Group-Based Active Query Selection for Rapid Diagnosis in Time-Critical Situations

Gowtham Bellala, Suresh K. Bhavnani, and Clayton Scott

***Abstract*—In applications such as active learning and disease/fault diagnosis, one often encounters the problem of identifying an unknown object through a minimal number of queries. This problem has been referred to as query learning or object/entity identification. We consider three extensions of this fundamental problem that are motivated by practical considerations in real-world,time-critical identification tasks such as emergency response. First, we consider the problem where the objects are partitioned into groups, and the goal is to identify only the group to which the object belongs. Second, we address the situation where the queries are partitioned into groups, and an algorithm may suggest a group of queries to a human user, who then selects the actual query. Third, we consider the problem of object identification in the presence of persistent query noise, and relate it to group identification. To address these problems we show that a standard algorithm for object identification, known as generalized binary search, may be viewed as a generalization of Shannon-Fano coding. We then extend this result to the group-based settings, leading to new algorithms, whose performance is demonstrated through a logarithmic approximation bound, and through experiments on simulated data and a database used for toxic chemical identification.**

***Index Terms*—Active learning, decision trees, generalized binary search, persistent noise, Shannon-Fano coding, submodularity.**

I. Introduction

In emergency response applications, as well as other time-critical diagnostic tasks, there is a need to rapidly identify a cause by selectively acquiring information from the environment. For example, in the problem of toxic chemical identification, a first responder may question victims of chemical exposure regarding the symptoms they experience. Chemicals that are inconsistent with the reported symptoms may then be eliminated. Because of the importance of this problem, several organizations have constructed extensive evidence-based databases (e.g., WISER[1]) that record toxic chemicals and the acute symptoms which they are known to cause. Unfortunately, many symptoms tend to be nonspecific (e.g., vomiting can be caused by many different chemicals), and it is therefore critical for the first responder to pose these questions in a sequence that leads to chemical identification in as few questions as possible.

This problem has been studied from a mathematical perspective for decades, and has been described variously as query learning (with membership queries) [1], active learning [2], object/entity identification [3], [4], and binary testing [4], [5]. In this work we refer to the problem as object identification. The standard mathematical formulation of object identification is often idealized relative to many real-world diagnostic tasks, in that it does not account for time constraints and resulting input errors. In this paper we investigate algorithms that extend object identification to such more realistic settings by addressing the need for rapid response, and error-tolerant algorithms.

In these problems, there is a set $\Theta = \{\theta_1, \ldots, \theta_M\}$ of M different objects and a set $Q = \{q_1, \ldots, q_N\}$ of N distinct subsets of Θ known as queries. An unknown object θ is generated from this set Θ with a certain *prior* probability distribution $\Pi = (\pi_1, \ldots, \pi_M)$, i.e., $\pi_i = \Pr(\theta = \theta_i)$. The goal is to determine the unknown object $\theta \in \Theta$ through as few queries from Q as possible, where a query $q \in Q$ returns a value 1 if $\theta \in q$, and 0 otherwise. An object identification algorithm thus corresponds to a decision tree, where the internal nodes are queries, and the leaf nodes are objects. Problems of this nature arise in applications such as fault testing [6], [7], machine diagnostics [8], disease diagnosis [5], [9], computer vision [10], [11], pool-based active learning [2], [12], [13] and the adaptive traveling salesperson problem [14]. Algorithms and performance guarantees have been extensively developed in the literature, as described in Section I-A below.

In the context of toxic chemical identification, the objects are chemicals, and the queries are symptoms. An object identification algorithm will prompt the first responder with a symptom. Once the presence or absence of that symptom is determined, a new symptom is suggested by the algorithm, and so on, until the chemical is uniquely determined. In this paper, we consider variations on this basic object identification framework that are motivated by toxic chemical identification, and are naturally applicable to other time-critical diagnostic tasks. In particular, we develop theoretical results and new algorithms for what might be described as group-based active learning.

First, we consider the case where Θ is partitioned into groups of objects, and it is only necessary to identify the group to which the unknown object belongs. For example, the appropriate response to a toxic chemical may only depend on the class of chemicals to which it belongs (pesticide, corrosive acid, etc.). As our experiments reveal, an active query selection algorithm designed to rapidly identify individual objects is not necessarily efficient for group identification.

Manuscript received November 21, 2009; revised April 01, 2011; accepted July 06, 2011. Date of current version January 06, 2012. This work was supported in part by NSF Awards 0830490 and 0953135, in part by NIH Grant UL1RR024986, and in part by CDC/NIOSH Grant R21 OH009441-01A2. The material in this paper was presented in part at the Neural Information Processing Systems Conference, Vancouver, BC, Canada, December 2010.

G. Bellala and C. Scott are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: gowtham@umich.edu; clayscot@umich.edu).

S. K. Bhavnani is with the Institute for Translational Sciences, University of Texas Medical Branch, Galveston, TX 77555 USA (e-mail: skbhavnani@gmail.com).

