OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0030-364X | EISSN 1526-5463 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000
© 0000 INFORMS

Optimal Decision Tree and Submodular Ranking with Noisy Outcomes

Su Jia, Fatemeh Navidi, Viswanath Nagarajan, R. Ravi

A fundamental task in active learning involves performing a sequence of tests to identify an unknown hypothesis that is drawn from a known distribution. This problem, known as optimal decision tree induction, has been widely studied for decades and the asymptotically best-possible approximation algorithm has been devised for it. We study a generalization where certain test outcomes are noisy, even in the more general case when the noise is persistent, i.e., repeating a test gives the same noisy output. We design new approximation algorithms for both the non-adaptive setting, where the test sequence must be fixed *a-priori*, and the adaptive setting where the test sequence depends on the outcomes of prior tests. Previous work in the area assumed at most a logarithmic number of noisy outcomes per hypothesis and provided approximation ratios that depended on parameters such as the minimum probability of a hypothesis. Our new approximation algorithms provide guarantees that are nearly best-possible and work for the general case of a large number of noisy outcomes per test or per hypothesis where the performance degrades smoothly with this number. In fact, our results hold in a significantly more general setting, where the goal is to cover stochastic submodular functions. We evaluate the performance of our algorithms on two natural applications with noise: toxic chemical identification and active learning of linear classifiers. Despite our theoretical logarithmic approximation guarantees, our methods give solutions with cost very close to the information theoretic minimum, demonstrating the effectiveness of our methods.

Key words: Approximation Algorithms, Optimal Decision Tree, Submodular functions, Active Learning

1. Introduction

The classic Optimal Decision Tree (ODT) problem involves identifying an initially unknown *hypothesis* h that is drawn from a known probability distribution over a set of hypotheses. We can perform *tests* in order to distinguish between these hypotheses. Each test produces a binary outcome (positive or negative) and the precise outcome of each test-hypothesis pair is known beforehand, and thus an instance of ODT can be viewed as a ± 1-valued matrix M with the tests as rows and hypotheses as columns. The goal is to identify the true hypothesis h using the fewest tests.

As a motivating application, consider the following task in medical diagnosis detailed in Loveland (1985). A doctor needs to diagnose a patient's disease by performing tests. Given an *a priori* probability distribution over possible diseases, what sequence of tests should the doctor perform to identify the disease as quickly as possible? Another application is in active learning (e.g. Dasgupta (2005)). Given a set of data points, one wants to learn a classifier that labels the points correctly as positive and negative. There is a set of m possible classifiers which is assumed to contain the true classifier. In the Bayesian setting, which we consider, the true classifier is drawn from some known probability distribution. The goal is to identify the true classifier by querying labels at the minimum number of points in expectation (over the prior distribution). Other applications include entity identification in databases (Chakaravarthy et al. (2011)) and experimental design to choose the most accurate theory among competing candidates (Golovin et al. (2010)).

Despite the considerable literature on the classic ODT problem, an important issue that is not considered is that of unknown or noisy outcomes. In fact, our research was motivated by a dataset involving toxic chemical identification where the outcomes of many hypothesis-test pairs are stated as unknown (see Section 6 for details). While prior work incorporating noise in ODT, for example Golovin et al. (2010), was restricted to settings with very few noisy outcomes, in this paper, we design approximation algorithms for the noisy optimal decision tree problem in full generality.

Specifically, we generalize the ODT problem to allow unknown/noisy entries (denoted by "$*$") in the test-hypothesis matrix M, to obtain the *Optimal Decision Tree with Noise* (ODTN) problem, in which the outcome of each noisy entry in the test-hypothesis matrix M is a random ± 1 value, independent of other noisy entries. More precisely, if the entry $M_{t,h} = *$ (for hypothesis h and test t) and the realized hypothesis is h, then the outcome of t will be a random ± 1 value. We will assume for simplicity that each noisy outcome is ± 1 with uniform probability, though our results extend directly to the case where each noisy outcome has a different probability. We consider the standard *persistent* noise model, where repeating the same test always produces the same outcome. Note that this model is more general than the non-persistent noise (where repeating a noisy test leads to "fresh" independent ± 1 outcomes), since one may create copies of tests and hypotheses to reduce to the persistent noise model.

We consider both non-adaptive policies, where the test sequence is fixed upfront, and adaptive policies, where the test sequence is built incrementally and depends on observed test outcomes. Evidently, adaptive policies perform at least as well as non-adaptive ones. Indeed, there exists instances where the relative gap between the best adaptive and non-adaptive policies is very large (see for example, Dasgupta (2005)). However, non-adaptive policies are very simple to implement, requiring minimal incremental computation, and may be preferred in time-sensitive applications.

In fact, our results hold in a significantly more general setting, where the goal is to cover *stochastic* submodular functions. In the absence of noisy outcomes, the non-adaptive and adaptive versions of this problem were studied by by Azar and Gamzu (2011) and Navidi et al. (2020). Other than the ODT problem, this submodular setting captures a number of applications such as multiple-intent search ranking, decision region determination and correlated knapsack cover: see Navidi et al. (2020) for details. Our work is the first to handle noisy outcomes in all these applications.

1.1. Contributions

We derive most of our results for the ODTN problem as corollaries of a more general problem, Submodular Function Ranking with Noisy Outcomes, which is a natural extension of the Submodular Function Ranking problem, introduced by Azar and Gamzu (2011). We first state our results before formally defining this problem in Section 2.3.

First, we obtain an $O(\log \frac{1}{\varepsilon})$-approximation algorithm (see Theorem 3) for *Non-Adaptive* Submodular Function Ranking with noisy outcomes (SFRN) where ε is a separability parameter of the underlying submodular functions. As a special case, for the ODTN (both adaptive and non-adaptive) problem, we consider submodular functions with separability $\varepsilon = \frac{1}{m}$, so the above result immediately implies an $O(\log m)$-approximation for non-adaptive ODTN. This bound is the best possible (up to constant factors) even in the noiseless case, assuming $P \neq NP$.

As our second contribution, we obtain an $O(\min\{c \log|\Omega|, r\} + \log \frac{m}{\varepsilon})$-approximation (Theorem 7) algorithm for *Adaptive* Submodular Ranking with noisy outcomes (ASRN), which implies an $O(\min\{c, r\} + \log m)$ bound for ODTN by setting $\varepsilon = \frac{1}{m}$, where Ω is the set of random outcomes we may observe when selecting elements. The term $\min\{c \log|\Omega|, r\}$ corresponds to the "noise sparsity" of the instance (see Section 2 for formal definitions). For the ODTN problem, c (resp. r) is the maximum number of noisy outcomes in each column (resp. row) of the test-hypothesis matrix M. In the noiseless case, $c = r = 0$ and our result matches the best approximation ratio for the ODT and the Adaptive Submodular Ranking problem (Navidi et al. (2020)). In the noisy case, our performance guarantee degrades smoothly with the noise sparsity. For example, we obtain a logarithmic approximation ratio (which is the best possible) as long as the number of noisy outcomes in each row or column is at most logarithmic. For ODTN, Golovin et al. (2010) obtained an $O(\log^2 \frac{1}{p_{min}})$-approximation algorithm which is polynomial-time only when $c = O(\log m)$; here $p_{min} \leq \frac{1}{m}$ is the minimum probability of any hypothesis. Our result improves this result in that (i) the running time is polynomial irrespective of the number of noisy outcomes and (ii) the approximation ratio is better by at least one logarithmic factor.

While the above algorithm admits a nice approximation ratio when there are *few* noisy entries in each row or column of M, as our third contribution, we consider the other extreme, when each test has only a few *deterministic* entries (or equivalently, a *large* number of noisy outcomes). Here, we focus on the special case of ODTN. At first sight, higher noise seems to only render the problem more challenging, but somewhat surprisingly, we obtain a much *better* approximation ratio in this regime. Specifically, if the number of noisy outcomes in each test is at *least* $m - O(\sqrt{m})$, we obtain an approximation algorithm whose cost is $O(\log m)$ times the optimum and returns the target hypothesis with high probability. We establish this result by relating the cost to a *Stochastic Set Cover* instance, whose cost lower-bounds that of the ODTN instance.

Finally, we tested our algorithms on synthetic as well as a real dataset (arising in toxic chemical identification). We compared the empirical performance guarantee of our algorithms to an information-theoretic lower bound. The cost of the solution returned by our non-adaptive algorithm is typically within 50% of this lower bound, and typically within 20% for the adaptive algorithm, demonstrating the effective practical performance of our algorithms.

As a final remark, although in this work we will consider uniform distribution for noisy outcomes, our results extend directly to the case where each noisy outcome has a different probability of being ± 1. Suppose that the probability of every noisy outcome is between δ and $1-\delta$. Then our results on ASRN continue to hold, irrespective of δ, and the result for the many-unknowns version holds with a slightly worse $O(\frac{1}{\delta}\log m)$ approximation ratio.

1.2. Related Work

The optimal decision tree problem (without noise) has been extensively studied for several decades: see Garey and Graham (1974), Hyafil and Rivest (1976/77), Loveland (1985), Arkin et al. (1998), Kosaraju et al. (1999), Adler and Heeringa (2008), Chakaravarthy et al. (2009), Gupta et al. (2017). The state-of-the-art result Gupta et al. (2017) is an $O(\log m)$-approximation, for instances with arbitrary probability distribution and costs. Chakaravarthy et al. (2011) also showed that ODT cannot be approximated to a factor better than $\Omega(\log m)$, unless P=NP.

