, an *unknown* true hypothesis h^* is drawn from a set of hypotheses H with some known probability distribution π. There is a set of known *tests*, each being a (deterministic) mapping from H to a finite *outcome space* set O. Thus, when performing a test, we can *rule out* the hypotheses that are inconsistent with the observed outcome, hence reducing the number of *alive* hypotheses. Moreover, the cost $c(T)$ of each test T is known,

Algorithm 2 Fully Adaptive Algorithm

Input: ASHT instance $(H, A, \pi, \mu, \mathcal{D})$ and error $\delta \in (0, 1/2)$.
$H_{\text{alive}} \leftarrow H$. ▷ *Alive* hypotheses.
while $|H_{\text{alive}}| \geq 2$ **do**
  $\widehat{a} \leftarrow \arg\max_{a \in A} \left\{ \min_{\omega \in \Omega_a} |H_{\text{alive}} \backslash T_a^{\omega}| \right\}$. ▷ Greedy step.
  $c(\hat{a}) \leftarrow \lceil s(\hat{a})^{-1} \log(|H|/\delta) \rceil$. ▷ Num. of times to boost for sufficient confidence.
  Select $\hat{a}$ for $c(\hat{a})$ times consecutively and observe outcomes $X_1, ..., X_{c(\hat{a})}$.
  $\hat{\mu} \leftarrow \sum_{i=1}^{c(\hat{a})} X_i$. ▷ Mean outcome.
  $\hat{\omega} \leftarrow \arg\min\{|\hat{\mu} - \omega| : \omega \in \Omega_a\}$. ▷ Round $\hat{\mu}$ to the closest ω.
  $H_{\text{alive}} \leftarrow H_{\text{alive}} \cap T_{\hat{a}}^{\hat{\omega}}$. ▷ Update the alive hypotheses.

and the *cost of a decision tree* is defined to be the expected total cost of the tests selected until one hypothesis remains *alive*, in which case we say the true hypothesis is *identified*. The goal is to find a valid decision tree with minimal expected cost.

Note that the ODT problem can be viewed as a special case of the fully adaptive version of our problem where there is no noise and δ is 0. Consider the following greedy algorithm: let A be the alive hypotheses. Define Score(T) for each test T to be the minimal (over all possible outcomes) number of alive hypotheses that it rules out in A. Then, we select the test T with the highest "bang-per-buck" Score(T)$/c(T)$. This algorithm is known to be an $O(\log |H|)$-approximation:

Theorem 5 (Chakaravarthy et al. 2009) *For any ODT instance with uniform prior, the above greedy algorithm returns a decision tree whose cost is $O(\log |H|)$ times the optimum.*

5.2. Our Algorithm

We will analyze our greedy algorithm by relating to the above result. Consider the following ODT instance $\mathcal{I}_{\text{ODT}}$ for any given ASHT instance $\mathcal{I}$. The hypotheses set and prior in $\mathcal{I}_{\text{ODT}}$ are the same as in $\mathcal{I}$. For each action $a \in A$, let $\Omega_a := \{\mu(h, a) | h \in H\}$ be the mean outcomes. By Chernoff bound, we can show that when h is the true hypothesis, with high probability the mean outcome is "close" to $\mu(h, a)$ when a is repeated for $c(a)$ times. This motivates us to define a test $T_a : H \to \Omega_a$ s.t. $T_a(h) = \mu(h, a)$, with cost $c(a) = \lceil s(a)^{-1} \log(|H|/\delta) \rceil$, where $s(a) = \min\{d(g, h; a) > 0 : g, h \in H\}$ is the separation parameter under action a. Such a test corresponds to selecting action a for $c(a)$ times consecutively in a row.

For each $\omega \in \Omega_a$, abusing the notation a bit, let $T_a^\omega \subseteq H$ denote the set of hypotheses whose outcome is ω when performing T_a, i.e., $T_a^\omega = \{h : \mu(h,a) = \omega\}$. At each step, Algorithm 2 selects an action $\hat{a}$ using the greedy rule (Step 4) and then repeat $\hat{a}$ for $c(\hat{a})$ times. Then we round the empirical mean of the observations to the closest element $\hat{\omega}$ in Ω_a, and rule out the hypotheses that are inconsistent with the observed outcome, i.e., the h's with $\mu(h,a) \neq \hat{\omega}$. We terminate when only one hypothesis remains alive.

5.3. Analysis

We sketch a proof for Theorem 2 and defer the details to Appendix C. Let h^* be the true hypothesis. By Hoeffding's inequality, in each iteration, with probability $1 - e^{-\log(|H|/\delta)} = 1 - \delta/|H|$ it holds $\hat{\omega} = \mu(h^*, \hat{a})$. Since in each iteration, $|H|$ decreases by at least 1, there are at most $|H| - 1$ iterations. Thus by union bound, the total error is at most δ.

Next we analyze the cost. Let GRE be the cost of Algorithm 2 and ODT^* be the optimum of the ODT instance $\mathcal{I}_{\mathrm{ODT}}$. For the sake of analysis, we consider a "fake" cost $c' := \lceil s^{-1} \log(|H|/\delta) \rceil$, which does not depend on a. The definition of the ODT instance I_{ODT} remains the same except that each test has **uniform** cost c' (as opposed to $c(a)$). Let $c(T)$ and $c'(T)$ be the costs of the greedy tree T returned by Algorithm 2 under c and c' respectively. Then by Theorem 5, $c'(T) \leq O(\log |H|) \cdot \mathrm{ODT}^*$. Note that $c' \leq c(a)$ for each a since the separation parameter s is no larger than $s(a)$ by definition. Hence,

$$\mathrm{GRE} \leq \mathrm{GRE} = c(T) \leq c'(T) \leq O(\log |H|) \cdot \mathrm{ODT}^*. \tag{1}$$

We relate ODT^* to OPT_δ^{FA} using the following result (see proof in Appendix C):

Proposition 2 $\mathrm{ODT}^* \leq O(s^{-1} \log(|H|/\delta)) \cdot \mathrm{OPT}_\delta^{FA}$.

The above is established by showing how to convert a δ-PAC-error fully adaptive algorithm to a valid decision tree, using only tests in $\{T_a\}$, and inflating the cost by a factor of $O(s^{-1} \log(|H|/\delta))$. Combining Proposition 2 with Equation (1), we obtain

$$GRE \leq O(s^{-1} \log \frac{|H|}{\delta} \log |H|) \cdot \mathrm{OPT}_\delta^{FA}.$$

Finally we remark that this analysis can easily be extended to general priors by reduction to the *adaptive submodular ranking* (ASR) problem (Navidi et al. 2020), which captures ODT as a special case. One may easily verify that the main theorem in Navidi et al. (2020)

implies that a (slightly different) greedy algorithm achieves $O(\log(|H|))$-approximation for the ODT problem with general prior, test costs, and an arbitrary number of branches in each test. Thus for general prior, the same analysis goes through if we first reduce ASHT to ASR, and then replace the greedy step (Step 4 in Algorithm 2) with the greedy criterion for ASR.

6. Experiments

We performed a large set of numerical experiments, on both synthetic and real-world data (extending the analysis on the cancer genomic data from Cohen et al. (2018) described in the Introduction). Our results demonstrate the following:

1. Our algorithms can be applied to the liquid biopsy application, potentially yielding a cost that is substantially lower than the existing commercial panels and those constructed by state-of-art benchmarks.
2. Unlike existing benchmarks, our algorithms can explicitly account for prior information, yielding superior performance under more realistic priors.
3. Although our theoretic guarantees depend on the separability parameters s (which was introduced by the boosting steps), with small modifications our algorithms perform well when s is small on both synthetic and real-world data, outperforming state-of-art benchmarks.

Our benchmarks include two algorithms (one partially adaptive and one fully adaptive) based on a polynomial-time policy proposed by Naghshvar and Javidi (2013) (*Policy 1*[10]) and a completely random policy. The rest of this section is organized as follows: we first describe the benchmark policies and the implementation of our own policies. Then in Section 6.2, we describe the setup and results of our synthetic experiments. Finally, in Section 6.3, we test the performance of our algorithms on a publicly-available dataset of genetic mutations for cancer—COSMIC (Tate et al. 2019).

6.1. Algorithm Details

In all algorithms, we start with a uniform prior, and update our prior distribution (over the hypotheses space) each time an observation is revealed. Unless otherwise mentioned, the algorithm terminates if the posterior probability of a hypothesis is above the threshold $1-\delta$. We first describe the random baseline and NJ's algorithms, and then discuss the modifications that we made to our algorithms.

[10] *Policy 2* in Naghshvar and Javidi (2013) does not have asymptotic guarantees and so is not considered in our experiments.

Random Baseline At each step, an action was uniformly chosen from the set of all actions.

NJ's Algorithms *NJ Adaptive* Naghshvar and Javidi (2013) is a two-phase algorithm that solves a relaxed version of our problem, where the objective is to minimize a weighted sum of the expected number of tests and the likelihood of identifying the wrong hypothesis, i.e., $\min \mathbb{E}(T) + Le$, where T is the termination time, L is the penalty for a wrong declaration, and e is the probability of making that wrong declaration. The problem was formulated as a Markov decision process whose state space is the posterior distribution over the hypotheses. In Phase 1, which lasts as long as the posterior probability of all hypotheses is below a carefully chosen threshold, the action is sampled according to a distribution that is selected to maximize the minimum expected KL divergence among all pairs of outcome variables. In Phase 2, when one of the hypotheses has posterior probability above the chosen threshold, r, the action is sampled according to a distribution selected to maximize the minimum expected KL divergence between the outcome of this hypothesis and the outcomes of all other hypotheses. This threshold was optimized over in both synthetic and real-world experiments. The algorithm stops if the posterior of a hypothesis is above the threshold $1 - L^{-1}$. *NJ Partially Adaptive* contains only the Phase 1 policy.