Communicated by A. Krzyzak, Associate Editor for Pattern Recognition, Statistical Learning, and Inference.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2011.2169296

[1]http://wiser.nlm.nih.gov/

0018-9448/$31.00 © 2012 IEEE

Second, we consider the problem where the set Q of queries is partitioned into groups (respiratory symptoms, cardio symptoms, etc.). Instead of suggesting specific symptoms to the user, we design an algorithm that suggests a group of queries, and allows the user the freedom to input information on any query in that group. Although such a system will theoretically be less efficient, it is motivated by the fact that in a practical application, some symptoms will be easier for a given user to understand and identify. Instead of suggesting a single symptom, which might seem "out of the blue" to the user, suggesting a query group will be less bewildering, and hence lead to a more efficient and accurate outcome. Our experiments demonstrate that the proposed algorithm based on query groups identifies objects in nearly as few queries as a fully active method.

Third, we apply our algorithm for group identification to the problem of object identification under persistent query noise. Persistent query noise occurs when the response to a query is in error, but cannot be re-sampled, as is often assumed in the literature. Such is the case when the presence or absence of a symptom is incorrectly determined, which is more likely in a stressful emergency response scenario. Experiments show our method offers significant gains over algorithms not designed for persistent query noise.

Our algorithms are derived in a common framework, and are based on a reinterpretation of a standard object identification algorithm (the splitting algorithm, or generalized binary search) as a generalized form of Shannon-Fano coding. We first establish an exact formula for the expected number of queries required to identify an object using an arbitrary decision tree, and show that the splitting algorithm effectively performs a greedy, top-down optimization of this objective. We then extend this formula to the case of group identification and query groups, and develop analogous greedy algorithms. In the process, we provide a new interpretation of impurity-based decision tree induction for multiclass classification. We also develop a logarithmic approximation bound for group identification, using the notion of submodular functions.

We apply our algorithms to both synthetic data and to the WISER database (version 4.21). WISER, which stands for **W**ireless **I**nformation **S**ystem for **E**mergency **R**esponders, is a decision support system developed by the National Library of Medicine (NLM) for first responders. This database describes the binary relationship between 298 toxic chemicals (corresponding to the number of distinguishable chemicals in this database) and 79 acute symptoms. The symptoms are grouped into 10 categories (e.g., neurological, cardio) as determined by NLM, and the chemicals are grouped into 16 categories (e.g., pesticides, corrosive acids) as determined by a toxicologist and a Hazmat expert.

A. Prior and Related Work

The problem of selecting an optimal sequence of queries from Q to uniquely identify an unknown object θ is equivalent to determining an optimal binary decision tree, where each internal node in the tree corresponds to a query, each leaf node corresponds to a unique object from the set Θ and the optimality is with respect to minimizing the expected depth of the leaf node corresponding to θ. In the special case when the query set Q is *complete* (a query set Q is said to be *complete* if for any $S \subseteq \Theta$ there exists a query $q \in Q$ such that either $q = S$ or $\Theta \setminus q = S$), the problem of constructing an optimal binary decision tree is equivalent to construction of optimal variable-length binary prefix codes with minimum expected length. This problem has been widely studied in information theory with both Shannon [15] and Fano [16] independently proposing a top-down greedy strategy to construct suboptimal binary prefix codes, popularly known as Shannon-Fano codes. Later Huffman [17] derived a simple bottom-up algorithm to construct optimal binary prefix codes. A well known lower bound on the expected length of binary prefix codes is given by the Shannon entropy of the probability distribution Π[18].

When the query set Q is not *complete*, an object identification problem can be considered as "constrained" prefix coding with the same lower bound on the expected depth of a tree. This problem has also been studied extensively in the literature with Garey [3], [4] proposing a dynamic programming based algorithm to find an optimal solution. This algorithm runs in exponential time in the worst case. Later, Hyafil and Rivest [19] showed that determining an optimal binary decision tree for this problem is NP-complete. Thereafter, various greedy algorithms [5], [20], [21] have been proposed to obtain a suboptimal binary decision tree. The most widely studied algorithm, known as the *splitting algorithm* [5] or generalized binary search (GBS) [2], [12], selects a query that most evenly divides the probability mass of the remaining objects [2], [5], [12], [22]. Various bounds on the performance of this greedy algorithm have been established in [2], [5], [12]. In addition, several variants of this problem such as multiway or k-ary splits (instead of binary splits) [23]–[25] and unequal query costs [13], [14], [25], [26] have also been studied in the literature.

Goodman and Smyth [22] observe that the splitting algorithm can be viewed as a generalized version of Shannon-Fano coding. In Section II, we demonstrate the same through an alternative approach that can be generalized to the group-based settings, leading to efficient algorithms in these settings. Golovin *et al.* [28] simultaneously studied the problem of group identification, and also proposed a near-optimal algorithm, which is discussed in more detail in Section III-C.

Though most of the above work has been devoted to object identification in the ideal setting assuming no noise, it is unrealistic to assume that the responses to queries are without error in many applications. The problem of identifying an unknown object in the presence of query noise has been studied in [12], [29], [30] where the queries can be re-sampled or repeated. However, in certain applications, re-sampling or repeating a query does not change the query response confining the algorithm to non-repeatable queries. The work by Rényi in [31] is regarded to be the first to consider this more stringent noise model, also referred to as persistent noise in the literature [32]–[34]. However, his work has focused on the passive setting where the queries are chosen at random. Learning under persistent noise model has also been studied in [32], [33], [35] where the goal was to identify or learn Disjunctive Normal Form (DNF) formulae from noisy data. The query (label) complexity of pool-based active learning in the Probably Approximately Correct (PAC) model in the presence of persistent classification noise has been studied

in [34] and active learning algorithms in this setting have been proposed in [34] and [36].