The application of ODT to Bayesian active learning was formalized in Dasgupta (2005). There are also several results on the *statistical complexity* of active learning. e.g. Balcan et al. (2006), Hanneke (2007), Nowak (2009), where the focus is on proving bounds for structured hypothesis classes. In contrast, we consider arbitrary hypothesis classes and obtain *computationally efficient* policies with provable approximation bounds relative to the optimal (instance specific) policy. This approach is similar to that in Dasgupta (2005), Guillory and Bilmes (2009), Golovin and Krause (2011), Golovin et al. (2010), Cicalese et al. (2014), Javdani et al. (2014).

The noisy ODT problem was studied previously in Golovin et al. (2010). Using a connection to adaptive submodularity, Golovin and Krause (2011) obtained an $O(\log^2 \frac{1}{p_{min}})$-approximation algorithm for noisy ODT in the presence of very few noisy outcomes, where $p_{min} \leq \frac{1}{m}$ is the minimum probability of any hypothesis.[*] In particular, the running time of the algorithm in Golovin et al. (2010) is exponential in the number of noisy outcomes per hypothesis, which is polynomial only if this number is at most logarithmic in the number of hypotheses/tests. As noted earlier, our result improves both the running time (it is now polynomial for any number of noisy outcomes) and the approximation ratio. We note that an $O(\log m)$ approximation ratio (still only for very sparse noise) follows from work on the "equivalence class determination" problem by Cicalese et al. (2014). For this setting, our result is also an $O(\log m)$ approximation, but our algorithm is simpler. More importantly, ours is the first result that can handle *any* number of noisy outcomes.

[*] The paper Golovin et al. (2010) states the approximation ratio as $O(\log \frac{1}{p_{min}})$ because it relied on an erroneous claim in Golovin and Krause (2011). The correct approximation ratio, based on Nan and Saligrama (2017), Golovin and Krause (2017), is $O(\log^2 \frac{1}{p_{min}})$.

Other variants of noisy ODT have also been considered, e.g. Naghshvar et al. (2012), Bellala et al. (2011), Chen et al. (2017), where the goal is to identify the correct hypothesis with at least some target probability. The theoretical results in Chen et al. (2017) provide "bicriteria" approximation bounds where the algorithm has a larger error probability than the optimal policy. Our setting is different because we enforce *zero* probability of error.

Many algorithms for ODT (including ours) rely on some underlying submodularity properties. We briefly survey some background results. In the basic Submodular Cover problem, we are given a set of elements and a submodular function f. The goal is to use the minimal number of elements to make the value of f reach certain threshold. Wolsey (1982) first considered this problem and proved that the natural greedy algorithm is a $(1+\ln\frac{1}{\varepsilon})$-approximation algorithm, where ε is the minimal positive marginal increment of the function. As a natural generalization, in the Submodular Function Ranking problem we are given *multiple* submodular functions, and need to *sequentially* select elements so as to minimize the total cover time of those functions. Azar and Gamzu (2011) obtained an $O(\log\frac{1}{\epsilon})$-approximation algorithm for this problem, and Im et al. (2016) extended this result to also handle costs. More recently, Navidi et al. (2020) studied an adaptive version of the submodular ranking problem.

Finally, we note that there is also work on minimizing the *worst-case* (instead of average case) cost in ODT and active learning; see e.g., Moshkov (2010), Saettler et al. (2017), Guillory and Bilmes (2010, 2011). These results are incomparable to ours because we are interested in the average case, i.e. minimizing expected cost.

2. Preliminaries

2.1. Optimal Decision Tree with Noise

In the Optimal Decision Tree with Noise (ODTN) problem, we are given a set of m possible *hypotheses* with a *prior* probability distribution $\{\pi_i\}_{i=1}^m$, from which an unknown hypothesis $\bar{i}$ is drawn. There is also a set $\mathcal{T}$ of n binary *tests*, each test $T \in \mathcal{T}$ associated with a 3-way partition T^+, T^-, T^* of $[m]$, where the outcome of test T is

- positive if $\bar{i} \in T^+$,
- negative if $\bar{i} \in T^-$, and
- positive or negative with probability $\frac{1}{2}$ each if $\bar{i} \in T^*$ (noisy outcomes).

We assume that conditioned on $\bar{i}$, each noisy outcome is independent. The outcomes for all test-hypothesis pairs can be summarized in a $\{1, -1, *\}$-valued $n \times m$ matrix M.

While we know the 3-way partition T^+, T^-, T^* for each test $T \in \mathcal{T}$ upfront, we are *not* aware of the actual outcomes for the noisy test-hypothesis pairs. It is assumed that the realized hypothesis $\bar{i}$ can be uniquely identified by performing all tests, regardless of the outcomes of $\star$-tests. This means that for every pair $i, j \in [m]$ of hypotheses, there is some test $T \in \mathcal{T}$ with $i \in T^+$ and $j \in T^-$ or vice-versa. We show how to relax this "identifiability" assumption in Appendix E. The goal is to perform a sequence of tests to identify hypothesis $\bar{i}$ using the minimum *expected* number of tests, which will be formally defined soon. Note that the expectation is taken over both the prior distribution of $\bar{i}$ and the random outcomes of noisy tests for $\bar{i}$.

Types of Policies. A *non-adaptive* policy is specified by a permutation of tests denoting the order in which they will be tried until identification of the underlying hypothesis. The policy performs tests in this sequence and eliminates incompatible hypotheses until there is a unique compatible hypothesis (which is $\bar{i}$). Note that the number of tests performed under such a policy is still random as it depends on $\bar{i}$ and the outcomes of noisy tests.

An *adaptive* policy chooses tests incrementally, depending on prior test outcomes. The *state* of a policy is a tuple (E, d) where $E \subseteq \mathcal{T}$ is a subset of tests and $d \in \{\pm 1\}^E$ denotes the observed outcomes of the tests in E. An adaptive policy is specified by a mapping $\Phi : 2^{\mathcal{T} \times \{\pm 1\}} \to \mathcal{T}$ from states to tests, where $\Phi(E, d)$ is the next test to perform at state (E, d). Define the (random) cost

$Cost(\Phi)$ of a policy Φ to be the number of tests performed until $\bar{i}$ is uniquely identified, i.e., all other hypotheses have been eliminated. The goal is to find policy Φ with minimum $\mathbb{E}[Cost(\Phi)]$. Again, the expectation is over the prior distribution of $\bar{i}$ as well as the outcomes of noisy tests.

Equivalently, we can view a policy as a *decision tree* with nodes corresponding to states, labels at nodes representing the test performed at that state and branches corresponding to the ±1 outcome at the current test. In particular, a non-adaptive policy is simply a decision tree where all nodes on each level are labelled with the same test.

As the number of states can be exponential, we cannot hope to specify arbitrary adaptive policies. Instead, we want implicit policies Φ, where given *any* state (E, d), the test $\Phi(E, d)$ can be computed *efficiently*. This would imply that the total time taken on any decision path is polynomial. We note that an optimal policy Φ^* can be very complex and the map $\Phi^*(E, d)$ may not be efficiently computable. We will still compare the performance of our (efficient) policy to Φ^*.

Noise Model. In this paper, we consider the *persistent noise* model. That is, repeating a test T with $\bar{i} \in T^*$ always produces the same outcome. An alternative model is non-persistent noise, where each run of test T with $\bar{i} \in T^*$ produces an independent random outcome. The persistent noise model is more appropriate to handle missing data. It also contains the non-persistent noise model as a special case (by introducing multiple tests with identical partitions). The persistent-noise model is also more challenging from an algorithmic point of view.

In fact, our results hold in a substantially more general setting (than ODT), that of covering arbitrary *submodular* functions. In Section 2.2 we first describe this setting in the noiseless case, which is well-understood (prior to our work). Then, in Section 2.3 we describe the setting with noisy outcomes, which is the focus of our paper.

2.2. Adaptive Submodular Ranking (Noiseless Case)

We now review the (non-adaptive and adaptive) Submodular Ranking problems introduced by Azar and Gamzu (2011) and Navidi et al. (2020) respectively.

Submodular Function Ranking. An instance of Submodular Function Ranking (SFR) consists of a ground set of *elements* $[n] := \{1, ..., n\}$ and a collection of monotone submodular functions $\{f_1, ..., f_m\}$, $f_i : 2^{[n]} \to [0,1]$, with $f_i(\emptyset) = 0$ and $f_i([n]) = 1$ for all $i \in [m]$. Each $i \in [m]$ is called a *scenario*. An unknown *target* scenario $\bar{i}$ is drawn from a known distribution $\{\pi_i\}$ over $[m]$.

A solution to SFR is a permutation $\sigma = (\sigma(1), ..., \sigma(n))$ of elements. Given any such permutation, the *cover time* of scenario i is $C(i, \sigma) := \min\{t \mid f_i(\sigma^t) = 1\}$ where $\sigma^t = (\sigma(1), ..., \sigma(t))$ is the t-prefix of permutation σ. In words, the cover time is the earliest time when the value of f_i reaches the unit threshold. The goal is to find a permutation σ of $[n]$ with minimal expected cover time $\mathbb{E}_{\bar{i}}[C(\bar{i}, \sigma)] = \sum_{i \in [m]} \pi_i \cdot C(i, \sigma)$.

The *separability* parameter $\varepsilon > 0$ is defined as minimum positive marginal increment of any function, i.e. $\varepsilon := \min\{f_i(S \cup \{e\}) - f_i(S) > 0 \mid \forall S \subseteq [n], i \in [m], e \in [n]\}$. We will use the following.