Partially Adaptive In our synthetic experiments, we implement Algorithm 1 described in Section 4 exactly, and set the boosting factor, α, to be 1. In our real-world experiments, we consider a variant of our algorithm. In particular, we consider the following modifications: 1) the amount of boosting is now a built-in feature of the algorithm, and 2) breaking ties according to some heuristic. Algorithm 3 formally describes our modified algorithm. To consider the amount of boosting as a built-in feature of the algorithm, we first generate a sequence of actions of length η for some large η value (with replacement) and then truncate the sequence to the minimum length to include all unique actions that have appeared in the sequence. When all actions in sequence σ has performed and we did not reach the target accuracy, then we repeat the entire sequence again. Our partially adaptive algorithm on COSMIC was generated by initializing η to be 800. Across all accuracy levels, the maximum truncated sequence length is 97.

Fully Adaptive We implement our algorithm described in Section 5, with the modifications that 1) the amount of boosting is considered as a tunable parameter, 2) a hypothesis is only considered to be ruled out when we are deciding which action to perform, 3) we do not boost if no action can further distinguish any hypotheses in the alive set, 4) we break ties

Algorithm 3 Partially Adaptive Algorithm in the COSMIC Experiment

Parameters: Coverage saturation level $B > 0$ and maximum sequence length $\eta > 0$.

Input: ASHT instance (H, A, π, μ)

Output: actions sequence σ

Initialize: $\sigma \leftarrow \emptyset$ ▷ Store the selected of actions

for $t = 1, 2, ..., \eta$ **do do** ▷ **Rank:** Compute a sequence of actions of length η

  $S \leftarrow \{\sigma(1), ..., \sigma(t-1)\}$. ▷ Actions selected so far

  for $a \in A$ **do** ▷ Compute scores for each action

$$\text{Score}(a; S) \leftarrow \sum_{h: f_h^B(S) < 1} \pi(h) \frac{f_h^B(S \cup \{a\}) - f_h^B(S)}{1 - f_h^B(S)}.$$

  $\sigma(t) \leftarrow \arg\max\{\text{Score}(a; S) : a \in A\}$. ▷ Select the greediest action and break ties according the heuristic described in Algorithm 4

Let i be the largest index for which the an action appears the first time in sequence σ, then we return the sequence $(\sigma(1), ..., \sigma(i))$.

according to some heuristic. In particular, Modification 1) addresses the issues that our fully adaptive algorithm in Section 5 over-boosts. Modification b) controls the error probability δ when we decrease the amount of boosting. Modification c) handles small s without increasing the boosting factor. We formally describe this modified algorithm below.

Similar to NJ's algorithm, we maintain a probability distribution, ρ, over the set of hypotheses to indicate the likelihood of each hypothesis being the true hypothesis h^*. A hypothesis is considered to be ruled out at each step if the probability of that hypothesis is below a threshold in ρ. Throughout our experiments, we set this threshold to be $\delta/|H|$. At each step, after an action is chosen with certain repetitions and observation(s) is (are) revealed, we update ρ according to the realizations that we observed. Thus, under this setup, a hypothesis that was considered to be ruled out in the previous steps (due to "bad luck") could potentially become alive again.

At each iteration, for each action $a \in A$ and $k \in \mathbf{N}$, we define $T_{a,k}$ to be the meta-test that repeats action a for k times consecutively, and we define its cost to be $c(T_{a,k}) = kc_a$. By Chernoff bound, with k i.i.d. samples, we may construct a confidence interval of width $\sim k^{-1/2}$. This motivates us to rule out the following hypotheses when $T_{a,k}$ is performed.

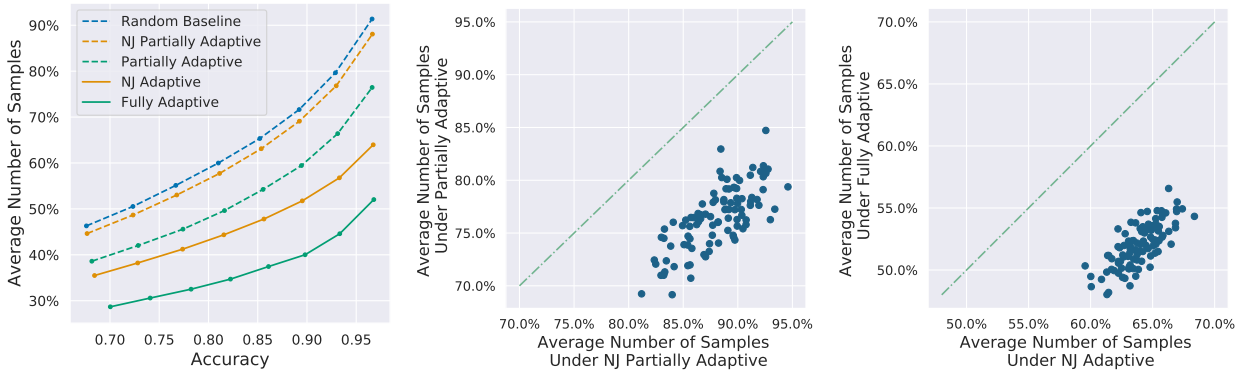


Figure 2 Comparison of our fully and partially adaptive algorithms with *NJ Adaptive*, *NJ Partially Adaptive* and *Random Baseline* on synthetic data. The average number of samples is normalized with respect to the largest number of sample required in *Random Baseline*. Left: each dot corresponds to the average performance of 100 randomly generated instances each averaged over 2,000 replications. Middle and Right: contains the same 100 instances in the left figure. Each dot corresponds to one instance and each averaged over 2,000 replications. Middle and Right: the average accuracies of those 100 instances in all algorithms equal to 0.97.

Let $\bar{\mu}$ be the observed mean outcome of these k samples, we define the elimination set to be $E_{\bar{\mu}}(T_{a,k}) := \{h : |\mu(h,a) - \bar{\mu}| \geq Ck^{-1/2}\}$, where C is set to be $1/2$ in our uniform prior experiment and set to be $1/3$ in our non-uniform prior experiment. To define greedy, we need to formalize the notion of bang-per-buck. Suppose H_{alive} is the current set of alive hypotheses. We define the score of a test as the number of alive hypotheses ruled out in the worst-case over all possible mean outcomes $\bar{\mu}$. Formally, the score of $T_{a,k}$ w.r.t mean outcome $\bar{\mu}$ is

$$\text{Score}_{\bar{\mu}}(T_{a,k}) = \text{Score}_{\bar{\mu}}(T_{a,k}; H_{alive}) = \frac{|E(T_{a,k}; \bar{\mu}) \cap H_{alive}|}{c(T_{a,k})},$$

and define its worst-case score to be $\text{Score}(T_{a,k}) = \min\{\text{Score}_{\bar{\mu}}(T_{a,k}) : \bar{\mu} \in \{0, 1/k, ..., 1\}\}$.

Our greedy policy simply selects the test T with the highest score, formally, select $T_{a,k} = \arg\max\{\text{Score}(T) : k \leq k_{\max}, a \in A\}$.

In the synthetic experiments, we set $k_{\max} = 5$. In the real-world experiments, we consider the cases where $k \in \{15, 20, 25, 30\}$ (with $k_{\max} = 30$) when the prior is uniform and $k \in \{5, 10, 15, 20\}$ (with $k_{\max} = 20$) when the prior is non-uniform. If several actions have the same greedy score, then we choose the action a^* whose sum of the KL divergence of pairs of $\mu(h, a^*)$ is the largest, and breaking ties arbitrarily. If no action can further distinguish any hypotheses in the alive set, then we set the boosting factor to be 1 and use the above heuristic to choose the action to perform. The algorithm is formally stated in Algorithm 4.

Algorithm 4 Adaptive experiments: FA($k_{\max}, \delta$)

Parameters: maximum boosting factor $k_{\max} > 0$ and convergence threshold $\delta > 0$

Input: ASHT instance (H, A, π, μ), current posterior of the true hypothesis vector ρ

Output: the test $T_{a,k}$ to perform in the next iteration

Let H_{alive} be the set of hypotheses i whose posterior probability ρ_i is above $\delta/|H|$.

for $k = 1, 2, ..., k_{\max}$ **do**

For each $a \in \widetilde{A}$ define:

$$\text{Score}_{\bar{\mu}}(T_{a,k}) = \text{Score}_{\bar{\mu}}(T_{a,k}; H_{\text{alive}}) = \frac{|E(T_{a,k}; \bar{\mu}) \cap H_{\text{alive}}|}{c(T_{a,k})},$$

where $E_{\bar{\mu}}(T_{a,k}) := \{h : |\mu(h,a) - \bar{\mu}| \geq Ck^{-1/2}\}$, and $c(T_{a,k}) = kc_a$. We define the worst-case score of a test to be:

$$\text{Score}(T_{a,k}) = \min\{\text{Score}_{\bar{\mu}}(T_{a,k}) : \bar{\mu} \in \{0, 1/k, ..., 1\}\}.$$

Compute greediest action

$$G = \arg\max\{\text{Score}(T) : k \leq k_{max}, a \in A\}.$$

if the Score of each test in G equals to 0, i.e, no test can further distinguish between the alive hypotheses under $k_{\max}$ **then**

we choose the action a^* such that $a^* = \arg\max \sum_{h,g \in H_{\text{alive}}} \text{KL}(\mu(h,a), \mu(g,a))$, breaking ties randomly, and return $k = 1$.

else

if G is a singleton, then we return G. Else, we choose the action a^* such that $a^* = \arg\max_G \sum_{h,g \in H_{\text{alive}}} \text{KL}(\mu(h,a), \mu(g,a))$, and breaking ties randomly.