Here, we focus on the problem of object identification under the persistent noise model where the goal is to uniquely identify the true object. A similar problem studied in the game-theoretic literature is known as the Rényi-Ulam's problem, where the goal is to identify an unknown number x from a known set of numbers $\{1,\ldots,n\}$ using as few binary questions (of the form "Is x a member of $S \subseteq \{1,\ldots,n\}$?") as possible, with at most k errors in the obtained responses [37]–[40]. This problem is similar to the problem of designing minimum length k-error correcting codes in communication theory, where the query set Q is complete [41]. However, it is different from the problem considered in this paper in that repetition of queries does not change the query response.

It is possible to extend our Theorems 1 and 2 to the case where the cost of additional queries grows exponentially [27]. Finally, this work was motivated by earlier work that applied GBS to WISER [42].

B. Notation

We denote an object identification problem by a pair $(\mathbf{B},\Pi)$ where $\mathbf{B}$ is a binary matrix with b_{ij} equal to 1 if $\theta_i \in q_j$, and 0 otherwise. We assume that the rows of $\mathbf{B}$ are distinct, i.e., we make the assumption of unique identifiability of every object in Θ. This is reasonable since objects that have similar query responses for all queries in Q, i.e., objects that are not distinguishable, can always be grouped into a single meta-object.

A decision tree T constructed on $(\mathbf{B},\Pi)$ has a query from the set Q at each of its internal nodes with the leaf nodes terminating in the objects from the set Θ. At each internal node in the tree, the object set under consideration is divided into two subsets, corresponding to the objects that respond 0 and 1 to the query, respectively. For a decision tree with L leaves, the leaf nodes are indexed by the set $\mathcal{L} = \{1,\ldots,L\}$ and the internal nodes are indexed by the set $\mathcal{I} = \{L+1,\ldots,2L-1\}$. At any internal node $a \in \mathcal{I}$, let $l(a), r(a)$ denote the "left" and "right" child nodes, where the set $\Theta_a \subseteq \Theta$ corresponds to the set of objects that reach node 'a', and the sets $\Theta_{l(a)} \subseteq \Theta_a, \Theta_{r(a)} \subseteq \Theta_a$ corresponds to the set of objects that respond 0 and 1 to the query at node 'a', respectively. We denote by $\pi_{\Theta_a} := \sum_{\{i:\theta_i\in\Theta_a\}} \pi_i$, the probability mass of the objects under consideration at any node 'a' in the tree. Also, at any node 'a,' the set $Q_a \subseteq Q$ corresponds to the set of queries that have been performed along the path from the root node up to node 'a.'

We denote the Shannon entropy of a vector $\Pi = (\pi_1,\ldots,\pi_M)$ by $H(\Pi) := -\sum_i \pi_i \log_2 \pi_i$ and the Shannon entropy of a proportion $\pi \in [0,1]$ by $H(\pi) := -\pi\log_2\pi - (1-\pi)\log_2(1-\pi)$, where we use the limit, $\lim_{\pi\to 0}\pi\log_2\pi = 0$ to define the limiting cases. Finally, given a tree T, we use the random variable $K(T)$ to denote the number of queries required to identify an unknown object θ or the group of an unknown object θ using the given tree.

II. Generalized Shannon-Fano Coding

Before proceeding to the group-based setting, we first present an exact formula for the standard object identification problem. This result allows us to interpret the splitting algorithm or GBS as generalized Shannon-Fano coding. Furthermore, our proposed algorithms for group-based settings are based on generalizations of this result.

First, we define a parameter called the *reduction factor* on the binary matrix/tree combination that provides a useful quantification on the expected number of queries required to identify an unknown object.

Definition 1: A *reduction factor* at any internal node 'a' in a decision tree is defined as $\rho_a = \max(\pi_{\Theta_{l(a)}}, \pi_{\Theta_{r(a)}})/\pi_{\Theta_a}$ and the overall reduction factor of a tree is defined as $\rho = \max_{a\in\mathcal{I}} \rho_a$.

Note from the above definition that $0.5 \le \rho_a \le \rho \le 1$ and we describe a decision tree with $\rho = 0.5$ to be a perfectly balanced tree.