Theorem 1 **(Azar and Gamzu (2011))**. *There is an $O(\log \frac{1}{\epsilon})$-approximation algorithm for SFR.*

Adaptive Submodular Ranking. In the Adaptive Submodular Ranking (ASR) problem, in addition to the above input to SFR, for each scenario $i \in [m]$ we are given a *response function* $r_i : [n] \to \Omega$ where Ω is a finite set of *outcomes* (or response, which we use interchangeably). A solution to ASR is an *adaptive* sequence of elements: the sequence is adaptive because it can depend on the outcomes from previous elements. When the policy selects an element $e \in [n]$, it receives an outcome $o = r_{\bar{i}}(e) \in \Omega$, thereby any scenario i with $r_i(e) \neq \bar{o}$ can be ruled out.

The *state* of an adaptive policy is a tuple (E, d) where $E \subseteq [n]$ is the subset of previously selected elements and $d \in \Omega^E$ denotes the observed responses on E. An adaptive policy is then specified by a mapping $\Phi : 2^{[n]\times\Omega} \to [n]$ from states to elements, where $\Phi(E,d)$ is the next element to select at state (E,d). Note that any adaptive policy Φ induces, for each scenario i, a unique sequence σ_i of elements that will be selected if the target scenario $\bar{i} = i$. The *cover time* of i is defined as $C(i,\Phi) := \min\{t \; | f_i(\sigma_i^t) = 1\}$. The goal is to find a policy Φ with minimal expected cover time $\sum_{i\in[m]} \pi_i \cdot C(i,\Phi)$. We will use the following result in Section 4.

Theorem 2 **(Navidi et al. (2020))**. *There is an $O(\log \frac{m}{\epsilon})$-approximation algorithm for ASR.*

As discussed in Navidi et al. (2020), the optimal decision tree problem (without noise) is a special case of ASR. We show later that even the noisy version ODTN can be reduced to a *noisy* variant of ASR (which we define next).

2.3. Adaptive Submodular Ranking with Noise

In this paper, we introduce a new variant of ASR by incorporating noisy outcomes, which generalizes the ODTN problem.

ASR with Noise. An instance of the Adaptive Submodular Ranking with Noise (ASRN) Problem consists of a ground set of elements $[n]$, a finite set Ω of *outcomes*, and a collection of monotone submodular functions $\{f_1,...,f_m\}$, where each $f_i : 2^{[n]\times\Omega} \to [0,1]$ satisfies $f_i(\emptyset) = 0$ and $f_i([n]\times\Omega) = 1$. Note that the groundset of each function f_i is $[n]\times\Omega$, i.e., all element-outcome pairs. As before, each $i \in [m]$ is called a scenario and an unknown target scenario $\bar{i}$ is drawn from a given distribution $\{\pi_i\}_{i=1}^m$. For each scenario $i \in [m]$, we are given a *response function* $r_i : [n] \to \Omega \cup \{*\}$. When an element e is selected, its outcome is:

- $r_i(e)$ if $r_i(e) \in \Omega$, and
- a uniformly random response from Ω if $r_i(e) = *$ (noisy outcome).

The responses can be summarized in an $n \times m$ matrix M with entries from $\Omega \cup \{*\}$. Conditioned on $\bar{i}$, we assume that all noisy outcomes are independent. Our results extend to arbitrary distributions for noisy outcomes, but we will work with the uniform case for simplicity.

As in the noiseless case, the state of a policy is the tuple (E, d) where $E \subseteq [n]$ denotes the previously selected elements and $d \in \Omega^E$ denotes their observed responses. A *non-adaptive* policy is simply given by a permutation of all elements and involves selecting elements in this (static) sequence. An *adaptive* policy is a mapping $\Phi : 2^{[n]\times\Omega} \to [n]$, where $\Phi(E, d)$ is the next element to select at state (E, d). Scenario i is said to be *covered* in state (E, d) if $f_i(\{(e, d_e) : e \in E\}) = 1$, i.e., function f_i is covered by the element-response pairs observed so far. The goal is to cover the target scenario $\bar{i}$ using the minimum expected number of elements.

Unlike the noiseless case, in ASRN, each scenario i may trace *multiple* paths in the decision tree corresponding to policy Φ. However, if we condition on the responses $\omega \in \Omega^n$ from all elements, each scenario i traces a unique path, corresponding to a sequence $\sigma_{i,\omega}$ of element-response pairs. The *cover time* of scenario i under ω is defined as $C(i, \Phi|\omega) := \min\{t | f_i(\sigma^t_{i,\omega}) = 1\}$ where $\sigma^t_{i,\omega}$ consists of the first t element-response pairs in $\sigma_{i,\omega}$. The expected cover time of scenario i is $\mathrm{ECT}(i, \Phi) := \sum_{\omega\in\Omega^n} \Pr(\omega|i) \cdot C(i, \Phi|\omega)$, where $\Pr(\omega|i)$ is the probability of observing responses ω conditioned on $\bar{i} = i$. Finally, the expected cost of policy Φ is $\sum_{i\in[m]} \pi_i \cdot \mathrm{ECT}(i, \Phi)$.

For each scenario i, we assume that the function f_i can always be covered irrespective of the noisy outcomes (when $\bar{i} = i$). In other words, for any $i \in [m]$ and $\omega \in \Omega^n$ that is *consistent* with scenario i (i.e., $\omega_e = r_i(e)$ for each e with $r_i(e) \neq *$), we must have $f_i(\{(e, \omega_e) : e \in [n]\}) = 1$. In the absence of this assumption, the optimal value (as defined above) will be unbounded.

Connection to ODTN. The ODTN problem can be cast as a special case of the ASRN problem, where the n tests $\mathcal{T}$ in ODTN corresponds to the elements $[n]$ in ASRN, and the m hypotheses in ODTN correspond to the scenarios in ASRN, with the same prior distribution. The outcomes

$\Omega = \{\pm 1\}$. Define the response function for each test $T \in \mathcal{T}$ as follows. Let (T^+, T^-, T^*) be the 3-way partition of $[m]$ for test T. For any hypothesis (scenario) $i \in [m]$, define $r_i(T) = o$ if $i \in T^o$ for each $o \in \Omega \cup \{*\}$. For any $i \in [m]$, define the submodular function

$$f_i(S) = \frac{1}{m-1} \cdot \Big| \bigcup_{T:(T,+1)\in S} T^- \bigcup \bigcup_{T:(T,-1)\in S} T^+ \Big|, \quad \forall S \subseteq \mathcal{T} \times \{+1, -1\}.$$

Note that the element-outcome pairs here are $U = \mathcal{T} \times \{+1, -1\}$. It is easy to see that each function $f_i : 2^U \to [0,1]$ is monotone and submodular. Also, these functions f_i happen to be uniform for all i. Moreover, the separability parameter $\varepsilon = \frac{1}{m-1}$. Crucially, $f_i(S)$ corresponds to the fraction of hypotheses (other than i) that are incompatible with at least one outcome in S: for example, if S has a positive outcome $(T, +1)$ then hypotheses T^- are incompatible (similarly for negative outcomes). So f_i has value one exactly when i is identified as the only compatible hypothesis. By the assumption that the target hypothesis can be uniquely identified, the function f_i can be covered (i.e. reaches value one) irrespective of the noisy outcomes.

2.4. Expanded Scenario Set

In our analysis for both the non-adaptive and adaptive ASRN problem, we will consider an equivalent noiseless ASR instance. Let $\mathcal{I}$ be a given ASRN instance with scenarios $[m]$. The ASR instance $\mathcal{J}$ considers an expanded set of scenarios. For any scenario $i \in [m]$, define

$$\Omega(i) := \{\omega \in \Omega^n : \omega_e = r_i(e) \text{ for all } e \in [n] \text{ with } r_i(e) \neq *\},$$

denoting all outcome vectors that are *consistent* with scenario i. For any $\omega \in \Omega(i)$, the *expanded* scenario (i, ω) corresponds to the original scenario $i \in [m]$ when the outcome of each element e is ω_e. Note that an expanded scenario also fixes all noisy outcomes. We write $H_i := \{(i, \omega) : \omega \in \Omega(i)\}$ and $H = \cup_{i=1}^m H_i$ for the set of all expanded scenarios.

To define the prior distribution in the ASR instance, let $c_i = |\{e \in [n] : r_i(e) = *\}|$ be the number of noisy outcomes for $i \in [m]$. Since the outcome of any $\star$-element for i is uniformly drawn from Ω, each of the $|\Omega|^{c_i}$ possible expanded scenarios for i occurs with the same probability $\pi_{i,\omega} = \pi_i/|\Omega|^{c_i}$.

To complete the reduction, for each $(i,\omega) \in H$, we define the response function

$$r_{i,\omega} : [n] \to \Omega, \quad r_{i,\omega}(e) = \omega_e, \qquad \forall e \in [n],$$

and the submodular coverage function

$$f_{i,\omega} : 2^{[n]} \to [0,1], \quad f_{i,\omega}(S) = f_i\big(\{(e,\omega_e) : e \in S\}\big), \qquad \forall S \subseteq [n].$$

By this definition, since f_i is monotone and submodular on $[n] \times \Omega$, the function $f_{i,\omega}$ is also monotone and submodular on $[n]$. We will also work with the ASR (noiseless) instance on the expanded scenarios with response functions $r_{i,\omega}$ and submodular functions $f_{i,\omega}$. In Appendix A, we will formally establish the following reduction.

PROPOSITION 1. *The ASRN instance $\mathcal{I}$ is equivalent to the ASR instance $\mathcal{J}$.*

Crucially, the number of expanded scenarios $|H|$ is exponentially large as $|H| \leq \sum_{i\in[m]} |\Omega|^{c_i}$. So we cannot merely apply existing algorithms for the noiseless ASR problems. In §3 and §4 we will show different ways for managing the expanded scenarios and obtaining *polynomial time* algorithms.