6.2. Synthetic Experiments

Parameter Generation and Setup Figure 2 summarizes the results of our partially and fully adaptive experiments on synthetic data. Both figures were generated with 100 instances: each with 25 hypotheses and 40 actions. The outcome of each action under each hypothesis is binary, i.e., the $D_{\mu(h,a)}$'s are the Bernoulli distributions, where $\mu(a,h)$ were *uniformly* sampled from the [0,1] interval. Each instance was then averaged over 2,000 replications, where a "ground truth" hypothesis was randomly drawn. The prior distribution, π, was

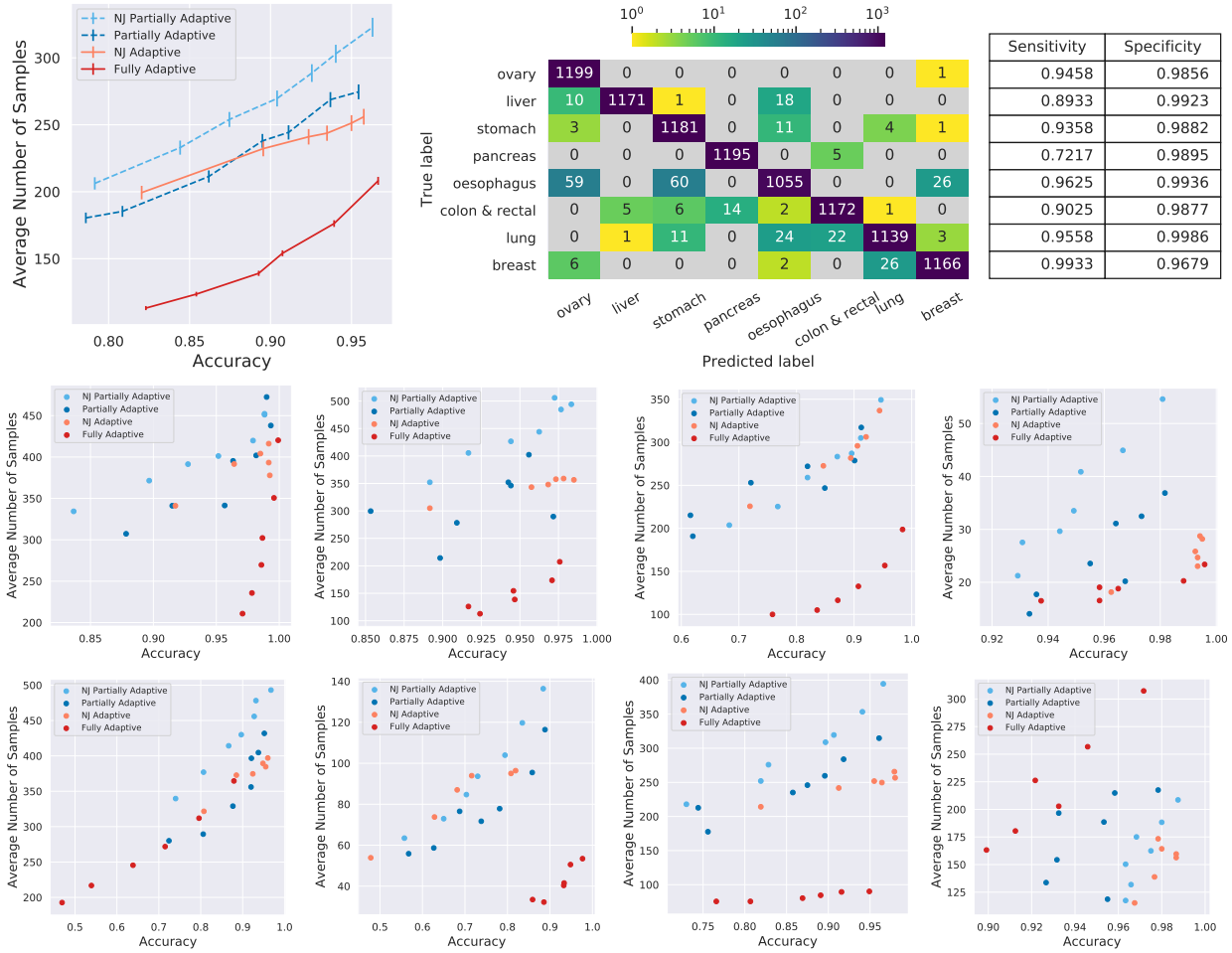
initialized to be uniform for all runs. On the horizontal axis, the accuracies of both algorithms were averaged over these 100 instances, where the accuracy is calculated as the percentage of correctly identified hypotheses among the 2,000 replications. On the vertical axis, the number of samples used by the algorithm is first averaged over the 2,000 replications and then averaged over the 100 instances.

Results In Figure 2 (left), we observe that 1) the performance of our fully adaptive algorithm dominates those of all other algorithms, 2) our partially adaptive algorithm outperforms all other partially adaptive algorithms, and 3) the performance of adaptive algorithms outperform those of partially adaptive algorithms. The threshold for entering Phase 2 policy in *NJ Adaptive* was set to be 0.1. Indeed, we observe that *NJ Adaptive* outperforms *NJ Partially Adaptive*. In Figure 2 (middle), δ equals to 0.05 for both *NJ Partially Adaptive* and *Partially Adaptive*. We observe that our partially and fully adaptive algorithms outperform *NJ Partially Adaptive* and *NJ Adaptive* instance-wise by large margins respectively in Figure 2 middle and left.

6.3. Real-World Experiments

Problem Setup Our real-world experiment is motivated by the design of DNA-based blood tests to detect cancer. In such a test, genetic mutations serve as potential signals for various cancer types, but DNA sequencing is, even today, expensive enough that the 'amount' of DNA that can be sequenced in a single test is limited if the test is to remain cost-effective. For example, one of the most-recent versions of these tests Cohen et al. (2018) involved sequencing just 4,500 *addresses* (from among 3 billion total addresses in the human genome), and other tests have had similar scale (e.g., Razavi et al. 2017, Chan et al. 2017, Phallen et al. 2017). Thus, one promising approach to the ultimate goal of a cost-effective test is adaptivity.

Our experiments are a close reproduction of the setup used by Cohen et al. (2018) to identify their 4,500 addresses. We use genetic mutation data from real cancer patients—the publicly-available COSMIC (Tate et al. 2019, Cosmic 2019) dataset, which includes the de-identified gene-screening panels for 1,350,015 patients. We treated 8 different types of cancer (as indicated in Cohen et al. 2018) as the 8 hypotheses, and identified 1,875,408 potentially mutated genetic addresses. To extract the tests, we grouped the the genetic addresses within an interval of 45 (see Cohen et al. 2018 for the biochemical reasons behind this choice), resulting in 581,754 potential tests. We then removed duplicated tests (i.e., the tests that



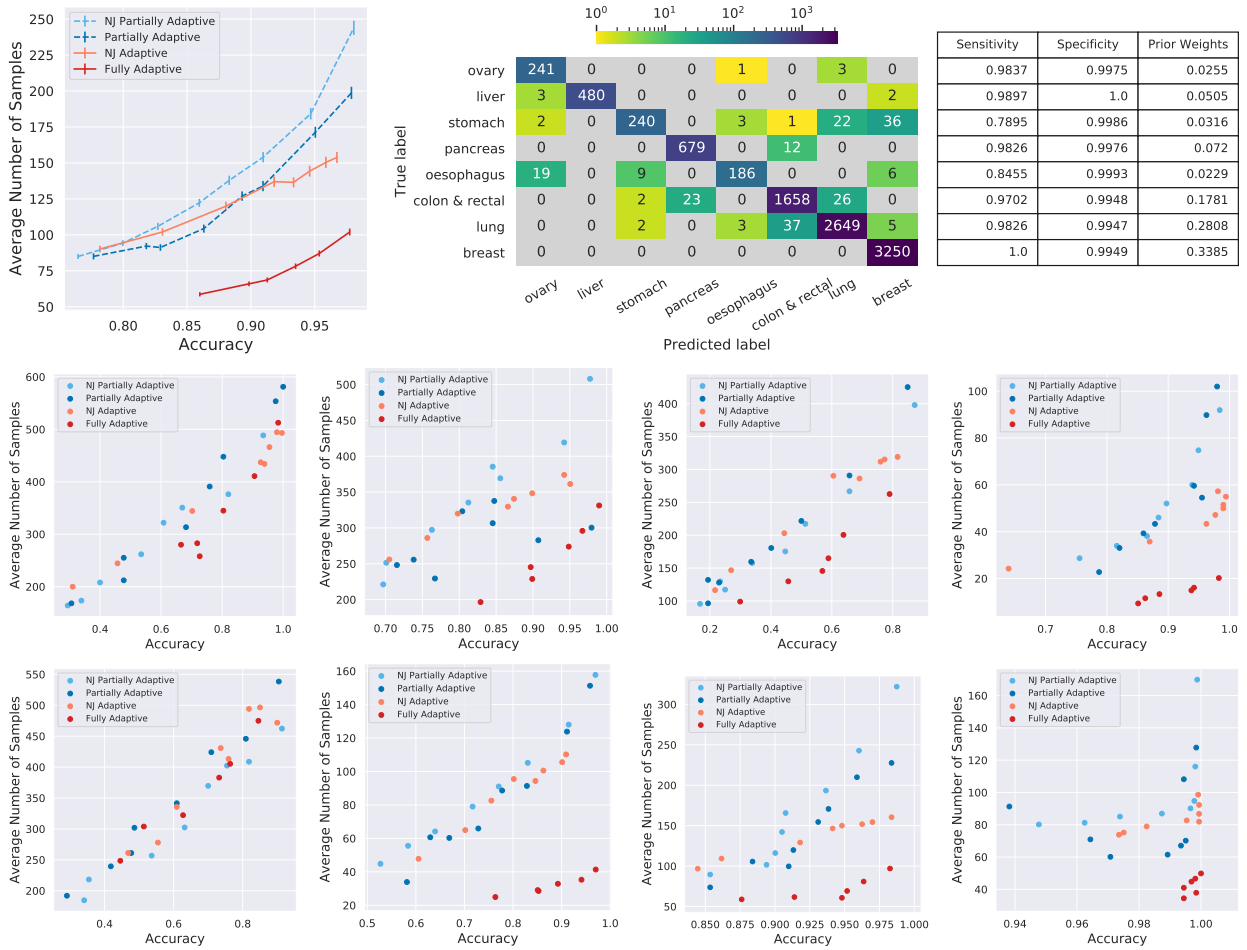
Sensitivity	Specificity
0.9458	0.9856
0.8933	0.9923
0.9358	0.9882
0.7217	0.9895
0.9625	0.9936
0.9025	0.9877
0.9558	0.9986
0.9933	0.9679