Given an object identification problem $(\mathbf{B},\Pi)$, let $\mathcal{T}(\mathbf{B},\Pi)$ denote the set of decision trees that can uniquely identify all the objects in the set Θ. For any decision tree $T \in \mathcal{T}(\mathbf{B},\Pi)$, let $\{\rho_a\}_{a\in\mathcal{I}}$ denote the set of reduction factors and let d_i denote the depth of object θ_i in the tree. Then, the expected number of queries required to identify an unknown object using the given tree is equal to

$$\mathbb{E}[K(T)] = \sum_{i=1}^{M} \Pr(\theta=\theta_i)\mathbb{E}[K(T)|\theta=\theta_i] = \sum_{i=1}^{M} \pi_i d_i.$$

Theorem 1: The expected number of queries required to identify an unknown object using a tree $T \in \mathcal{T}(\mathbf{B},\Pi)$ with reduction factors $\{\rho_a\}_{a\in\mathcal{I}}$ is given by

$$\begin{aligned}\mathbb{E}[K(T)] &= H(\Pi) + \sum_{a\in\mathcal{I}} \pi_{\Theta_a}[1-H(\rho_a)] \\ &= \frac{H(\Pi)}{\sum_{a\in\mathcal{I}} \tilde{\pi}_{\Theta_a} H(\rho_a)} \end{aligned} \tag{1}$$

where $\tilde{\pi}_{\Theta_a} := \frac{\pi_{\Theta_a}}{\sum_{r\in\mathcal{I}} \pi_{\Theta_r}}$.

Proof: The first equality is a special case of Theorem 2. The second equality follows from the observation $\mathbb{E}[K(T)] = \sum_{i=1}^{M} \pi_i d_i = \sum_{a\in\mathcal{I}} \pi_{\Theta_a}$. Hence replacing π_{Θ_a} with $\tilde{\pi}_{\Theta_a} \cdot \mathbb{E}[K(T)]$ in the first equality leads to the result. ■

In the second equality, the term $\sum_{a\in\mathcal{I}} \tilde{\pi}_{\Theta_a} H(\rho_a)$ denotes the average entropy of the reduction factors, weighted by the proportion of times each internal node 'a' is queried in the tree. This theorem reiterates an earlier observation that the expected number of queries required to identify an unknown object using a tree constructed on $(\mathbf{B},\Pi)$ (where the query set Q is not necessarily a *complete* set) is bounded below by its entropy $H(\Pi)$. It also follows from the above result that a tree attains this minimum value (i.e., $\mathbb{E}[K(T)] = H(\Pi)$) iff it is perfectly balanced, i.e., the overall reduction factor ρ of the tree is equal to 0.5.

From the first equality, the problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$ can be formulated as the following optimization problem:

$$\min_{T\in\mathcal{T}(\mathbf{B},\Pi)} H(\Pi) + \sum_{a\in\mathcal{I}} \pi_{\Theta_a}[1-H(\rho_a)]. \tag{2}$$

Since Π is fixed, the optimization problem reduces to minimizing $\sum_{a\in\mathcal{I}} \pi_{\Theta_a}[1-H(\rho_a)]$ over the set of trees $\mathcal{T}(\mathbf{B},\Pi)$. Note that the reduction factor ρ_a depends on the query chosen

at node 'a' in a tree T. As aforementioned, finding a global optimal solution for this optimization problem is NP-complete.

Instead, we may take a top down approach and minimize the objective function by minimizing the term $\pi_{\Theta_a}[1 - H(\rho_a)]$ at each internal node, starting from the root node. Since π_{Θ_a} is independent of the query chosen at node 'a,' this reduces to minimizing ρ_a (i.e., choosing a split as balanced as possible) at each internal node $a \in \mathcal{I}$. The algorithm can be summarized as shown in Algorithm 1.

Algorithm 1: Greedy decision tree algorithm for object identification

Generalized Binary Search (GBS)

Initialization : Let the leaf set consist of the root node

while some leaf node 'a' has $|\Theta_a| > 1$ **do**

for each query $q \in Q \setminus Q_a$ **do**

Find $\Theta_{l(a)}$ and $\Theta_{r(a)}$ produced by making a split with query q

Compute the reduction factor ρ_a produced by query q

end

Choose a query with the smallest reduction factor

Form child nodes $l(a), r(a)$

end

Note that when the query set Q is *complete*, Algorithm 1 is similar to Shannon-Fano coding [15], [16]. The only difference is that in Shannon-Fano coding, for computational reasons, the queries are restricted to those that are based on thresholding the prior probabilities π_i.

Corollary 1: The standard splitting algorithm/GBS is a greedy algorithm to minimize the expected number of queries required to uniquely identify an object.

Corollary 2 below follows from Theorem 1. It states that given a tree T with overall reduction factor $\rho < 1$, the average complexity of identifying an unknown object using this tree is $O(\log_2 M)$. Recently, Nowak [12] showed there are geometric conditions (incoherence and neighborliness) that also bound the worst-case depth of the tree to be $O(\log_2 M)$, assuming a uniform prior on objects. These conditions imply that the reduction factors are close to $\frac{1}{2}$ except possibly near the very bottom of the tree where they could be close to 1. Because ρ_a could be close to 1 for deeper nodes, the upper bound on $\mathbb{E}[K(T)]$ based on the overall reduction factor ρ given below could be very loose in practice.

Corollary 2: The expected number of queries required to identify an unknown object using a tree T with overall reduction factor ρ constructed on $(\mathbf{B}, \Pi)$ is bounded above by

$$\mathbb{E}[K(T)] \leq \frac{H(\Pi)}{H(\rho)} \leq \frac{\log_2 M}{H(\rho)}.$$

	q_1	q_2	q_3	Group label, y
θ_1	0	1	1	1
θ_2	1	1	0	1
θ_3	0	1	0	1
θ_4	1	0	0	2

Fig. 1. Toy Example 1.