3. Nonadaptive Algorithm

This main result in this section is an $O(\log \frac{1}{\varepsilon})$-approximation for Non-Adaptive Submodular Function Ranking (SFRN) where $\varepsilon > 0$ is the separability parameter of the submodular functions. By Proposition 1, the SFRN problem is equivalent to the SFR problem on the expanded scenarios. However, as noted above, we cannot use Theorem 1 directly as the SFR instance has an exponential number of scenarios. Nevertheless, we can obtain the following result.

Theorem 3. *There is a poly($\frac{1}{\varepsilon}, n, m$) time $O(\log \frac{1}{\varepsilon})$-approximation for the SFRN problem.*

Observe that for ODTN, $\varepsilon = \frac{1}{m-1}$, thus we obtain the following result for ODTN.

Corollary 1. *There is an $O(\log m)$-approximation for non-adaptive ODTN.*

High Level Ideas. The algorithm of Azar and Gamzu (2011) for SFR is a greedy-style algorithm that at any iteration, having already chosen elements E, assigns to each $e \in [n] \setminus E$ a score that measures the *coverage gain* when it is selected, defined as

$$G_E(e) := \sum_{(i,\omega)\in H: f_{i,\omega}(E)<1} \pi_{i,\omega} \frac{f_{i,\omega}(\{e\}\cup E) - f_{i,\omega}(E)}{1 - f_{i,\omega}(E)} = \sum_{(i,\omega)\in H} \pi_{i,\omega} \cdot \Delta_E(i,\omega;e), \tag{1}$$

$$\Delta_E(i,\omega,e) = \begin{cases} \frac{f_{i,\omega}(\{e\}\cup E) - f_{i,\omega}(E)}{1 - f_{i,\omega}(E)}, & \text{if } f_{i,\omega}(E) < 1; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The algorithm then selects the element with the maximum score.

Since this summation involves exponentially many terms, we do not know how to compute the *exact* value of (1) in polynomial time. However, using the fact that $G_E(e)$ is the *expectation* of $\Delta_E(i,\omega;e)$ over the expanded scenarios $(i,\omega) \in H$, we will show how to obtain a randomized constant-approximate maximizer by sampling from H. Moreover, we use the following extension of Theorem 1, which follows directly from the analysis in Im et al. (2016).

Theorem 4 **(Azar and Gamzu (2011), Im et al. (2016))**. *Consider the SFR algorithm that selects at each step, an element e with $G_E(e) \geq \Omega(1) \cdot \max_{e' \in U} G_E(e')$. This is an $O(\log \frac{1}{\epsilon})$-approximation algorithm.*

Consequently, if we always find an approximate maximizer for $G_E(e)$ by sampling then Theorem 3 would follow from Theorem 4. However, this sampling approach is not sufficient because it can fail when the value $G_E(e)$ is very small. In order to deal with this, a key observation is that when the

Algorithm 1 Non-adaptive SFRN algorithm.

```
Initialize $E \leftarrow \emptyset$ and sequence $\sigma = \emptyset$.
while $E \neq [n]$ do                                                    ▷ Phase 1 begins
    For each $e \in [n]$, compute an estimate $\overline{G_E}(e)$ of the score $G_E(e)$ by sampling from $H$ independently $N = m^3 n^4 \varepsilon^{-1}$ times.
    Let $e^*$ denote the element $e \in [n] \setminus E$ that maximizes $\overline{G_E}(e)$.
    if $\overline{G_E}(e) \geq \frac{1}{4} m^{-2} n^{-4} \varepsilon$ then
        Update $E \leftarrow E \cup \{e^*\}$ and append $e^*$ to sequence $\sigma$.
    else
        Exit the while loop.                                             ▷ Phase 1 ends
Append the elements in $[n] \setminus E$ to sequence $\sigma$ in arbitrary order.    ▷ Phase 2
Output non-adaptive sequence $\sigma$.
```

score $G_E(e)$ is small *for all* elements e, then it must be that (with high probability) the already-selected elements E have covered $\bar{i}$, so any future elements would not affect the expected cover time. The formal analysis is given in Appendix B.

4. Adaptive Algorithms

In this section we present the $O\left(\log \frac{m}{\varepsilon} + \min\{c \log |\Omega|, r\}\right)$-approximation for ASRN where we recall that c, r are the maximum number of noisy entries ("stars") per column and per row in the outcome matrix M, and ε is the separability parameter of the submodular functions. We propose two algorithms, achieving $O\left(r + \log \frac{m}{\varepsilon}\right)$ and $O\left(c \log |\Omega| + \log \frac{m}{\varepsilon}\right)$ approximations respectively, which combined imply our main result.

In both algorithms, we maintain the posterior probability of each scenario based on the previous element responses, and use these probabilities to calculate a *score* for each element, which comprises (i) a term that prioritizes splitting the candidate scenarios in a balanced manner and (ii) terms corresponding to the expected number of scenarios eliminated. Different than the noiseless setting, in ASRN (and ODTN), each scenario may trace *multiple* paths in the decision tree due to outcome randomness. In fact, each scenario may trace an exponential number of paths in the tree, so a naive generalization of the analysis in Navidi et al. (2020) incurs an extra exponential factor in the approximation ratio.

We circumvent this challenge by reducing to an ASR instance $\mathcal{J}$ (as defined in Proposition 1) using the *expanded* scenarios. In this way, the noise is removed, since we recall that the outcome of each element is deterministic *conditional* on any expanded scenario (i,ω). Our first result, an $O(c\log|\Omega|+\log\frac{m}{\varepsilon})$-approximation, then follows from Navidi et al. (2020).

However, as $\mathcal{J}$ involves exponentially many scenarios, a naive implementation of the algorithm in Navidi et al. (2020) leads to exponential running time. To improve the computational efficiency, in Section 4.1 we exploit the special structure of $\mathcal{J}$ and devise a polynomial time algorithm. Then, in Section 4.2, we propose a slightly different algorithm than that of Navidi et al. (2020), and show an $O(r+\log\frac{m}{\varepsilon})$ approximation ratio.

4.1. An $O(c\log|\Omega|+\log\frac{m}{\varepsilon})$-Approximation Algorithm

Our first adaptive algorithm is based on the $O(\log\frac{m}{\varepsilon})$-approximation algorithm for ASR from Navidi et al. (2020), formally stated as Algorithm 2. Applying this result on the instance $\mathcal{J}$ and recalling $|H|\le|\Omega|^c\cdot m$, we immediately obtain the desired guarantee. Their algorithm, rephrased in our notations, maintains the set $H'\subseteq H$ of all expanded scenarios that are *consistent* with all the observed outcomes, and iteratively selects the element with maximum score, as defined in (3)[‡].

As the heart of the algorithm, this score strikes a balance between *covering* the submodular functions of the consistent scenarios and *shrinking* H' hence reducing the uncertainty in the target

[‡] We use the subscript c to distinguish from the score function Score_r considered in Section 4.2, but for ease of notation, we will suppress the subscript in this subsection.

scenario. The second term in Score_c, similar to the score in our non-adaptive algorithm (Algorithm 1), involves the sum of the incremental coverage (for selecting e) over all uncovered expanded scenarios, weighted by their current coverage, with higher weights on the expanded scenarios closer to being covered.

To interpret the first term in Score_c, let us for simplicity assume $\Omega = \{\pm 1\}$ and $\pi_{i,\omega}$ is uniform over H. Upon selecting an element, H' is split into two subsets, among which $L_e(H')$ is the lighter (in cardinality), or equivalently – since we just assumed $\pi_{i,\omega}$ to be uniform – in the total prior probabilities. Thus, this term is simply the number of expanded scenarios eliminated in the worst case (over the outcomes in Ω). This is reminiscent of the greedy algorithm for the ODT problem (e.g. Kosaraju et al. (1999)) which iteratively selects a test that maximizes the number of scenarios ruled out, in the *worst* case over all test outcomes. Evidently, the higher this term, the more progress is made towards identifying the target (expanded) scenario.

Algorithm 2 Algorithm for ASR instance $\mathcal{J}$, based on Navidi et al. (2020).

```
Initialize $E \leftarrow \emptyset, H' \leftarrow H$.
while $H' \neq \emptyset$ do
    For any element $e \in [n]$, let $B_e(H')$ be the largest *cardinality* set among
        $\{(i,\omega) \in H' : r_{i,\omega}(e) = o\} \qquad \forall o \in \Omega$
    Define $L_e(H') = H' \setminus B_e(H')$
    Select the element $e \in [n] \setminus E$ maximizing
        $\text{Score}_c(e, E, H') = \pi\big(L_e(H')\big) + \sum_{(i,\omega)\in H', f_{i,\omega}(E)<1} \pi_{i,\omega} \cdot \frac{f_{i,\omega}(e \cup E) - f_{i,\omega}(E)}{1 - f_{i,\omega}(E)}$   (3)
    Observe response $o$ and update $H'$ as $H' \leftarrow \{(i,\omega) \in H' : \omega_e = o$ and $f_{i,\omega}(E \cup e) < 1\}$
    $E \leftarrow E \cup \{e\}$
```

As noted earlier, a key issue is the exponential size of the expanded scenario set H. The naive implementation, which computes the summation in Score_c by evaluating each term in H', requires exponential time. Nonetheless, as the main focus of this subsection, we explain how to utilize the structure of the ASRN instance $\mathcal{J}$ to reformulate each of the two terms in Score_c in a manageable form, hence enabling a polynomial time implementation.