Figure 3 Comparison of our partially and fully adaptive algorithms with those of NJ's on real-world data, COSMIC, under *uniform prior*. Top row (left): each point is averaged over 9,600 replications. The error bars are the 95 percentage confidence intervals for the estimated means. Top row (middle): the confusion matrix of *Fully Adaptive* where the algorithm accuracy equals to 0.97, and each row sums up to 1,200. Top row (right): the sensitivity and specificity our algorithm (top row middle) for each cancer type. Middle and Bottom rows: the sensitivity figures for each cancer type. The order of the figures follows the order that they appear the confusion matrix (top row right). Each point in these figures is averaged over 1,200 replications contained in the top left figure.

share the same outcome distribution for all 8 cancer types), resulting in 23,135 final tests that we consider in our experiments. From the data, we extracted a "ground-truth" table of mutation probabilities containing the likelihood of a mutation in any of the 23,135 genetic address intervals being found in patients with any of the 8 cancer types. This served as the instance for our experiment. The majority of the mutation probabilities in our instances was either zero or some small positive number. To calculate the KL divergence between these probabilities, we replace zero with the number 10^{-10} in our instance.

Uniform Prior Results Although in reality, all patients have different priors for having different cancers, in our first set of experiments, to demonstrate the advantage of our algorithms, we assume that the truth hypothesis (cancer type) was drawn uniformly, and we initialize uniform priors for all algorithms. Figure 3, summarizes the performance of our algorithms under the assumption of uniform prior. Similar to Figure 2, in Figure 3 (top row left) we observe first that the performance of our algorithm dominates those to the rest algorithms. We also observe that our partially adaptive algorithm outperforms *NJ Partially Adaptive*. However, unlike Figure 2, we observe that *NJ Adaptive* underperforms *Partially Adaptive* when the accuracies are low on this instance. The threshold for entering Phase 2 policy, r, in *NJ Adaptive* was set to be 0.3. Since Phase 1 policy is less efficient than Phase 2 policy, we observe that the performance of *NJ Adaptive* is convex with respect to r—when r is small, the algorithm is more likely to alternate between Phases 1 and 2 policies and when r is large we spend more time in Phase 1 policy. As a result, we observe that variance of *NJ Adaptive* is relatively high when compared with those of our algorithms. On the other hand, we observe that our fully adaptive algorithm enjoys a narrower confidence interval as well as a better performance. Note that due to the nature of the sparsity of our instance, the performance of the random baseline was very poor when compared with these of *NJ Adaptive* and *Fully Adaptive* and thus was excluded. Figure 3 (top middle) is the confusion matrix corresponding to our fully adaptive algorithm where the algorithm accuracy equals to 0.97, and Figure 3 (top right) corresponds to the sensitivity and specificity of our algorithm for each cancer type (in the same ordering as) in the top middle figure. We observe that our adaptive algorithm performs reasonably well across different underlying true hypotheses (i.e., different cancer types). The middle and bottom rows of Figure 3 contain the sensitivity figures for each cancer type that are analogous to Figure 1. Due to the number of samples for each cancer type is limited, these plots are more volatile than Figure 3 top left. We observe that our fully adaptive algorithm outperforms the rest algorithms for the majority of cancer types. In addition, for those cancer types that our fully algorithm underperforms (e.g., breast cancer), we show below that when the prior of having a particular cancer increases, our algorithm becomes better at identifying this cancer.

Non-Uniform Prior Results To mimic the distribution of different cancer types in the real-world population, we exacted the number of newly diagnosed cancer cases from cancer.org (2021) to form the priors in our algorithms (the prior information is included in Figure 4



Sensitivity	Specificity	Prior Weights
0.9837	0.9975	0.0255
0.9897	1.0	0.0505
0.7895	0.9986	0.0316
0.9826	0.9976	0.072
0.8455	0.9993	0.0229
0.9702	0.9948	0.1781
0.9826	0.9947	0.2808
1.0	0.9949	0.3385

Figure 4 Comparison of our partially and fully adaptive algorithms with those of NJ's on real-world data, COSMIC, under non-uniform prior. Top row (left): each point is averaged over 9,600 replications. The error bars are the 95 percentage confidence intervals for the estimated means. Top row (middle): the confusion matrix of *Fully Adaptive* where the algorithm accuracy equals to 0.97. The sum of each row (the number of replications for each cancer type) equals to the corresponding prior multiplied by 9,600 and rounded to the nearest integer. Top row (right): the sensitivity and specificity our algorithm (top row middle) for each cancer type. Middle and Bottom rows: the sensitivity figures for each cancer type. The order of the figures follows the order that they appear the confusion matrix (top row right). The number of replications that we average over for each point in each figure equals to the number of replications for each cancer type.

bottom right). Note that neither *NJ Partially Adaptive* nor *NJ Adaptive* takes the prior into account when constructing the action sequences (or policies when randomization is allowed). Thus, the Phases 1 and 2 policies remain the same. (The prior was taken into account in the exiting criterion of both algorithms, and in addition, was taken into account in *NJ Adaptive* when entering Phase 2 policies.) On the other hand, both our partially adaptive and fully adaptive algorithms take the prior into the account when constructing the action sequences, i.e., the action sequences produced by our algorithms change as the prior changes. Thus, we

expect our algorithms to outperform their partially or fully adaptive counterparts under the non-uniform prior as well. Indeed, Figure 4 summarizes the results, and we observe that our partially adaptive and fully adaptive algorithms dominate NJ's partially and fully adaptive algorithms by a large margin, respectively. The threshold for entering Phase 2 policy, r, in *NJ Adaptive* was increased to 0.45 under this prior, and similar to Figure 3, we observe that the variance of *NJ Adaptive* is high when compared with our algorithms, indicating the trade off between variance and performance. Similar to Figure 3, Figure 4 (top middle) is the confusion matrix corresponding to our fully adaptive algorithm where the realized algorithm accuracy equals to 0.97. Figure 4 (top right) contains the sensitivity, specificity, and the prior of our algorithm for each cancer type in the top middle figure. Similarly, we include the sensitivity figures for each cancer type in Figure 4 middle and bottom rows. When compared with Figure 3, we observe that when the weight of a cancer type increases, with tuning, our algorithm becomes better at identifying those cancers (e.g., breast cancer and lung cancer) while maintaining the accuracies in identifying the rest cancer types.

7. Conclusion

In this work, we studied problem of active sequential hypothesis testing, motivated particularly by the design of adaptive liquid biopsies. We provided the first approximation guarantees for the ASHT problem in both the partially adaptive and fully adaptive setting, which grows linearly in the separability parameter s^{-1} and logarithmically in the number of candidate hypotheses and the (inverse) error rate δ^{-1}. Moreover for the partially adaptive version, by combining the SFR framework with a novel LP-based analysis, we improved the dependence on δ from $\log \frac{1}{\delta}$ in the naive analysis to $\log\log \frac{1}{\delta}$, which is much more favorable since in practice δ is usually very small. We further extend the fully adaptive algorithm to the total-error version by introducing a novel chance-constrained ODT problem (Appendix D).

To illustrate the applicability of our proposed method to the liquid biopsy problem, we conducted numerical studies on the COSMIC dataset. We found that our algorithms outperform the existing state-of-art benchmarks by large margins. Furthermore, our algorithms consider the priors for having different cancer types explicitly when constructing the action sequences, yielding superior performances under non-uniform priors. Finally, although the theoretical guarantees of our algorithms depend on the separability parameter s, we showed numerically that our modified algorithms work well in practice on both synthetic and real-world data.

References

Adler M, Heeringa B (2008) Approximating optimal binary decision trees. *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, 1–9 (Springer).

Arkin EM, Meijer H, Mitchell JS, Rappaport D, Skiena SS (1998) Decision trees for geometric models. *International Journal of Computational Geometry & Applications* 8(03):343–363.

Armitage P (1950) Sequential analysis with more than two alternative hypotheses, and its relation to discriminant function analysis. *Journal of the Royal Statistical Society. Series B* 12(1):137–144.

Awasthi P, Balcan MF, Long PM (2017) The power of localization for efficiently learning linear separators with noise. *Journal of the ACM (JACM)* 63(6):1–27.

Azar Y, Gamzu I (2011) Ranking with submodular valuations. *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, 1070–1079 (SIAM).

Azar Y, Gamzu I, Yin X (2009) Multiple intents re-ranking. *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 669–678 (ACM).

Balcan M, Beygelzimer A, Langford J (2006) Agnostic active learning. *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, 65–72.

Banerjee S, Karri SPK, Chatterjee S, Pal M, Paul RR, Chatterjee J (2016) Multimodal diagnostic segregation of oral leukoplakia and cancer. *Systems in Medicine and Biology (ICSMB), 2016 International Conference on*, 67–70 (IEEE).

Best MG, Sol N, Kooi I, Tannous J, Westerman BA, Rustenburg F, Schellen P, Verschueren H, Post E, Koster J, et al. (2015) Rna-seq of tumor-educated platelets enables blood-based pan-cancer, multiclass, and molecular pathway cancer diagnostics. *Cancer cell* 28(5):666–676.

Bettegowda C, Sausen M, Leary RJ, Kinde I, Wang Y, Agrawal N, Bartlett BR, Wang H, Luber B, Alani RM, et al. (2014) Detection of circulating tumor dna in early-and late-stage human malignancies. *Science translational medicine* 6(224):224ra24–224ra24.

Bubeck S, Munos R, Stoltz G (2009) Pure exploration in multi-armed bandits problems. *International conference on Algorithmic learning theory*, 23–37 (Springer).

cancerorg (2021) American cancer society: Cancer facts & statistics. URL https://cancerstatisticscenter.cancer.org/.