Proof: Using the second equality in Theorem 1, we get

$$\mathbb{E}[K(T)] = \frac{H(\Pi)}{\sum_{a \in \mathcal{I}} \tilde{\pi}_{\Theta_a} H(\rho_a)} \leq \frac{H(\Pi)}{H(\rho)} \leq \frac{\log_2 M}{H(\rho)}$$

where the first inequality follows from the definition of ρ, $\rho \geq \rho_a \geq 0.5, \forall a \in \mathcal{I}$ and the last inequality follows from the concavity of the entropy function. ■

In the sections that follow, we show how Theorem 1 and Algorithm 1 may be generalized, leading to principled strategies for group identification, object identification with group queries and object identification with persistent noise.

III. Group Identification

We now move to the problem of group identification, where the goal is not to determine the unknown object $\theta \in \Theta$, rather the group to which the object belongs. Here, in addition to the binary matrix $\mathbf{B}$ and *a priori* probability distribution Π on the objects, the group labels for the objects are also provided, where the groups are assumed to be disjoint. Note that if the groups are overlapping, it can be reduced to the disjoint setting by finding the smallest partition of the objects such that the group labels are constant on each cell of the partition. Then, a group identification algorithm would identify precisely those groups to which the object belongs. For example, in toxic chemical identification, a first responder may only need to know whether a chemical is a pesticide, a corrosive acid, or both. Hence, it could be reasonable to reduce a group identification problem with overlapping groups to that of disjoint groups arising out of its partition. Thus, we devote our attention to the problem of group identification with disjoint groups.

We denote a group identification problem by $(\mathbf{B}, \Pi, \mathbf{y})$, where $\mathbf{y} = (y_1, \ldots, y_M)$ denotes the group labels of the objects $y_i \in \{1, \ldots, m\}$. Let $\{\Theta^i\}_{i=1}^m$ be a partition of the object set Θ, where Θ^i denotes the set of objects in Θ that belong to group i. It is important to note here that the group identification problem cannot be simply reduced to an object identification problem with groups $\{\Theta^1, \ldots, \Theta^m\}$ as "meta-objects," since the objects within a group need not respond the same to each query. For example, consider the toy example shown in Fig. 1 where the objects θ_1, θ_2 and θ_3 belonging to group 1 cannot be considered as one single meta-object as these objects respond differently to queries q_1 and q_3.

In this context, we also note that GBS can fail to find a good solution for a group identification problem as it does not take the group labels into consideration while choosing queries. Once again, consider the toy example shown in Fig. 1 where just one query (query q_2) is sufficient to identify the group of an unknown object, whereas GBS requires 2 queries to identify the group when the unknown object is either θ_2 or θ_4, as shown in

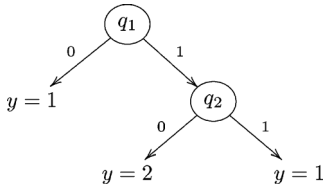

Fig. 2. Decision tree constructed using GBS for group identification on toy example 1.

Fig. 2. Hence, we develop a new strategy which accounts for the group labels when choosing the best query at each stage.

Note that when constructing a tree for group identification, a greedy, top-down algorithm terminates splitting when all the objects at the node belong to the same group. Hence, a tree constructed in this fashion can have multiple objects ending in the same leaf node and multiple leaves ending in the same group.

For a tree with L leaves, we denote by $\mathcal{L}^i \subset \mathcal{L} = \{1, \ldots, L\}$ the set of leaves that terminate in group i. Similar to $\Theta^i \subseteq \Theta$, we denote by $\Theta^i_a \subseteq \Theta_a$ the set of objects that belong to group i at any internal node $a \in \mathcal{I}$ in the tree. Also, in addition to the reduction factors defined in Section II, we define a new set of reduction factors called the group reduction factors at each internal node.

Definition 2: The group reduction factor of group i at any internal node 'a' in a decision tree is defined as $\rho^i_a = \max(\pi_{\Theta^i_{l(a)}}, \pi_{\Theta^i_{r(a)}})/\pi_{\Theta^i_a}$.

Given a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$, let $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ denote the set of decision trees that can uniquely identify the groups of all objects in the set Θ. For any decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$, let ρ_a denote the reduction factor and let $\{\rho^i_a\}_{i=1}^m$ denote the set of group reduction factors at each of its internal nodes. Also, let d_j denote the depth of leaf node $j \in \mathcal{L}$ in the tree. Then the expected number of queries required to identify the group of an unknown object using the given tree is equal to

$$\mathbb{E}[K(T)] = \sum_{i=1}^{m} \Pr(\theta \in \Theta^i)\mathbb{E}[K(T)|\theta \in \Theta^i] = \sum_{i=1}^{m} \pi_{\Theta^i} \left[\sum_{j \in \mathcal{L}^i} \frac{\pi_{\Theta_j}}{\pi_{\Theta^i}} d_j \right].$$

Theorem 2: The expected number of queries required to identify the group of an object using a tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ with reduction factors $\{\rho_a\}_{a \in \mathcal{I}}$ and group reduction factors $\{\rho^i_a\}_{i=1}^m, \forall a \in \mathcal{I}$, is given by

$$\mathbb{E}[K(T)] = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta^i_a}}{\pi_{\Theta_a}} H(\rho^i_a) \right] \quad (3)$$

where $\Pi_{\mathbf{y}}$ denotes the probability distribution of the object groups induced by the labels $\mathbf{y}$, i.e., $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \ldots, \pi_{\Theta^m})$.