Computing the First Term in Score_c**.** Recall that H_i is the set of all expanded scenarios for i. Since each $(i,\omega) \in H_i$ is has an equal share $\pi_{i,\omega} = |\Omega|^{-c_i}\pi_i$ of prior probability mass the (original) scenario $i \in [m]$, computing the first term in $Score_c$ reduces to maintaining the *number* $n_i = |H_i \cap H'|$ of consistent copies of i. We observe that n_i can be easily updated in each iteration. In fact, suppose outcome $o \in \Omega$ is observed upon selecting element e. We consider how $H' \cap H_i$ changes after selecting in the following three cases.

1. if $r_i(e) \notin \{\star, o\}$, then none of i's expanded scenarios would remain in H', so n_i becomes 0,
2. if $r_i(e) = o$, then all of i's expanded scenarios would remain in H', so n_i remains the same,
3. if $r_i(e) = \star$, then only those (i,ω) with $\omega(e) = o$ will remain, and so n_i shrinks by an $|\Omega|$ factor.

As n_i's can be easily updated, we are also able to compute the first term in Score_c efficiently. Indeed, for any element e (that is not yet selected), we can implicitly describe the set $L_e(H')$ as follows. Note that for any outcome $o \in \Omega$,

$$|\{(i,\omega) \in H' : r_{i,\omega}(e) = o\}| = \sum_{i\in[m]:r_i(e)=o} n_i + \frac{1}{|\Omega|}\sum_{i\in[m]:r_i(e)=\star} n_i,$$

so the largest cardinality set $B_e(H')$ can then be easily determined using n_i's. In fact, let b be the outcome corresponding to $B_e(H')$. Then,

$$\pi\left(L_e\left(H'\right)\right) = \sum_{i\in[m]:r_i(e)\notin\{b,\star\}} \frac{\pi_i}{|\Omega|^{c_i}} \cdot n_i + \frac{|\Omega|-1}{|\Omega|}\sum_{i\in[m]:r_i(e)=\star} \frac{\pi_i}{|\Omega|^{c_i}} \cdot n_i.$$

Computing the Second Term in Score_c**.** The second term in Score_c involves summing over exponentially many terms, so a naive implementation is inefficient. Instead, we will rewrite this summation as an *expectation* that can be calculated in polynomial time.

We introduce some notations before formally stating this equivalence. Suppose the algorithm selected a subset E of elements, and observed outcomes $\{\nu_e\}_{e\in E}$. We overload notation slightly and use $f(\nu_E) := f\big(\{(e,\nu_e) : e \in E\}\big)$ for any function f defined on $2^{[m]\times\Omega}$. For each scenario $i \in [m]$, let $p_i = n_i \cdot \frac{\pi_i}{|\Omega|^{c_i}}$ be the total probability mass of the surviving expanded scenarios for i.[†] Finally, for any element e and scenario i, let $\mathbb{E}_{i,\nu_e}$ be the expectation over the outcome ν_e of element e conditional on i being the realized scenario. We can then rewrite the second term in Score_c as follows.

Lemma 1. *For each $i \in [m]$, and $e \notin E$,*

$$\sum_{(i,\omega)\in H'} \pi_{i,\omega} \cdot \frac{f_{i,\omega}(e\cup E) - f_{i,\omega}(E)}{1 - f_{i,\omega}(E)} = \sum_{i\in[m]} p_i \cdot \frac{\mathbb{E}_{i,\nu_e}[f_i(\nu_E \cup \{\nu_e\}) - f_i(\nu_E)]}{1 - f_i(\nu_E)} \tag{4}$$

This lemma suggests the following efficient implementation of Algorithm 2. For each i, compute and maintain p_i using n_i. To find the expectation in the numerator, note that if $r_i(e) \neq \star$, then ν_e is deterministic and hence it is straightforward to find this expectation. In the other case, if $r_i(e) = \star$, recalling that the outcome is uniform over Ω, we may simply evaluate $f_i(\nu_E \cup \{(e,o)\}) - f_i(\nu_E)$ for each $o \in \Omega$ and take the average, since the noisy outcome is uniformly distributed over Ω.

Now we are ready to formally state and prove the main result of this subsection.

Theorem 5. *Algorithm 2 is an $O(c\log|\Omega| + \log m + \log\frac{1}{\varepsilon})$-approximation algorithm for ASRN where c is the maximum number of noisy outcomes in each column of the response matrix M.*

Proof. Consider the ASR instance $\mathcal{J}$ and Algorithm 2. As discussed above, this algorithm can be implemented in polynomial time. By Theorem 2, this algorithm has an $O\big(\log(|\Omega|^c m) + \log\frac{m}{\varepsilon}\big) = O(c\log|\Omega| + \log\frac{m}{\varepsilon})$ approximation ratio since $|H| \le |\Omega|^c \cdot m$. □

Algorithm 3 Modified algorithm for ASR instance $\mathcal{J}$.

```
Initialize $E \leftarrow \emptyset, H' \leftarrow H$
**while** $H' \neq \emptyset$ **do**
    $S \leftarrow \{i \in [m] : H_i \cap H' \neq \emptyset\}$                ▷ Consistent original scenarios
    For $e \in [n]$, let $U_e(S) = \{i \in S : r_i(e) = *\}$ and $C_e(S)$ be the largest cardinality set among
        $\{i \in S : r_i(e) = o\}, \quad \forall o \in \Omega,$
    and let $o_e(S) \in \Omega$ be the outcome corresponding to $C_e(S)$.
    For each $e \in [n]$, let
        $\overline{R_e}(H') = \{(i,\omega) \in H' : i \in C_e(S)\} \bigcup \{(j, o_e(S)) \in H' : j \in U_e(S)\},$
    be those expanded-scenarios that have outcome $o_e(S)$ for element $e$, and $R_e(H') := H' \setminus \overline{R_e}(H')$.
    Select element $e \in [n] \setminus E$ that maximizes
        $\text{Score}_r(e, E, H') = \pi\big(R_e(H')\big) \; + \sum_{(i,\omega) \in H', f_{i,\omega}(E) < 1} \pi_{i,\omega} \cdot \frac{f_{i,\omega}(e \cup E) - f_{i,\omega}(E)}{1 - f_{i,\omega}(E)}$     (5)
    Observe outcome $o$
    $H' \leftarrow \{(i,\omega) \in H' : r_{i,\omega}(e) = o \text{ and } f_{i,\omega}(E \cup e) < 1\}$     ▷ Update the (expanded) scenarios
    $E \leftarrow E \cup \{e\}$
```

4.2. An $O(r + \log \frac{m}{\varepsilon})$-Approximation Algorithm

In this section, we consider a slightly different score function, Score_r, and obtain an $O(r + \log \frac{m}{\varepsilon})$-approximation. Unlike the previous section where the approximation factor follows as an immediately corollary from Theorem 2, to prove this result, we need to also modify the analysis.

[†] One may easily verify via the Bayesian rule that $p_i/p([m])$ is indeed the posterior probability of scenario $i \in [m]$, given the previously observed outcomes.

The only difference from Algorithm 2 is in the first term of the score function. Recall that in Score_c, upon selecting an element, the surviving expanded scenarios is partitioned into $|\Omega|$ subsets, among which $L_e(H')$ is defined to be the lightest cardinality. Its counterpart in Score_r, however, is defined more indirectly, by first considering the *original* scenarios. The element e partitions the original scenarios with *deterministic* outcomes into $|\Omega|$ subsets, with the largest (in cardinality) being $C_e(S) \subseteq [m]$. The set $R_e(H') \subseteq H'$ is then defined to be the consistent expanded scenarios that have a different outcome than $C_e(S)$.

Computational Complexity. By definition, S can be directly computed using the n_i's, which can be updated in polynomial time as explained in Section 4.1. Similar to Algorithm 2, the second term here also involves summing over exponentially many terms, but by following the same recipe as in Section 4.1, one may also implement it in polynomial time.

The main result of this section, stated below, is proved by adapting the proof technique from Navidi et al. (2020). The proof appears in Appendix C.3.

THEOREM 6. *Algorithm 3 is a polynomial-time $O(r + \log \frac{m}{\varepsilon})$-approximation algorithm for ASRN, where r is the maximum number of noisy outcomes in any row of the response matrix M.*

Combining the above result with Theorem 5 and selecting the one with lower approximation ratio between Algorithm 2 and Algorithm 3, we immediately obtain the following.

THEOREM 7. *There is an adaptive $O\big(\min\{c \log |\Omega|, r\} + \log \frac{m}{\varepsilon})$-approximation algorithm for the ASRN problem.*

When applied to the ODTN problem, this implies an $O\big(\min\{c, r\} + \log \frac{m}{\varepsilon})$-approximation algorithm. In Appendix C.2, we also provide closed-form expressions for the scores used in Algorithms 3 and 2 in the special case of ODTN: this is also used in our computational results.

5. ODTN with Many Unknowns

Our adaptive algorithm in Section 4 has a performance guarantee that grows with the noise *sparsity* $\min(r, c\log|\Omega|)$. In this section, we consider the special case of ODTN (which is our primary application) and focus on instances with a large number of noisy outcomes. We show that an $O(\log m)$-approximation algorithm can be achieved even in this regime.

An ODTN instance is called *α-sparse* $(0 \le \alpha \le 1)$ if there $\max\{|T^+|, |T^-|\} \le m^\alpha$ for all tests $T \in \mathcal{T}$. In particular, when $\alpha < 1$, this means the vast majority of entries are noisy in every test. Our main result is the following.