Castro RM, Nowak RD (2007) Minimax bounds for active learning. *International Conference on Computational Learning Theory*, 5–19 (Springer).

Chakaravarthy VT, Pandit V, Roy S, Sabharwal Y (2009) Approximating decision trees with multiway branches. *International Colloquium on Automata, Languages, and Programming*, 210–221 (Springer).

Chan KA, Woo JK, King A, Zee BC, Lam WJ, Chan SL, Chu SW, Mak C, Tse IO, Leung SY, et al. (2017) Analysis of plasma epstein–barr virus dna to screen for nasopharyngeal cancer. *New England Journal of Medicine* 377(6):513–522.

Chawla S, Gergatsouli E, Teng Y, Tzamos C, Zhang R (2019) Learning optimal search algorithms from data. *arXiv preprint arXiv:1911.01632* .

Chen Y, Hassani SH, Karbasi A, Krause A (2015) Sequential information maximization: When is greedy near-optimal? *Conference on Learning Theory*, 338–363.

Chernoff H (1959) Sequential design of experiments. *The Annals of Mathematical Statistics* 30(3):755–770.

Cohen JD, Li L, Wang Y, Thoburn C, Afsari B, Danilova L, Douville C, Javed AA, Wong F, Mattox A, et al. (2018) Detection and localization of surgically resectable cancers with a multi-analyte blood test. *Science* 359(6378):926–930.

Cosmic (2019) Cosmic - catalogue of somatic mutations in cancer. URL https://cancer.sanger.ac.uk/.

Cuzick J, Thorat MA, Andriole G, Brawley OW, Brown PH, Culig Z, Eeles RA, Ford LG, Hamdy FC, Holmberg L, et al. (2014) Prevention and early detection of prostate cancer. *The lancet oncology* 15(11):e484–e492.

Dasgupta S (2005) Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 337–344.

Du D, Hwang FK, Hwang F (2000) *Combinatorial group testing and its applications*, volume 12 (World Scientific).

Etzioni R, Urban N, Ramsey S, McIntosh M, Schwartz S, Reid B, Radich J, Anderson G, Hartwell L (2003) The case for early detection. *Nature Reviews Cancer* 3(4):243–252.

Even-Dar E, Mannor S, Mansour Y (2002) Pac bounds for multi-armed bandit and markov decision processes. *International Conference on Computational Learning Theory*, 255–270 (Springer).

Ferguson MK, Wang J, Hoffman PC, Haraf DJ, Olak J, Masters GA, Vokes EE (2000) Sex-associated differences in survival of patients undergoing resection for lung cancer. *The Annals of thoracic surgery* 69(1):245–249.

Gan K, Li AA, Lipton ZC, Tayur S (2020) Causal inference with selectively deconfounded data. *arXiv preprint arXiv:2002.11096* .

Golovin D, Krause A (2011) Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Intell. Res.* 42:427–486.

Guillory A, Bilmes JA (2011) Simultaneous learning and covering with adversarial noise. *ICML*.

Hanneke S, Yang L (2015) Minimax analysis of active learning. *The Journal of Machine Learning Research* 16(1):3487–3602.

Hanneke S, et al. (2014) Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning* 7(2-3):131–309.

Im S, Nagarajan V, Zwaan RVD (2016) Minimum latency submodular cover. *ACM Transactions on Algorithms (TALG)* 13(1):13.

Jemal A, Siegel R, Xu J, Ward E (2010) Cancer statistics, 2010. *CA: a cancer journal for clinicians* 60(5):277–300.

Jerant AF, Johnson JT, Sheridan C, Caffrey TJ, et al. (2000) Early detection and treatment of skin cancer. *American family physician* 62(2):357–386.

Jia S, Navidi F, Nagarajan V, Ravi R (2019) Optimal decision tree with noisy outcomes. *Advances in neural information processing systems* .

Kamath G, Tzamos C (2019) Anaconda: A non-adaptive conditional sampling algorithm for distribution testing. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 679–693.

Kim Y, Jeon J, Mejia S, Yao CQ, Ignatchenko V, Nyalwidhe JO, Gramolini AO, Lance RS, Troyer DA, Drake RR, et al. (2016) Targeted proteomics identifies liquid-biopsy signatures for extracapsular prostate cancer. *Nature communications* 7.

Kosaraju SR, Przytycka TM, Borgstrom R (1999) On an optimal split tree problem. *Workshop on Algorithms and Data Structures*, 157–168 (Springer).

Krause A, McMahan HB, Guestrin C, Gupta A (2008) Robust submodular observation selection. *Journal of Machine Learning Research* 9(Dec):2761–2801.

Laurent H, Rivest RL (1976) Constructing optimal binary decision trees is np-complete. *Information processing letters* 5(1):15–17.

Liu M, Oxnard G, Klein E, Swanton C, Seiden M, Liu MC, Oxnard GR, Klein EA, Smith D, Richards D, et al. (2020) Sensitive and specific multi-cancer detection and localization using methylation signatures in cell-free dna. *Annals of Oncology* 31(6):745–759.

Lorden G (1977) Nearly-optimal sequential tests for finitely many parameter values. *The Annals of Statistics* 1–21.

Mannor S, Tsitsiklis JN (2004) The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research* 5(Jun):623–648.

Manterola L, Guruceaga E, Pérez-Larraya JG, González-Huarriz M, Jauregui P, Tejada S, Diez-Valle R, Segura V, Samprón N, Barrena C, et al. (2014) A small noncoding rna signature found in exosomes of gbm patient serum as a diagnostic tool. *Neuro-oncology* 16(4):520–527.

Miller A, Hoogstraten B, Staquet M, Winkler A (1981) Reporting results of cancer treatment. *Cancer* 47(1):207–214.

Mitzenmacher M, Upfal E (2017) *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis* (Cambridge university press).

Naghshvar M, Javidi T (2013) Active sequential hypothesis testing. *The Annals of Statistics* 41(6):2703–2738.

Navidi F, Kambadur P, Nagarajan V (2020) Adaptive submodular ranking and routing. *Operations Research* .

Nowak RD (2009) Noisy generalized binary search. *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, 1366–1374.

O'Rourke N, Edwards R (2000) Lung cancer treatment waiting times and tumour growth. *Clinical Oncology* 12(3):141–144.

Paez JG, Jänne PA, Lee JC, Tracy S, Greulich H, Gabriel S, Herman P, Kaye FJ, Lindeman N, Boggon TJ, et al. (2004) Egfr mutations in lung cancer: correlation with clinical response to gefitinib therapy. *Science* 304(5676):1497–1500.

Phallen J, Sausen M, Adleff V, Leal A, Hruban C, White J, Anagnostou V, Fiksel J, Cristiano S, Papp E, et al. (2017) Direct detection of early-stage cancers using circulating tumor dna. *Science translational medicine* 9(403):eaan2415.

Razavi P, Li BT, Abida W, Aravanis A, Jung B, Shen R, Hou C, De Bruijn I, Gnerre S, Lim RS, et al. (2017) Performance of a high-intensity 508-gene circulating-tumor dna (ctdna) assay in patients with metastatic breast, lung, and prostate cancer. *J Clin Oncol* 35(18 Suppl).

Settles B (2009) Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Siegel RL, Miller KD, Jemal A (2015) Cancer statistics, 2015. *CA: a cancer journal for clinicians* 65(1):5–29.

Tate JG, Bamford S, Jubb HC, Sondka Z, Beare DM, Bindal N, Boutselakis H, Cole CG, Creatore C, Dawson E, et al. (2019) Cosmic: the catalogue of somatic mutations in cancer. *Nucleic acids research* 47(D1):D941–D947.

Vershynin R (2018) *High-dimensional probability: An introduction with applications in data science*, volume 47 (Cambridge university press).

Wald A (1945) Sequential tests of statistical hypotheses. *The annals of mathematical statistics* 16(2):117–186.

Wang Y, Singh A (2016) Noise-adaptive margin-based active learning and lower bounds under tsybakov noise condition. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Appendix A: Prerequisite: Subgaussian Random Variables

We will consider the commonly used subgaussian distributions (Vershynin (2018)). Loosely speaking, a random variable is subgaussian if its tail vanishes at a rate faster than some Gaussian distributions.

Definition 4 (Subgaussian norm) *Let X be a random variable, its subgaussian norm is defined as $\|X\|_{\psi_2} := \inf\{t : \mathbb{E}[e^{X^2/t^2}] \leq 2\}$. Moreover, X is called subgaussian if $\|X\|_{\psi_2} < \infty$.*

Many commonly used distributions satisfy this assumption, e.g., Bernoulli, uniform, and Gaussian distributions etc. We introduce a standard concentration bound for subgaussian random variables.

Theorem 6 (Hoeffding Inequality Vershynin 2018) *Let $X_1, ..., X_n$ be independent subgaussian random variables. Then for any $\eta > 0$, it holds that*

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \mathbb{E}X_i\right| \geq \eta\right] \leq 2\exp\left(-\frac{2\eta^2}{\sum_{i=1}^{n} \|X_i\|_{\psi_2}^2}\right).$$

To show the correctness of our algorithm, we need to consider the *log-likelihood ratio* (LLR), formally defined as follows:

Definition 5 *For any $a \in A$ and $h, g \in H$, define $Z(h, g; a) = \log \frac{\mathbb{P}_{h,a}(\xi)}{\mathbb{P}_{g,a}(\xi)}$ where $\xi \sim D_{\mu(h,a)}$.*

We will assume that the subgaussian norm of the LLR between two hypotheses is not too large when compared to the difference of their parameters, as formalized below.

Definition 6 *Let $\rho > 0$ be the minimal number s.t. for any distinct pair of hypotheses $h, g \in H$ and action $a \in A$, it holds that $\|Z(h, g; a)\|_{\psi_2} \leq \rho \cdot |\mu(g, a) - \mu(h, a)|$.*

We will present an error analysis for general ρ. Prior to that, we first point out that many common distributions satisfy $\rho = O(1)$.