Proof: Special case of Theorem 7 below. See also [27]. ■

The above theorem states that given a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$, the expected number of queries required to identify the group of an unknown object is lower bounded by the entropy of the probability distribution of the groups. It also follows from the above result that this lower bound is achieved iff there exists a perfectly balanced tree (i.e., $\rho = 0.5$) with the group reduction factors equal to 1 at every internal node in the tree. Also, note that Theorem 1 is a special case of this theorem where each group has size 1 leading to $\rho^i_a = 1$ for all groups at every internal node.

Using Theorem 2, the problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$ can be formulated as the following optimization problem:

$$\min_{T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})} \sum\nolimits_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta^i_a}}{\pi_{\Theta_a}} H(\rho^i_a) \right]. \quad (4)$$

Note that here both the reduction factor ρ_a and the group reduction factors $\{\rho^i_a\}_{i=1}^m$ depend on the query chosen at node 'a.' Also, the above optimization problem being a generalized version of the optimization problem in (2) is NP-complete. Hence, we propose a suboptimal approach to solve the above optimization problem where we optimize the objective function locally instead of globally. We take a top-down approach and minimize the objective function by minimizing the term $\Delta_a := \left[1 - H(\rho_a) + \sum_{i=1}^m \frac{\pi_{\Theta^i_a}}{\pi_{\Theta_a}} H(\rho^i_a)\right]$ at each internal node, starting from the root node. The algorithm can be summarized as shown in Algorithm 2. This algorithm is referred to as Group Identification Splitting Algorithm (GISA) in the rest of this paper.

Algorithm 2: Greedy decision tree algorithm for group identification

Group Identification Splitting Algorithm (GISA)

Initialization : Let the leaf set consist of the root node

while some leaf node 'a' has more than one group of objects **do**

for each query $q_j \in Q \setminus Q_a$ **do**

Compute $\{\rho^i_a\}_{i=1}^m$ and ρ_a produced by making a split with query q_j

Compute the cost $\Delta_a(j)$ of making a split with query q_j

end

Choose a query with the least cost Δ_a at node 'a'

Form child nodes $l(a), r(a)$

end

Note that the objective function in this algorithm consists of two terms. The first term $[1 - H(\rho_a)]$ favors queries that evenly distribute the probability mass of the objects at node 'a' to its child nodes (regardless of the group) while the second term $\sum_i \frac{\pi_{\Theta^i_a}}{\pi_{\Theta_a}} H(\rho^i_a)$ favors queries that transfer an entire group of objects to one of its child nodes.

A. Connection to Impurity-Based Decision Tree Induction

As a brief digression, in this section we show a connection between the above algorithm and impurity-based decision tree

induction. In particular, we show that the above algorithm is equivalent to the decision tree splitting algorithm used in the C4.5 software package [43]. Before establishing this result, we briefly review the multi-class classification setting where impurity-based decision tree induction is popularly used.

In the multiclass classification setting, the input is training data $\mathbf{x}_1, \ldots, \mathbf{x}_M$ sampled from some input space (with an underlying probability distribution) along with their class labels, $y_1, \ldots, y_M$ and the task is to construct a classifier with the least probability of misclassification. Decision tree classifiers are grown by maximizing an impurity-based objective function at every internal node to select the best classifier from a set of base classifiers. These base classifiers can vary from simple axis-orthogonal splits to more complex nonlinear classifiers. The impurity-based objective function is

$$I(\Theta_a) - \left[\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} I(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} I(\Theta_{r(a)})\right] \tag{5}$$

which represents the decrease in impurity resulting from split 'a.' Here $I(\Theta_a)$ corresponds to the measure of impurity in the input subspace at node 'a' and π_{Θ_a} corresponds to the probability measure of the input subspace at node 'a.'

Among the various impurity functions suggested in the literature [44], [45], the entropy measure used in the C4.5 software package [43] is popular. In the multiclass classification setting with m different class labels, this measure is given by

$$I(\Theta_a) = -\sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \tag{6}$$

where $\pi_{\Theta_a}, \pi_{\Theta_a^i}$ are empirical probabilities based on the training data.

Similar to a group identification problem, the input here is a binary matrix $\mathbf{B}$ with b_{ij} denoting the binary label produced by base classifier j on training sample i, and a probability distribution Π on the training data along with their class labels $\mathbf{y}$. But unlike a group identification problem where the nodes in a tree are not terminated until all the objects belong to the same group, the leaf nodes here are allowed to contain some impurity in order to avoid overfitting. The following result extends Theorem 2 to the case of impure leaf nodes.