Theorem 8. *There is a polynomial time adaptive algorithm whose cost is $O(\log m)$ times the optimum for ODTN on any α-sparse instance with $\alpha \le \frac{1}{2}$, and returns the true hypothesis with probability $1 - m^{-1}$.*

Moreover, by repeating the algorithm for $c \ge 1$ times, the error probability will decrease to m^{-c}.

5.1. Main Idea and the Stochastic Set Cover Problem

Stochastic Set Cover. The design and analysis of our algorithm are both closely related to that of the *Stochastic Set Cover* (SSC) problem (Liu et al. (2008), Im et al. (2016)). An instance of SSC consists of a *ground set* $[m]$ of *items* and a collection of *random* subsets $S_1, \cdots, S_n$ of $[m]$, where the distribution of each S_i is known to the algorithm. The instantiation of each set is only known after it is selected. The goal is to find an adaptive policy that minimizes the expected number of sets to cover all elements in the ground set.

The following natural adaptive greedy algorithm is known to be an $O(\log m)$-approximation (Liu et al. (2008), Im et al. (2016)). Suppose at some iteration, $A \subseteq [m]$ is the set of uncovered elements. A random set S is said to be *β-greedy* if its expected coverage of the uncovered elements is at least $1/\beta$ the maximum, i.e.

$$\mathbb{E}\big[|S \cap A|\big] \ge \frac{1}{\beta} \max_{j \in [n]} \mathbb{E}\big[|S_j \cap A|\big].$$

An SSC algorithm is (β,ρ)-*greedy* if for every $t \geq 1$, the algorithm picks a β-greedy set in no less than t/ρ iterations among the first t. By slightly modifying the analysis in Im et al. (2016), one may obtain the following guarantee which will serve as the cornerstone of our analysis.

Theorem 9 **(Im et al. (2016))**. *For any stochastic set cover instance, a (β,ρ)-greedy policy costs at most $O(\beta\rho\log m)$ times the optimum.*

Relating ODTN Optimum and SSC: A Lower Bound. We now derive a lower bound on the ODTN optimum, in terms of the optima of SSC instances constructed as follows. For any hypothesis $i \in [m]$, let SSC(i) denote the stochastic set cover instance with ground set $[m]\setminus\{i\}$ and n random sets, given by

$$S_T(i) = \begin{cases} T^+ \text{ with prob. } 1 & \text{if } i \in T^- \\ T^- \text{ with prob. } 1 & \text{if } i \in T^+ \\ T^- \text{ or } T^+ \text{ with prob. } \frac{1}{2} \text{ each} & \text{if } i \in T^* \end{cases}, \qquad \forall T \in [n].$$

To see the connection between SSC and ODTN, observe that when i is the target hypothesis in the ODTN instance, any feasible algorithm must identify i by *eliminating* all other hypotheses which, in the language of SSC, translates to *covering* all items in $[m]\backslash\{i\}$. This leads to the following key lower bound that our algorithm exploits.

Lemma 2. $\mathrm{OPT} \geq \sum_{i\in[m]} \pi_i \cdot \mathrm{OPT}_{\mathrm{SSC}(i)}$.

We now explain why "good" progress made in SSC(i) also leads to "good" progress in ODTN. Consider a hypothesis i and a test T with $i \in T^*$, and let A be the set of consistent hypotheses. When test T is selected, the expected coverage of the corresponding (random) set $S_T(i)$ in SSC(i) is $\frac{1}{2}(|T^+\cap A| + |T^-\cap A|)$. The following result shows that if T maximizes $\frac{1}{2}(|T^+\cap A| + |T^-\cap A|)$, then it is 2-greedy for SSC(i).

Lemma 3. *Let T be a test that maximizes $\frac{1}{2}(|T^+\cap A| + |T^-\cap A|)$. Then for any $i \in T^*$,*

$$\frac{1}{2}\left(|T^+\cap A| + |T^-\cap A|\right) = \mathbb{E}\left[|S_T(i)\cap(A\backslash i)|\right] \geq \frac{1}{2}\cdot\max_{T'\in[n]}\mathbb{E}\left[|S_{T'}(i)\cap(A\backslash i)|\right].$$

Hence, by our sparsity assumption, since the vast majority of hypotheses are in T^*, such a test T is 2-greedy for most SSC instances. This motivates the following greedy algorithm. When A is the set of consistent hypotheses, pick test T that maximizes $\frac{1}{2}|T^+ \cap A| + \frac{1}{2}|T^- \cap A|$. Suppose the following *ideal condition* holds. At each iteration t (when t tests have been selected), for *every* hypothesis i, the algorithm has selected *at least* t/ρ tests that are $\star$-tests for i. Then, the sequence of tests selected is $(2,\rho)$-greedy for *every* i, hence making nearly-optimal progress in *every* instance SSC(i). Therefore by Theorem 9, the expected cost of this algorithm under i is $O(\rho \log m) \cdot \mathrm{OPT}_{\mathrm{SSC}}(i)$. Taking expectation over the target hypothesis i and combining with Lemma 2, it then follows that this algorithm is an $O(\rho \log m)$-approximation to ODTN.

However, in general, the ideal condition assumed above may not hold. In other words, up until some point, the sequence of tests selected is no longer $(2,\rho)$-greedy for some hypothesis i. To handle this issue, we modify the above greedy algorithm at all *power-of-two* iterations as follows (see Section 5.3). At each $t = 2^k$ where $k = 1, 2, \ldots \log m$, we consider the set Z of $O(m^\alpha)$ hypotheses with the fewest $\star$-tests selected thus far. Then, we invoke a *membership oracle* Member(Z), to check whether the target hypothesis $\bar{i} \in Z$ (see Section 5.2). If so, then the algorithm halts and returns $\bar{i}$. Otherwise, it continues with the greedy algorithm until the next power-of-two iteration. We will show that the membership oracle only incurs cost $O(m^\alpha)$, which can be bounded using the following lower bound.

Lemma 4. *The optimal value* $\mathrm{OPT} \geq \Omega(m^{1-\alpha})$ *for any* α*-sparse instance.*

In particular, when $\alpha < \frac{1}{2}$ the above implies that the cost $O(m^\alpha)$ for each call of the membership oracle is lower than OPT, and hence the total cost incurred at power-of-two steps is $O(\log m \cdot \mathrm{OPT})$.

5.2. Overview of the Membership Oracle

The *membership oracle* Member(Z) takes a (small) subset $Z \subseteq [m]$ as input, and decides whether the target hypothesis $\bar{i} \in Z$. At a high level, Member(Z) works as follows. Whenever $|Z| \geq 2$, we pick an arbitrary pair (j,k) of hypotheses in Z and let them "duel" (i.e. choose a test T with $M_{T,j} = -M_{T,k}$) until there is only a unique *survivor* i.

Let $i \in [m]$ be an arbitrary hypothesis. We show that if $\bar{i} \neq i$ then with high probability we can rule out i using very few tests. In fact, we first select an arbitrary set W of $4\log m$ deterministic tests for i, and let Y be the set of consistent hypotheses after performing these tests. Without loss of generality, we assume $i \in T^+$ for all $T \in W$. There are three cases:

- **Trivial Case:** if $\bar{i} \in T^-$ for *some* $T \in W$, then we rule out i when any test T is performed.
- **Good Case:** if $\bar{i} \in T^*$ for more than half of the tests T in W, then by Chernoff's inequality, with high probability we observe at least one "-", hence ruling out i.
- **Bad Case:** if $\bar{i} \in T^+$ for less than half of the tests T in W, then concentration bounds can not ensure a high enough probability for ruling out i. In this case, we let each hypothesis in Y *duel* with i until either i loses a duel or wins all the duels. This takes $|Z|-1$ iterations.

We formalize the above ideas in the Algorithm 5 (Appendix D.1), and prove bound the cost of Member(Z) as follows.

Lemma 5. *If* $\bar{i} \in Z$*, then* Member(Z) *declares* $\bar{i} = i$ *with probability one; otherwise, it declares* $\bar{i} \notin Z$ *with probability at least* $1 - m^{-2}$*. Moreover, the expected cost of* Member(Z) *is* $O(|Z| + \log m)$*.*

5.3. The Main Algorithm

The overall algorithm is given in Algorithm 4. The algorithm maintains a subset of consistent hypotheses, and iteratively computes the greediest test, as formally specified in Step 7. At each $t = 2^k$ where $k = 1, 2, \ldots \log m$, we invoke the membership oracle.

Truncated Decision Tree. Let $\mathbb{T}$ denote the decision tree corresponding to our algorithm. We only consider tests that correspond to step 7. Recall that H is the set of *expanded* hypotheses and that any expanded hypothesis traces a unique path in $\mathbb{T}$. For any $(i,\omega) \in H$, let $P_{i,\omega}$ denote this path traced; so $|P_{i,\omega}|$ is the number of tests performed in Step 7 under (i,ω). We will work with a truncated decision tree $\overline{\mathbb{T}}$, defined below.