Examples. It is straightforward to verify that $\rho = O(1)$ for the following common distributions:

- Bernoulli distributions: $D_\theta = Ber(\theta)$ where $\theta \in [\theta_{min}, \theta_{max}]$ for constants $\theta_{min}, \theta_{\max} \in (0, 1)$, and
- Gaussian distributions: $D_\theta = N(\theta, 1)$ where $\theta \in [\theta_{min}, \theta_{max}]$ for constants $\theta_{min} < \theta_{\max}$.

Take Bernoulli distribution as an example. Fix any hypotheses $h, g \in H$ and action $a \in A$, write $\Delta = \mu(h, a) - \mu(g, a)$. Then, $Z = Z(h, g; a)$ can be rewritten as

$$Z = \begin{cases} \log(1 + \frac{\Delta}{\mu(g,a)}), & \text{w.p. } \mu(h, a), \\ \log(1 - \frac{\Delta}{1-\mu(g,a)}), & \text{w.p. } 1 - \mu(h, a). \end{cases}$$

Since $0 < \theta_{min} \leq \mu(g, a) \leq \theta_{\max} < 1$, we have $|Z| \leq C|\Delta|$ almost surely where $C = 2\max\{(1 - \theta_{max})^{-1}, \theta_{min}^{-1}\}$. Moreover, it is known that (see Vershynin 2018) any subgaussian random variable Z satisfies $\|Z\|_{\psi_2} \leq \frac{1}{\ln 2}\|Z\|_\infty$, so it follows that

$$\|Z\|_{\psi_2} \leq \frac{1}{\ln 2}\|Z\|_\infty \leq \frac{C\Delta}{\ln 2} = O(\Delta).$$

Thus $\rho = O(1)$.

Appendix B: Proof of Proposition 1

B.1. Error Analysis

We first prove that at each timestamp $\tau(h)$, with high probability our algorithm terminates and returns h.

Lemma 1 *Let $B>0$. If $h\in H$ is the true hypothesis, then w.p. $1-e^{-\Omega(\rho^{-2}\alpha B)}$, it holds $\log\Lambda(h,g;\tau(h))\geq\frac{1}{2}\alpha B$ for all $g\neq h$.*

Proof of Lemma 1 Let $\tilde{\sigma}=(a_1,a_2,...)$ be the sequence *after* the boosting step, so $a_1=...=a_\alpha, a_{\alpha+1}=...=a_{2\alpha}$, so on so forth. Write $Z_i=Z(h,g;a_i)$, then for any $t\geq 1$, it holds $\log\Lambda(h,g;t)=\sum_{i=1}^{t}Z_i$. By the definition of cover time, $\sum_{i=1}^{\tau(h)}d(h,g;a_i)=\sum_{i=1}^{\tau(h)}\mathbb{E}[Z_i]\geq\alpha B$. Thus,

$$\begin{aligned}\mathbb{P}_h\left[\log\Lambda\big(h,g;\tau(h)\big)<\frac{1}{2}\alpha B\right]&=\mathbb{P}_h\left[\sum_{i=1}^{\tau(h)}Z_i<\frac{1}{2}\alpha B\right]\\&\leq\mathbb{P}_h\left[\left|\sum_{i=1}^{\tau(h)}Z_i-\sum_{i=1}^{\tau(h)}\mathbb{E}Z_i\right|>\frac{1}{2}\sum_{i=1}^{\tau(h)}\mathbb{E}Z_i\right].\end{aligned}\tag{2}$$

By Theorem 6,

$$\text{Equation}(2)\leq\exp\left(-\Omega\left(\frac{(\alpha B)^2}{\sum_{i=1}^{\tau(h)}\|Z_i\|_{\psi_2}^2}\right)\right).\tag{3}$$

We next show that $\sum_{i=1}^{\tau(h)}\|Z_i\|_{\psi_2}^2\leq O(\rho^2\alpha B)$. Write $\Delta_i=\mu(h,a_i)-\mu(g,a_i)$, then by Assumption 2, $\Delta_i^2\leq C_2\cdot d(h,g;a_i)$. Note that $\|Z_i\|_{\psi_2}\leq\rho\Delta_i$, so it follows that

$$\sum_{i=1}^{\tau(h)}\|Z_i\|_{\psi_2}^2\leq\rho^2\sum_{i=1}^{\tau(h)}\Delta_i^2\leq C_2\rho^2\sum_{i=1}^{\tau(h)}d(h,g;a_i).\tag{4}$$

Recall that σ is the sequence *before* boosting. Write $t=CT(f_h^B,\sigma)$ for simplicity. By the definition of cover time,

$$\sum_{i=1}^{\alpha t}d(h,g;a_i)\geq\alpha B\geq\sum_{i=1}^{\alpha(t-1)}d(h,g;a_i).$$

Note that $\tau(h)=\alpha t$, so

$$\sum_{i=1}^{\alpha t}d(h,g;a_i)\leq 2\sum_{i=1}^{\alpha(t-1)}d(h,g;a_i)\leq 2\alpha B.$$

Combining the above with Equation (4), we have

$$\sum_i\|Z_i\|_{\psi_2}^2\leq 2C_2\rho^2\alpha B.$$

Substituting into Equation (3), we obtain

$$\mathbb{P}_h\left[\log\Lambda\big(h,g;\tau(h)\big)<\frac{1}{2}\alpha B\right]\leq e^{-\Omega(\rho^{-2}\alpha B)}.$$

The proof completes by applying the union bound over all $g\in H\backslash\{h\}$. □

By a similar approach we may also show that it is unlikely that the algorithm terminates at a wrong time stamp before scanning the correct one.

Lemma 2 *Let $B > 0$. If $h \in H$ is the true hypothesis, then for any $g \neq h$, it holds that $\log \Lambda(g, h; \tau(g)) < \frac{1}{2}\alpha B$ with probability $1 - e^{-\Omega(\rho^{-2}\alpha B)}$.*

We are able to bound the error of the RnB algorithm by combining Lemma 1 and Lemma 2.

Proposition 3 *For any true hypothesis $h \in H$, algorithm $RnB(B, \alpha)$ returns h with probability at least $1 - |H|e^{-\Omega(\rho^{-2}\alpha B)}$. In particular, if the outcome distribution D_μ is $Ber(\mu)$, then $\rho = O(1)$ and the above probability becomes $1 - |H|e^{-\Omega(\alpha B)}$.*

B.2. Cost Analysis

Recall that in Section 4, only Step (C) remains to be shown, which we formally state below.

Proposition 4 *Let (σ, T) be a δ-PAC-error partially adaptive algorithm. For any $B \leq \log \delta^{-1}$ and $h \in H$, it holds that $\mathbb{E}_h[T] \geq \Omega\big(s \cdot \mathrm{CT}(f_h^B, \sigma)\big)$.*

We fix an arbitrary $h \in H$ and write $\mathrm{CT}_h := \mathrm{CT}(f_h^B, \sigma)$, where we recall that σ is the sequence of actions before boosting (do not confuse with $\tilde{\sigma}$). To relate the stopping time T (under h) to the cover time of the submodular function for h in σ, we introduce a linear program. We will show that for suitable choice of d, we have

- $LP^*(d, \mathrm{CT}_h - 1) \leq \mathbb{E}_h[T]$, and
- $LP^*(d, \mathrm{CT}_h - 1) \geq \Omega(s \cdot \mathrm{CT}_h)$.

Hence proving Step (C) in the high-level proof sketched in section 4.

We now specify our choice of d. For any $d_1, ..., d_N \in \mathbb{R}_+$, write $d^t := \sum_{i=1}^t d_i$ for any t and consider

$$\begin{aligned} LP(d,t): \quad \min_z & \sum_{i=1}^N i \cdot z_i \\ s.t. & \sum_{i=1}^N d^i z_i \geq d^t, \\ & \sum_{i=1}^N z_i = 1, \\ & z \geq 0. \end{aligned}$$

We will consider the following choice of d_i's. Suppose (σ, T) has δ-PAC-error where $\delta \in (0, 1/4]$. For any pair of hypotheses h, g and any set of actions S, define

$$K_{h,g}^B(S) = \min\left\{1, B^{-1} \sum_{a \in S} d(h, g; a)\right\}.$$

Hence,

$$f_h^B(S) = \frac{1}{|H| - 1} \sum_{g \in H \setminus \{h\}} K_{h,g}^B(S).$$

Fix any $B \leq \log \delta^{-1}$ and let g be the last hypothesis separated from h, i.e.,

$$g := \arg \max_{h' \in H \setminus \{h\}} \left\{\mathrm{CT}(K_{h,h'}^B, \sigma)\right\}.$$

Then by the definition of cover time, we have $\mathrm{CT}_h = \mathrm{CT}(f_h^B, \sigma) = \mathrm{CT}(K_{hg}^B, \sigma)$. Without loss of generality[11], we assume all actions a satisfy $\mu(h,a) = \mu(g,a)$ in $\tilde{\sigma} = (a_1, .., a_N)$. We choose the LP parameters to be $d_i = d(h,g,a_i)$ for $i \in [N]$.

Outline We will first show that the LP optimum is upper bounded by the expected termination time T (Proposition 5). We then lower bound it in terms of CT_h (Proposition 6).

Proposition 5 *Suppose (σ, T) has δ-PAC-error for some $0 < \delta \le \frac{1}{4}$. Let $z_i = \mathbb{P}_h[T = i]$ for $i \in [N]$, then $z = (z_1, ..., z_N)$ is feasible to $LP(d, \mathrm{CT}_h - 1)$.*

Note that $\mathbb{E}_h[T]$ is simply the objective value of z, thus Proposition 5 immediately implies:

Corollary 1 $\mathbb{E}_h[T] \ge LP^*(d, \mathrm{CT}_h - 1)$.

We next lower bound the expected log-likelihood when the algorithm stops.