Theorem 3: The expected depth of a leaf node in a decision tree classifier $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ with reduction factors $\{\rho_a\}_{a\in\mathcal{I}}$ and class reduction factors $\{\rho_a^i\}_{i=1}^m, \forall a \in \mathcal{I}$, is given by

$$\mathbb{E}[K(T)] = H(\Pi_{\mathbf{y}}) + \sum_{a\in\mathcal{I}} \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right] - \sum_{a\in\mathcal{L}} \pi_{\Theta_a} I(\Theta_a) \tag{7}$$

where $\Pi_{\mathbf{y}}$ denotes the probability distribution of the classes induced by the class labels $\mathbf{y}$, i.e., $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \ldots, \pi_{\Theta^m})$ and $I(\Theta_a)$ denotes the impurity in leaf node 'a' given by (6).

Proof: The proof is given in Appendix A. ∎

The only difference compared to Theorem 2 is the last term, which corresponds to the average impurity in the leaf nodes.

Theorem 4: At every internal node in a tree, minimizing the objective function $\Delta_a := 1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)$ is equivalent to maximizing $I(\Theta_a) - \left[\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} I(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} I(\Theta_{r(a)})\right]$ with entropy measure as the impurity function.

Proof: The proof is given in Appendix B. ∎

Therefore, greedy optimization of (7) at internal nodes corresponds to greedy optimization of impurity. Also, note that optimizing (7) at a leaf assigns the majority vote class label. Therefore, we conclude that impurity-based decision tree induction with entropy as the impurity measure amounts to a greedy optimization of the expected depth of a leaf node in the tree. Also, Theorem 3 allows us to interpret impurity based splitting algorithms for multiclass decision trees in terms of reduction factors, which also appears to be a new insight.

B. Modified GISA With Near-Optimal Performance

As mentioned in Section I-A, the splitting algorithm or GBS has been shown to be near-optimal with a logarithmic approximation ratio [2], [12], [13], i.e.,

$$\mathbb{E}[K(\widehat{T})] \leq O\left(\ln \frac{1}{\pi_{\min}}\right) \mathbb{E}[K(T^*)]$$

where $\pi_{\min} := \min_i \pi_i$ is the minimum prior probability of any object, $\widehat{T}$ is a greedy tree constructed using GBS and T^* is an optimal tree for the given problem.

Recently, Golovin *et al.* [13] introduced the notion of adaptive submodularity and strong adaptive monotonicity (refer Appendix C), and showed that a greedy optimization algorithm with these properties can be near-optimal and achieve a logarithmic approximation ratio, with GBS being a specific instance of this class. Unfortunately, the objective function in GISA, i.e.,

$$H(\rho_a) - \sum_{i=1}^{m} \frac{\pi_a^i}{\pi_a} H(\rho_a^i) \tag{8}$$

does not satisfy these properties. We present a modified version of GISA that can be shown to be adaptive submodular and strong adaptive monotone, and hence can achieve a logarithmic approximation to the optimal solution.

The modified algorithm is to construct a top-down, greedy decision tree where at each internal node, a query that maximizes

$$\pi_{l(a)}\pi_{r(a)} - \sum_{i=1}^{m} \frac{\pi_a^i}{\pi_a} \pi_{l(a)}^i \pi_{r(a)}^i \tag{9}$$

is chosen. Essentially, the binary entropy terms $H(\rho_a)$ and $H(\rho_a^i)$ in (8) are approximated by the weighted Gini indices, $\pi_a^2(\rho_a(1-\rho_a))$ and $\left(\pi_a^i\right)^2(\rho_a^i(1-\rho_a^i))$, respectively. Note that in the special case where each group is of size 1, the query selection criterion in (9) reduces to $\pi_{l(a)}\pi_{r(a)}$, thereby reducing modified GISA to the standard splitting algorithm.

Given a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$, recall that $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ denotes the set of all possible trees that can uniquely

identify the group of any object from the set Θ. Then, let T^* denote a tree with the least expected depth, i.e.,

$$T^* \in \underset{T \in \mathcal{T}(\mathbf{B},\Pi,\mathbf{y})}{\arg\min}\ \mathbb{E}[K(T)]$$

and let $\widehat{T}$ denote a tree constructed using modified GISA. The following theorem states that the expected depth of $\widehat{T}$ is logarithmically close to that of an optimal tree.

Theorem 5: Let $(\mathbf{B}, \Pi, \mathbf{y})$ denote a group identification problem. For a greedy decision tree $\widehat{T}$ constructed using modified GISA, it holds that

$$\mathbb{E}[K(\widehat{T})] \leq \left(2\ln\left(\frac{1}{\sqrt{3}\pi_{\min}}\right)+1\right)\mathbb{E}[K(T^*)] \qquad (10)$$

where $\pi_{\min} := \min\{\pi \in \Pi : \pi > 0\}$ is the minimum prior probability of any object.

Proof: The proof is given in Appendix C. ■

In addition, if the query costs are unequal, the query selection criterion in modified GISA can be changed to $\arg\max_{q \notin Q_a} \Delta_a(q)/c(q)$, where $\Delta_a(q)$ is as defined in (9), and $c(q)$ is the cost of obtaining the response to query q. This simple heuristic has been shown to retain the near-optimal property [13], i.e.,

$$c(\widehat{T}) \leq \left(2\ln\left(\frac{1}{\sqrt{3}\pi_{\min}}\right)+1\right)c(T^*)$$

where $\widehat{T}$ is a greedy tree constructed using the above heuristic, and T^* is a tree with minimum expected cost. The cost of a tree T is defined as $c(T) := \mathbb{E}_\theta[c(T,\theta)]$, where $c(T,\theta_i)$ is the total cost of the queries made along the path from the root node to the leaf node ending in object θ_i.