Algorithm 4 Main algorithm for large number of noisy outcomes

```
Initialization: consistent hypotheses $A \leftarrow [m]$, weights $w_i \leftarrow 0$ for $i \in [m]$, iteration index $t \leftarrow 0$
while $|A| > 1$ do
    if $t$ is a power of 2 then
        Let $Z \subseteq A$ be the subset of $2m^\alpha$ hypotheses with lowest $w_i$
        Invoke Member($Z$)
        If a hypothesis is identified in $Z$, then Break
    Select a test $T \in \mathcal{T}$ maximizing $\frac{1}{2}(|T^+ \cap A| + |T^- \cap A|)$ and observe outcome $o_T$
    Set $R \leftarrow \{i \in [m] : M_{T,i} = -o_T\}$ and $A \leftarrow A \backslash R$        ▷ Remove incompatible hypotheses
    Set $w_i \leftarrow w_i + 1$ for each for each $i \in T^*$        ▷ Update the weights of the hypotheses in $T^*$
    $t \leftarrow t + 1$.
```

Fix any expanded hypothesis $(i,\omega) \in H$. For any $t \geq 1$, let $\theta_{i,\omega}(t)$ denote the fraction of the first t tests in $P_{i,\omega}$ that are $\star$-tests for hypothesis i. Recall that $P_{i,\omega}$ only contains tests from Step 7. Let $\rho = 4$ and define

$$t_{i,\omega} = \max\left\{ t \in \{2^0, 2^1, \cdots, 2^{\log m}\} \ : \ \theta_{i,\omega}(t') \geq \frac{1}{\rho} \text{ for all } t' \leq t \right\}. \tag{6}$$

If $t_{i,\omega} > |P_{i,\omega}|$ then we simply set $t_{i,\omega} = |P_{i,\omega}|$.

Now we define the *truncated* decision tree $\overline{\mathbb{T}}$. By abuse of notation, we will use $\theta_i(t)$ and t_i as *random variables*, with randomness over ω. Observe that for any (i,ω), at the next power-of-two step[†] $2^{\lceil \log t_i \rceil}$, which we call the *truncation time*, the membership oracle will be invoked. Moreover, $2^{\lceil \log t_i \rceil} \leq 2t_i$, . This motivates us to define $\overline{\mathbb{T}}$ is the subtree of $\mathbb{T}$ consisting of the first $2^{\lceil \log t_{i,\omega} \rceil}$ tests along path $P_{i,\omega}$, for each $(i,\omega) \in H$. Under this definition, the cost of Algorithm 4 clearly equals the sum of the cost the truncated tree and cost for invoking membership oracles.

Our proof proceeds by bounding the cost of Algorithm 4 at power-of-two steps and other steps. In other words, we will decompose the cost into the cost incurred by invoking the membership oracle

[†] Unless stated otherwise, we denote $\log := \log_2$.

and selecting the greedy tests. We start with the easier task of bounding the cost for the membership oracle. The oracle Member is always invoked on $|Z| = O(m^\alpha)$ hypotheses. Using Lemma 5, the expected total number of tests due to Step 4 is $O(m^\alpha \log m)$. By Lemma 4, when $\alpha \le \frac{1}{2}$, this cost is $O(\log m \cdot \mathrm{OPT})$.

The remaining part of this subsection focuses on bounding the cost of the truncated tree as $O(\log m) \cdot \mathrm{OPT}$. With this inequality, we obtain an expected cost of

$$O(\log m)\cdot(m^\alpha + OPT) \le_{(\text{as } \alpha<\frac{1}{2})} O(\log m)\cdot(m^{1-\alpha} + OPT) \le_{\left(\text{Lemma } 4\right)} O(\log m)\cdot OPT,$$

and Theorem 8 follows. At a high level, for a fixed hypothesis $i \in [m]$, we will bound the cost of the truncated tree as follows:

$$\begin{aligned} & i \text{ has low fraction of } \star\text{-tests at } t_i \\ \underset{Lemma\ 6}{\Longrightarrow}\ & i \text{ is among the top } O(m^\alpha) \text{ hypotheses at } t_i \\ \underset{Lemma\ 5}{\Longrightarrow}\ & i \text{ is identified w.h.p. by Member}(Z) \text{ at } 2^{\lceil \log t_i \rceil} \le 2t_i, \text{ hence the truncated path is } (2,2)\text{-greedy} \\ \underset{Theorem\ 9}{\Longrightarrow}\ & \text{the expected cost conditional on } i \text{ is } O(\log m)\cdot \mathrm{SSC}(i) \end{aligned}$$

and finally by summing over $i \in [m]$, it follows from Lemma 2 that the cost of the *truncated* tree is $O(\log m){\cdot}\mathrm{OPT}$. We formalize each step below.

Consider the first step, formally we show that if $\theta_i(t) < \frac{1}{4}$, then there are $O(m^\alpha)$ hypotheses with fewer $\star$-tests than i. Suppose i is the target hypothesis and $\theta_i(t)$ drops below $\frac{1}{4}$ at t, that is, only less than a quarter of the tests selected are 2-greedy for $\mathrm{SSC}(i)$. Recall that if $i \in T^*$ where T maximizes $\frac{1}{2}(|A\cap T^+| + |A \cap T^-|)$, then $S_T(i)$ is 2-greedy set for $\mathrm{SSC}(i)$, so we deduce that less than a $\frac{t}{4}$ tests selected are $\star$-tests for i, or, at least $\frac{3t}{4}$ tests selected thus far are *deterministic* for i. We next utilize the sparsity assumption to show that there can be at most $O(m^\alpha)$ such hypotheses.

Lemma 6. *Consider any $W \subseteq \mathcal{T}$ and $I \subseteq [m]$. For $i \in I$, let $D(i) = |\{T \in W : M_{T,i} \neq *\}|$ denote the number of tests in W for which i has deterministic (i.e. ± 1) outcomes. For each $\kappa \ge 1$, define $I' = \{i \in I : D(i) > |W|/\kappa\}$. Then, $|I'| \le \kappa m^\alpha$.*

Proof. By definition of I' and α-sparsity, it holds that

$$|I'| \cdot \frac{|W|}{\kappa} < \sum_{i \in I} D(i) = \sum_{T \in W} |\{i \in I : M_{T,i} \neq *\}| \leq |W| \cdot m^{\alpha},$$

where the last step follows since $|T^*| \leq m^{\alpha}$ for each test T. The proof follows immediately by rearranging. □

We now complete the analysis using the relation to SSC. Fix any hypothesis $i \in [m]$ and consider decision tree $\overline{\mathbb{T}}_i$ obtained by *conditioning* $\overline{\mathbb{T}}$ on $\bar{i} = i$. Lemma 3 and the definition of truncation together imply that $\overline{\mathbb{T}}_i$ is $(2,4)$-greedy for SSC(i), so by Theorem 9, the expected cost of $\overline{\mathbb{T}}_i$ is $O(\log m) \cdot \mathrm{OPT}_{\mathrm{SSC}(i)}$. Now, taking expectations over $i \in [m]$, the expected cost of $\overline{\mathbb{T}}$ is $O(\log m) \sum_{i=1}^{m} \pi_i \cdot \mathrm{OPT}_{\mathrm{SSC}(i)}$. Recall from Lemma 2 that

$$\mathrm{OPT} \geq \sum_{i \in [m]} \pi_i \cdot \mathrm{OPT}_{\mathrm{SSC}(i)},$$

and therefore the cost of $\overline{\mathbb{T}}$ is $O(\log m) \cdot \mathrm{OPT}$.

Correctness. We finally show that our algorithm identifies the target hypothesis $\bar{i}$ with high probability. By definition of t_i, where the path is truncated, $\bar{i}$ has less than $\frac{1}{4}$ fraction of $\star$-tests. Thus, at iteration $2^{\lceil \log t_{\bar{i}} \rceil}$, i.e. the first time the membership oracle is invoked after t_i, $\bar{i}$ has less than $\frac{1}{2}$ fraction of $\star$-tests. Hence, by Lemma 6, $\bar{i}$ is among the $O(m^{\alpha})$ hypotheses with fewest $\star$-tests. Finally it follows from Lemma 5 that $\bar{i}$ is identified correctly with probability at least $1 - \frac{1}{m}$.

6. Experiments

We implemented our algorithms on real-world and synthetic data sets. We compared our algorithms' cost (expected number of tests) with an information theoretic lower bound on the optimal cost and show that the difference is negligible. Thus, despite our logarithmic approximation ratios, the practical performance is much better.

Chemicals with Unknown Test Outcomes. We considered a data set called WISER[‡], which includes 414 chemicals (hypothesis) and 78 binary tests. Every chemical has either positive, negative

[‡] https://wiser.nlm.nih.gov

or unknown result on each test. The original instance (called WISER-ORG) is not identifiable: so our result does not apply directly. In Appendix E we show how our result can be extended to such "non-identifiable" ODTN instances (this requires a more relaxed stopping criterion defined on the "similarity graph"). In addition, we also generated a modified dataset by removing chemicals that are not identifiable from each other, to obtain a perfectly identifiable dataset (called WISER-ID). In generating the WISER-ID instance, we used a greedy rule that iteratively drops the highest-degree hypothesis in the similarity graph until all remaining hypotheses are uniquely identifiable. WISER-ID has 255 chemicals.

Random Binary Classifiers with Margin Error. We construct a dataset containing 100 two-dimensional points, by picking each of their attributes uniformly in $[-1000, 1000]$. We also choose 2000 random triples (a, b, c) to form linear classifiers $\frac{ax+by}{\sqrt{a^2+b^2}} + c \leq 0$, where $a, b \sim N(0,1)$ and $c \sim U(-1000, 1000)$. The point labels are binary and we introduce noisy outcomes based on the distance of each point to a classifier. Specifically, for each threshold $d \in \{0, 5, 10, 20, 30\}$ we define dataset CL-d that has a noisy outcome for any classifier-point pair where the distance of the point to the boundary of the classifier is smaller than d. In order to ensure that the instances are perfectly identifiable, we remove "equivalent" classifiers and we are left with 234 classifiers.

Distributions. For the distribution over the hypotheses, we considered permutations of power law distribution ($\Pr[X = x; \alpha] = \beta x^{-\alpha}$) for $\alpha = 0, 0.5$ and 1. Note that, $\alpha = 0$ corresponds to uniform distribution. To be able to compare the results across different classifiers' datasets meaningfully, we considered the same permutation in each distribution.