Lemma 3 (Nowak 2009) *Let $\mathbb{A}$ be any algorithm (not necessarily partially adaptive) for the ASHT problem. Let $h, g \in H$ be any pair of distinct hypotheses and O be the random output of $\mathbb{A}$. Define the error probabilities $P_{hh} = \mathbb{P}_h(O = h)$ and $P_{hg} = \mathbb{P}_h(O = g)$. Let Λ be the likelihood ratio between h and g when $\mathbb{A}$ terminates. Then,*

$$\mathbb{E}_h[\log \Lambda] \ge P_{hh} \log \frac{P_{hh}}{P_{hg}} + (1 - P_{hh}) \log \frac{1 - P_{hh}}{1 - P_{hg}}.$$

Proof of Lemma 3 Let $\mathcal{E}$ be the event that the output is h. Then by Jensen's inequality,

$$\mathbb{E}_h[\log \Lambda_T | \mathcal{E}] \ge -\log \mathbb{E}_h[\Lambda^{-1} | \mathcal{E}] = -\log \frac{\mathbb{E}_h[\mathbb{1}(\mathcal{E}) \cdot \Lambda^{-1}]}{\mathbb{P}_h(\mathcal{E})}. \tag{5}$$

Recall that an algorithm can be viewed as a decision tree in the following way. Each internal node is labeled with an action, and each edge below it corresponds to a possible outcome; each leaf corresponds to termination, and is labeled with a hypothesis corresponding to the output. Write $\sum_x$ as the summation over all leaves and let $p_h(x)$ (resp. $p_g(x)$) be the probability that the algorithm terminates in leaf x under h (resp. g), then,

$$\begin{aligned}
\mathbb{E}_h[\mathbb{1}(\mathcal{E}) \cdot \Lambda^{-1}] &= \sum_x \mathbb{1}(x \in \mathcal{E}) \cdot \Lambda^{-1}(x) \cdot p_h(x) \\
&= \sum_x \mathbb{1}(x \in \mathcal{E}) \cdot \frac{p_g(x)}{p_h(x)} p_h(x) \\
&= \sum_x \mathbb{1}(x \in \mathcal{E}) \cdot p_h(x) \\
&= \mathbb{E}_h[\mathbb{1}(x \in \mathcal{E})] = P_{hg}.
\end{aligned}$$

Combining the above with Equation (5), we obtain

$$\mathbb{E}_h[\log \Lambda | \mathcal{E}] \ge \log \frac{P_{hh}}{P_{hg}}.$$

Similarly, we have $\mathbb{E}_h(\log \Lambda | \bar{\mathcal{E}}) \ge \log \frac{1 - P_{hh}}{1 - P_{hg}}$, where $\bar{\mathcal{E}}$ is the event that the output is not h. The proof follows immediately by combining these two inequalities. □

[11] If there is some action a with $d(h, g; a) = 0$, then we simply remove it. This will not change the argument.

To show Proposition 5 we need a standard concept—*stopping time*.

Definition 7 (Stopping time Mitzenmacher and Upfal 2017) *Let $\{X_i\}$ be a sequence of random variables and T be an integer-valued random variable. If for any integer t, the event $\{T=t\}$ is independent with $X_{t+1}, X_{t+2}, ...$, then T is called a* **stopping time** *for X_i's.*

Lemma 4 (Wald's Identity) *Let $\{X_i\}_{i\in\mathbb{N}}$ be independent random variables with means $\{\mu_i\}_{i\in\mathbb{N}}$, and let T be a stopping time w.r.t. X_i's. Then, $\mathbb{E}[\sum_{i=1}^{T} X_i] = \mathbb{E}[\sum_{i=1}^{T} \mu_i]$.*

Proof of Proposition 5. One may verify that the lower bound in Lemma 3 is increasing w.r.t P_{hh} and decreasing w.r.t P_{hg}. Therefore, since $\mathbb{A}$ has δ-PAC-error, by Lemma 3 it holds that

$$\mathbb{E}_h[\log \Lambda(h,g;T)] \geq (1-\delta)\log\frac{1-\delta}{\delta} + \delta\log\frac{\delta}{1-\delta} \geq \frac{1}{2}\log\frac{1}{\delta} \geq B \geq d^{\mathrm{CT}_h - 1}.$$

By Lemma 4,

$$\sum_{i=1}^{N} d^i z_i = \sum_{i=1}^{N} d^i \cdot \mathbb{P}_h(T=i) = \mathbb{E}_h[\log \Lambda(h,g;T)].$$

The proof follows by combining the above. □

So far we have upper bounded $LP^*(d, \mathrm{CT}_h - 1)$ using $\mathbb{E}_h[T]$. To complete the proof, we next lower bound $LP^*(d, \mathrm{CT}_h - 1)$ by $\Omega(s \cdot \mathrm{CT}_h)$.

Lemma 5 *$LP^* = \min_{i\leq t<j} LP^*_{ij}$ where $LP^*_{ij} = i + (j-i)\frac{d^t - d^i}{d^j - d^i}$.*

Proof of Lemma 5 Observe that for any optimal solution, the inequality constraint must be tight. By linear algebra, we deduce that any basic feasible solution has support size two.

Consider the solutions whose only nonzero entries are i, j. Then, $LP(d,t)$ becomes

$$\begin{aligned} LP_{ij}(d,t): \quad \min_{z_i, z_j} \quad & iz_i + jz_j \\ s.t. \ & d^i z_i + d^j z_j = d^t, \\ & z_i + z_j = 1, \\ & z \geq 0. \end{aligned}$$

Note that since $d^i < d^j$, $LP_{i,j}(d,t)$ admits exactly one feasible solution, whose objective value can be easily verified to be $LP^*_{ij} := i + (j-i)\frac{d^t - d^i}{d^j - d^i}$. □

Now we are ready to lower bound the LP optimum.

Proposition 6 *For any $d = (d_1, ..., d_N) \in \mathbb{R}^N$ and $t \in \mathbb{N}$, it holds that $LP^*(d,t) \geq t \cdot \min\{d_i\}_{i\in[N]}$.*

Proof of Lemma 6 By Lemma 5, it suffices to show that $LP^*_{ij} \geq d^t$ for any $i \leq t < j$. Since $d^k < k$ for any integer k,

$$(j - d^t)(d^t - d^i) \geq (d^j - d^t)(d^t - i).$$

Rearranging, the above becomes

$$i(d^j - d^i) + (j-i)(d^t - d^i) \geq d^t(d^j - d^i),$$

i.e.,

$$i + (j-i)\frac{d^t - d^i}{d_j - d^i} \geq d^t.$$

Note that the LHS is exactly LP^*_{ij}, thus $LP^*(d,t) \geq d^t \geq t \cdot \min\{d_i\}_{i\in[N]}$ for any $t \in \mathbb{N}$. □

It immediately follows that $LP^*(d,t) \geq st$, completing the proof of Proposition 1.

Appendix C: Proof of Proposition 2

We first formally define a decision tree, not only for mathematical rigor but more importantly, for the sake of introducing a novel variant of ODT. Recall that Ω is the space of the test outcomes, which we assume to be discrete for simplicity.

Definition 8 (Decision Trees) *A decision tree is a rooted tree, each of whose interior (i.e., non-leaf) node v is associated with a state (A_v, T_v), where T_v is a test and $A_v \subseteq H$. Each interior node has $|\Omega|$ children, each of whose edge to v is labeled with some outcome. Moreover, for any interior node v, the set of alive hypotheses A_v is the set of hypotheses consistent with the outcomes on the edges of the path from the root to v. A node ℓ is a leaf if $|A_\ell| = 1$. The decision tree terminates and outputs the only alive hypothesis when it reaches a leaf.*

To relate OPT^{FA}_δ to the optimum of a suitable ODT instance, we introduce a novel variant of ODT. As opposed to the ordinary ODT where the output needs to be correct with probability 1, in the following variant, we consider decision trees which may *err* sometimes:

Definition 9 (Incomplete Decision Trees) *An incomplete decision tree is a decision tree whose leaves ℓ's are associated with* states *(A_ℓ, p_ℓ)'s, where A_ℓ represents the subset of hypotheses consistent with all outcomes so far, and p_ℓ is a distribution over A_ℓ. A hypothesis is randomly drawn from p_ℓ and is returned as the identified hypothesis (possibly wrong).*

Now we already to introduce *chance-constrained ODT problem* (CC-ODT). Given an error budget $\delta > 0$, we aim to find the minimal cost decision tree whose error is within δ. There are two natural ways to interpret "error", which will both be considered in Appendices C and D. In the first one, we require the error probability under *any* hypothesis to be lower than the given error budget. In the other one, we only require the *expected* error probability over all hypotheses to be within the budget. Intuitively, the second version allows for more flexibility since the errors under different hypotheses may differ significantly, rendering the analysis more challenging since we do not know how the error budget is allocated to each hypothesis. We formalize these two versions below. Let O be the random outcome returned by the tree.

CC-ODT with PAC-Error. An incomplete decision tree is *δ-PAC-Valid* if, for any true hypothesis h, it returns h with probability at least $1-\delta$, formally,

$$\mathbb{P}_h(O \neq h) \leq \delta, \quad \forall h \in H.$$

CC-ODT with Total-Error. An incomplete decision tree is *δ-Total-Valid* if, for the total error probability is at most δ, formally,

$$\sum_{h\in H} \pi(h) \cdot \mathbb{P}_h[O \neq h] \leq \delta,$$

where π is the prior distribution. The goal in both versions is to find an incomplete decision tree with minimal expected cost, subject to the corresponding error constraint.

For the proof of Proposition 2, consider the PAC-error version of CC-ODT. It turns out that this version of CC-ODT is indeed quite trivial (unlike the total-error version): below we show that under PAC-error, CC-ODT is almost equivalent to the ordinary ODT problem.

Lemma 6 *Suppose $\delta \in (0, \frac{1}{2})$, and $\mathbb{T}$ is a δ-PAC-valid decision tree. Then, $\mathbb{T}$ must also be 0-valid.*

Proof of Lemma 6 It suffices to show that there is no incomplete node in $\mathbb{T}$. For the sake of contradiction, assume $\mathbb{T}$ has an incomplete node ℓ with state (A_ℓ, p_ℓ). By the definition of incomplete node, $|A_\ell| \geq 2$, so there is an $h \in A_\ell$ with $p_\ell(h) \leq \frac{1}{2}$. Now suppose h is the true hypothesis. Since each hypothesis traces a unique path in any decision tree, regardless of whether or not it is incomplete, h will reach node ℓ with probability 1. Then at ℓ, the decision tree returns h with probability $p_\ell(h) = 1 - \sum_{g \in A_\ell : g \neq h} p_\ell(g) \leq \frac{1}{2}$, and hence $\mathbb{P}_h[O \neq h] \geq \frac{1}{2}$, reaching a contradiction. □

For the reader's convenience, we recall that an ASHT instance $\mathcal{I}$ is associated with an ODT instance $\mathcal{I}_{ODT}$, defined as follows. Each action corresponds to a test $T_a : H \to \Omega_a$ with $T_a(h) = \mu(h, a)$, where $\Omega_a = \{\mu(h, a) : h \in H\}$, and the cost T_a is $c(a) = \lceil s(a)^{-1} \log(|H|/\delta) \rceil$. Denote ODT^*_δ the minimal cost of any δ-PAC-valid decision tree for $\mathcal{I}_{ODT}$. Then we immediately obtain the following from the Lemma 6.

Corollary 2 *If $\delta \in (0, \frac{1}{2})$, then $ODT^*_0 = ODT^*_\delta$.*

Now we are able to complete the proof of the main proposition.

Proof of Proposition 2. Given a δ-PAC-error algorithm $\mathbb{A}$, we show how to construct a δ-PAC-valid decision tree $\mathbb{T}$ as follows. View $\mathbb{A}$ as a decision tree (discretize the outcome space if it is continuous). Replace each action a in $\mathbb{A}$ with the test T_a. Note that the cost of T_a is $s(a)^{-1} \log(|H|/\delta) \leq s^{-1} \log(|H|/\delta)$. Therefore by Lemma 6,

$$ODT^*_0 = ODT^*_\delta \leq c(\mathbb{T}) \leq s^{-1} \log \frac{|H|}{\delta} \cdot OPT^{FA}_\delta. \qquad \square$$

Appendix D: Total Error Version

In the last section we defined the total-error version of the CC-ODT problem. The total error version of the ASHT problem can be defined analogously, so we do not repeat it here. We say an algorithm is said to be **δ-total-error** if the total probability (averaged with respect to the prior π) of erroneously identified a wrong hypothesis is at most δ. The following is our main result for the total-error version.

Theorem 3 *Given an s-separated instance with uniform prior π and any $\delta \in (0, 1/4)$, for both the partially and fully adaptive versions, there exist polynomial-time δ-total-error algorithms with expected cost $O\left(s^{-1}\left(1 + |H|\delta^2\right) \log\left(|H|\delta^{-1}\right) \log |H|\right)$ times the optimum.*

In particular, if $\delta \leq O(|H|^{-1/2})$, then the above is polylog-approximation for fixed s.

We will first prove Theorem 3 for the fully adaptive version, and then show how the same proof works for the partially adaptive version. Unlike the PAC-error version where CC-ODT is almost equivalent to ODT, in

the total-error version their optima can differ by a $\Omega(|H|)$ factor. We construct a sequence of ODT instances $\mathcal{I}_n$, where $n \in \mathcal{Z}^+$, with $ODT^*_\delta(\mathcal{I}_n)/ODT^*_0(\mathcal{I}_n) = O(\frac{1}{n})$. Suppose there are $n+2$ hypotheses $h_1, ..., h_n$ and g, h, with $\pi(g) = \pi(h) = 0.49$ and $\pi(h_i) = \frac{1}{50n}$ for $i = 1, ..., 50$. Each (binary) test partitions $[n+2]$ into a singleton and its complement. Consider error budget $\delta = \frac{1}{4}$, then for each n we have $ODT^*_\delta(\mathcal{I}_n) = 1$. In fact, we may simply perform a test to separate g and h, and then return the one (out of g and h) that is consistent with the outcome. The total error of this algorithm is $1/50 < \delta$. On the other hand, $ODT^*_0(\mathcal{I}_n) = n+1$.

However, for uniform prior, this gap is bounded:

Proposition 7 *Suppose the prior π is uniform. Then, for any $\delta \in (0, \frac{1}{4})$, it holds*

$$ODT^*_0 \le \big(1 + O(|H|\delta^2)\big) \cdot ODT^*_\delta.$$

To show the above, we need the following building block.

Lemma 7 *Suppose the prior π is uniform. Then, for any $\delta \in [0, \frac{1}{4})$, the total prior probability density on the incomplete nodes is bounded by $\sum_{\ell \text{ inc.}} \pi(A_\ell) \le 2\delta$.*

Proof of Lemma 7 Let ℓ be an incomplete node with state (A_ℓ, p_ℓ) and write $p = p_\ell$ for simplicity. Then, the error probability contributed by ℓ is

$$\begin{aligned}\sum_{h \in A_\ell} \pi(h) \cdot (1 - p(h)) &= \sum_{h \in A_\ell} \pi(h) - \sum_{h \in A_\ell} \pi(h) \cdot p(h) \\ &= \pi(A_\ell) - \frac{1}{n} \sum_{h \in A_\ell} p(h) \\ &= \frac{|A_\ell|}{n} - \frac{1}{n} \ge \frac{1}{2}\pi(A_\ell),\end{aligned}$$

where the last inequality follows since $|A_\ell| \ge 2$. By the definition of δ-PAC-error, it follows that

$$\delta \ge \sum_{\ell \text{ inc.}} \sum_{h \in A_\ell} \pi(h) \cdot (1 - p(h)) \ge \frac{1}{2} \sum_{\ell \text{ inc.}} \pi(A_\ell),$$

i.e., $\sum_{\ell \text{ inc.}} \pi(A_\ell) \le 2\delta$. □

Proof of Proposition 7. It suffices to show how to convert a decision tree $\mathbb{T}$ with δ-total-error to one with 0-total-error, without increasing the cost by too much. Consider each incomplete node A_ℓ in $\mathbb{T}$. We will replace A_ℓ with a (small) decision tree that uniquely identifies a hypothesis in A_ℓ. Consider any distinct hypotheses $g, h \in A_\ell$. Then by Assumption 2, there is an action $a \in A$ with $d(g, h; a) \ge s$. So if we select T_a, then by Hoeffding bound (Theorem 6), we have that with high probability at least one of g and h will be eliminated, and the number of alive hypotheses in A_ℓ reduces by at least 1. Thus, by repeating this procedure iteratively for at most $|A_\ell| - 1$ times, we can identify a unique hypothesis. Since each test T_a corresponds to selecting a for $c(a) = s(a)^{-1} \log(|H|/\delta) \le s^{-1} \log(|H|/\delta)$ times in a row, this procedure increases the total cost by $\sum_{\ell \text{ inc.}} \pi(A_\ell) \cdot (|A_\ell| \cdot s^{-1} \log(|H|/\delta)$. Therefore,

$$\begin{aligned}ODT^*_0 &\le ODT^*_\delta + \sum_{\ell \text{ inc.}} \pi(A_\ell)|A_\ell| s^{-1} \log \frac{|H|}{\delta} \\ &= ODT^*_\delta + \sum_{\ell \text{ inc.}} \pi(A_\ell)|H|\pi(A_\ell) \cdot s^{-1} \log \frac{|H|}{\delta} \\ &= ODT^*_\delta + O\big(s^{-1}|H| \log \frac{|H|}{\delta} \cdot \sum_{\ell \text{ inc.}} \pi(A_\ell)^2\big). \qquad (6)\end{aligned}$$

Since $\sum_{\ell \text{ inc.}} \pi(A_\ell) \le 2\delta$ and each $\pi(A_\ell)$'s is non-negative, we have $\sum_{\ell \text{ inc.}} \pi(A_\ell)^2 \le \big(\sum_{\ell \text{ inc.}} \pi(A_\ell)\big)^2 \le 4\delta^2$. Further, by Pinsker's inequality, we have $ODT^*_\delta = \Omega(s^{-1} \log \frac{|H|}{\delta})$. Combining these two facts with Equation (6), we obtain $ODT^*_0 \le \big(1 + O(|H|\delta^2)\big) \cdot ODT^*_\delta$. □

The following lemma can be proved using the same idea of the proof of Proposition 2.

Lemma 8 $ODT^*_\delta \le O\big(s^{-1} \log(|H|/\delta)\big) OPT^{FA}_\delta$.

Now we are ready to show Theorem 3.

$$\begin{aligned} GRE &\le O(\log |H|) \cdot ODT^*_0 && \text{(Theorem 5)} \\ &\le O\big((1 + O(|H|\delta^2)) \log |H|\big) \cdot ODT^*_\delta && \text{(Lemma 7)} \\ &\le O\big((1 + O(|H|\delta^2)) s^{-1} \log^2 \frac{|H|}{\delta} \log |H|\big) \cdot OPT^{FA}_\delta. && \text{(Lemma 8)} \end{aligned}$$

The above proof can be adapted to the partially adaptive version straightforwardly as follows. Observing that partially adaptive algorithms can be viewed as a special case of the fully adaptive, we can define $ODT^*_{0,\text{PA}}$ and $ODT^*_{\delta,\text{PA}}$ (analogous to ODT^*_0 and ODT^*_δ) for the partially adaptive version, as the optimal cost of any partially adaptive decision tree with 0 or δ error. By replacing ODT^*_δ and ODT^*_0 with $ODT^*_{0,\text{PA}}$ and $ODT^*_{\delta,\text{PA}}$, one may immediatly verify that inequalities in Lemma 7 and 8 hold for the partially adaptive version. Furthermore, the first inequality above can be established for the partially adaptive version by replacing Theorem 5 with Theorem 4, hence completing the proof.