Algorithms. We implement the following algorithms: the adaptive $O(r + \log m + \log \frac{1}{\varepsilon})$-approximation (which we denote ODTN$_r$), the adaptive $O(c \log |\Omega| + \log m + \log \frac{1}{\varepsilon})$-approximation (ODTN$_c$), the non-adaptive $O(\log m)$-approximation (Non-Adap) and a slightly adaptive version of Non-Adap (Low-Adap). Algorithm Low-Adap considers the same sequence of tests as Non-Adap while (adaptively) skipping non-informative tests based on observed outcomes. For the non-identifiable instance (WISER-ORG) we used the $O(d + \min(c, r) + \log m + \log \frac{1}{\varepsilon})$-approximation algorithms with both *neighborhood* and *clique* stopping criteria (see Appendix E). The implementations of the adaptive and non-adaptive algorithms are available online.[§]

Data / Algorithm	WISER-ID	Cl-0	Cl-5	Cl-10	Cl-20	Cl-30
Low-BND	**7.994**	**7.870**	**7.870**	**7.870**	**7.870**	**7.870**
ODTN$_r$	8.357	7.910	7.927	7.915	7.962	8.000
ODTN$_h$	9.707	7.910	7.979	8.211	8.671	8.729
Non-Adap	11.568	9.731	9.831	9.941	9.996	10.204
Low-Adap	9.152	8.619	8.517	8.777	8.692	8.803

Table 1 Cost of Different Algorithms for $\alpha = 0$ (Uniform Distribution).

Data / Algorithm	WISER-ID	Cl-0	Cl-5	Cl-10	Cl-20	Cl-30
Low-BND	7.702	7.582	7.582	7.582	7.582	7.582
ODTN$_r$	8.177	7.757	7.780	7.789	7.831	7.900
ODTN$_h$	9.306	7.757	7.829	8.076	8.497	8.452
Non-Adap	11.998	9.504	9.500	9.694	9.826	9.934
Low-Adap	8.096	7.837	7.565	7.674	8.072	8.310

Table 2 Cost of Different Algorithms for $\alpha = 0.5$.

Results. Tables 1, Tables 2 and Tables 3 show the expected costs of different algorithms on all uniquely identifiable data sets when the parameter α in the distribution over hypothesis is $0, 0.5$ and 1 correspondingly. These tables also report values of an information theoretic lower bound (the entropy) on the optimal cost (Low-BND). As the approximation ratio of our algorithms are dependent on maximum number c of unknowns per hypothesis and maximum number r of

[§] https://github.com/FatemehNavidi/ODTN ; https://github.com/sjia1/ODT-with-noisy-outcomes

Data Algorithm	WISER-ID	Cl-0	Cl-5	Cl-10	Cl-20	Cl-30
Low-BND	6.218	6.136	6.136	6.136	6.136	6.136
ODTN$_r$	7.367	6.998	7.121	7.150	7.299	7.357
ODTN$_h$	8.566	6.998	7.134	7.313	7.637	7.915
Non-Adap	11.976	9.598	9.672	9.824	10.159	10.277
Low-Adap	9.072	8.453	8.344	8.609	8.683	8.541

Table 3 Cost of Different Algorithms for $\alpha = 1$.

Data Parameters	WISER-ORG	WISER-ID	Cl-0	Cl-5	Cl-10	Cl-20	Cl-30
r	388	245	0	5	7	12	13
Avg-r	50.46	30.690	0	1.12	2.21	4.43	6.54
h	61	45	0	3	6	8	8
Avg-h	9.51	9.39	0	0.48	0.94	1.89	2.79

Table 4 Maximum and Average Number of Stars per Hypothesis and per Test in Different Datasets.

Algorithm	Neighborhood Stopping	Clique Stopping
ODTN$_r$	11.163	11.817
ODTN$_h$	11.908	12.506
Non-Adap	16.995	21.281
Low-Adap	16.983	20.559

Table 5 Algorithms on WISER-ORG dataset with Neighborhood and Clique Stopping for Uniform Distribution.

unknowns per test, we also have included these parameters as well as their average values in Table 4. Table 5 summarizes the results on WISER-ORG with clique and neighborhood stopping criteria. We can see that ODTN$_r$ consistently outperforms the other algorithms and is very close to the information-theoretic lower bound.

Acknowledgements

A preliminary version of this paper appeared as Jia et al. (2019) in the proceedings of Neural Information Processing Systems (NeurIPS) 2019.

References

Micah Adler and Brent Heeringa. Approximating optimal binary decision trees. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 1–9. Springer, 2008.

Esther M Arkin, Henk Meijer, Joseph SB Mitchell, David Rappaport, and Steven S Skiena. Decision trees for geometric models. *International Journal of Computational Geometry & Applications*, 8(03):343–363, 1998.

Yossi Azar and Iftah Gamzu. Ranking with submodular valuations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1070–1079. SIAM, 2011.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 65–72, 2006.

Gowtham Bellala, Suresh K. Bhavnani, and Clayton Scott. Active diagnosis under persistent noise with unknown noise distribution: A rank-based approach. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 155–163, 2011.

Venkatesan T Chakaravarthy, Vinayaka Pandit, Sambuddha Roy, and Yogish Sabharwal. Approximating decision trees with multiway branches. In *International Colloquium on Automata, Languages, and Programming*, pages 210–221. Springer, 2009.

Venkatesan T. Chakaravarthy, Vinayaka Pandit, Sambuddha Roy, Pranjal Awasthi, and Mukesh K. Mohania. Decision trees for entity identification: Approximation algorithms and hardness results. *ACM Trans. Algorithms*, 7(2):15:1–15:22, 2011.

Yuxin Chen, Seyed Hamed Hassani, and Andreas Krause. Near-optimal bayesian active learning with correlated and noisy tests. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 223–231, 2017.

Ferdinando Cicalese, Eduardo Sany Laber, and Aline Medeiros Saettler. Diagnosis determination: decision trees optimizing simultaneously worst and expected testing cost. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 414–422, 2014.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Advances in neural information processing systems*, pages 337–344, 2005.

M.R. Garey and R.L. Graham. Performance bounds on the splitting algorithm for binary testing. *Acta Informatica*, 3:347–355, 1974.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Intell. Res.*, 42:427–486, 2011. doi: 10.1613/jair.3278. URL `https://doi.org/10.1613/jair.3278`.

Daniel Golovin and Andreas Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. *CoRR*, abs/1003.3967, 2017. URL `http://arxiv.org/abs/1003.3967`.

Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 766–774, 2010.

Andrew Guillory and Jeff A. Bilmes. Average-case active learning with costs. In *Algorithmic Learning Theory, 20th International Conference, ALT 2009, Porto, Portugal, October 3-5, 2009. Proceedings*, pages 141–155, 2009.

Andrew Guillory and Jeff A. Bilmes. Interactive submodular set cover. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 415–422, 2010.

Andrew Guillory and Jeff A. Bilmes. Simultaneous learning and covering with adversarial noise. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 369–376, 2011.

Anupam Gupta, Viswanath Nagarajan, and R Ravi. Approximation algorithms for optimal decision trees and adaptive tsp problems. *Mathematics of Operations Research*, 42(3):876–896, 2017.

Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 353–360, 2007.

Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Lett.*, 5(1):15–17, 1976/77.

Sungjin Im, Viswanath Nagarajan, and Ruben Van Der Zwaan. Minimum latency submodular cover. *ACM Transactions on Algorithms (TALG)*, 13(1):13, 2016.

Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, Drew Bagnell, and Siddhartha S. Srinivasa. Near optimal bayesian active learning for decision making. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pages 430–438, 2014.

Su Jia, Viswanath Nagarajan, Fatemeh Navidi, and R. Ravi. Optimal decision tree with noisy outcomes. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 3298–3308, 2019.

S Rao Kosaraju, Teresa M Przytycka, and Ryan Borgstrom. On an optimal split tree problem. In *Workshop on Algorithms and Data Structures*, pages 157–168. Springer, 1999.

Zhen Liu, Srinivasan Parthasarathy, Anand Ranganathan, and Hao Yang. Near-optimal algorithms for shared filter evaluation in data stream systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 133–146, 2008.

D. W. Loveland. Performance bounds for binary testing with arbitrary weights. *Acta Inform.*, 22(1):101–114, 1985.

Mikhail Ju. Moshkov. Greedy algorithm with weights for decision tree construction. *Fundam. Inform.*, 104 (3):285–292, 2010.

Mohammad Naghshvar, Tara Javidi, and Kamalika Chaudhuri. Noisy bayesian active learning. In *50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012, Allerton Park & Retreat Center, Monticello, IL, USA, October 1-5, 2012*, pages 1626–1633, 2012.

Feng Nan and Venkatesh Saligrama. Comments on the proof of adaptive stochastic set cover based on adaptive submodularity and its implications for the group identification problem in "group-based active query selection for rapid diagnosis in time-critical situations". *IEEE Trans. Information Theory*, 63 (11):7612–7614, 2017.

Fatemeh Navidi, Prabhanjan Kambadur, and Viswanath Nagarajan. Adaptive submodular ranking and routing. *Oper. Res.*, 68(3):856–877, 2020.

Robert D. Nowak. Noisy generalized binary search. In *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.*, pages 1366–1374, 2009.

Aline Medeiros Saettler, Eduardo Sany Laber, and Ferdinando Cicalese. Trading off worst and expected cost in decision tree problems. *Algorithmica*, 79(3):886–908, 2017.

Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.