Golovin *et al.* [28] simultaneously studied the problem of group identification, and, like us, used it in the context of object identification with persistent noise. They proposed an extension of the algorithm in [46] for group identification, and showed a logarithmic approximation similar to us. However, their result holds only when the priors π_i are rational. In addition, the bound achieved by modified GISA is marginally tighter than theirs.

IV. Object Identification Under Group Queries

In this section, we return to the problem of object identification. The input is a binary matrix $\mathbf{B}$ denoting the relationship between M objects and N queries, where the queries are grouped *a priori* into n disjoint categories, along with the *a priori* probability distribution Π on the objects. However, unlike the decision trees constructed in the previous two sections where the end user (e.g., a first responder) has to go through a fixed set of questions as dictated by the decision tree, here, the user is offered more flexibility in choosing the questions at each stage. More specifically, the decision tree suggests a query group from the n groups instead of a single query at each stage, and the user can choose a query to answer from the suggested query group.

A decision tree constructed with a group of queries at each stage has multiple branches at each internal node, corresponding to the size of the query group. Hence, a tree constructed in this fashion has multiple leaves ending in the same object. While traversing this decision tree, the user chooses the path at each internal node by selecting the query to answer from the given list of queries. Fig. 4 demonstrates a decision tree constructed in this fashion for the toy example shown in Fig. 3. The circled nodes correspond to the internal nodes, where each internal node is associated with a query group. The numbers associated with a dashed edge correspond to the probability that the user will choose that path over the others. The probability of reaching a node $a \in \mathcal{I}$ in the tree given $\theta \in \Theta_a$ is given by the product of the probabilities on the dashed edges along the path from the root node to that node, for example, the probability of reaching leaf node θ_1^* given $\theta = \theta_1$ in Fig. 4 is 0.45. The problem now is to select the query categories that will identify the object most efficiently, on average.

	Q^1		Q^2	
	0.5	0.5	0.9	0.1
	q_1	q_2	q_3	q_4
θ_1	0	1	1	0
θ_2	1	0	1	1
θ_3	1	1	0	1

Fig. 3. Toy Example 2.

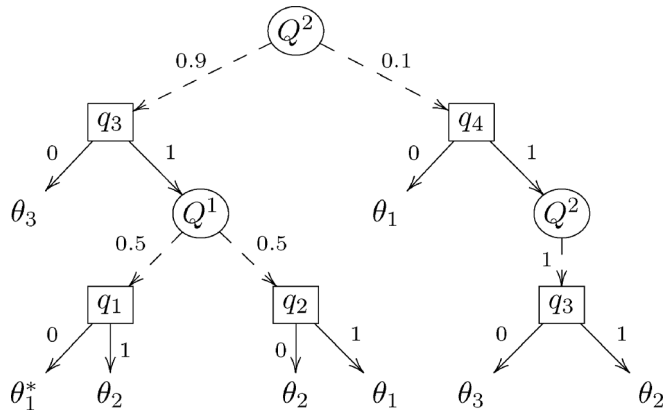

Fig. 4. Decision tree constructed on toy example 2 for object identification under group queries.

In addition to the terminology defined in Sections I-B and II, we also define $\mathbf{z} = (z_1, \ldots, z_N)$ to be the group labels of the queries, where $z_j \in \{1, \ldots, n\}, \forall j = 1, \ldots, N$. Let $\{Q^i\}_{i=1}^n$ be a partition of the query set Q, where Q^i denotes the set of queries in Q that belong to group i. Similarly, at any node 'a' in a tree, let Q_a^i and $\overline{Q_a^i}$ denote the set of queries in Q_a and $Q \setminus Q_a$ that belong to group i respectively. Let $p_i(q)$ be the *a priori* probability of the user selecting query $q \in Q^i$ at any node with query group i in the tree, where $\sum_{q \in Q^i} p_i(q) = 1$. In addition, at any node 'a' in the tree, the function $p_i(q) = 0, \forall q \in Q_a^i$, since the user would not choose a query which has already been answered, in which case $p_i(q)$ is renormalized. In our experiments we take $p_i(q)$ to be uniform on $\overline{Q_a^i}$. Finally, let $z_a \in \{1, \ldots, n\}$ denote the query group selected at an internal node 'a' in the tree and let $\tilde{p}_a$ denote the probability of reaching that node given $\theta \in \Theta_a$.

We denote an object identification problem with query groups by $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$. Given $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, let $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$ denote the set of decision trees that can uniquely identify all the objects in the set Θ with query groups at each internal node. For a decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, let $\{\rho_a(q)\}_{q \in Q^{z_a}}$ denote the reduction factors of all the queries in the query group at each internal node $a \in \mathcal{I}$ in the tree, where the reduction factors are treated as functions with input being a query.

Also, for a tree with L leaves, let $\mathcal{L}^i \subset \mathcal{L} = \{1, \ldots, L\}$ denote the set of leaves terminating in object θ_i and let d_j denote the depth of leaf node $j \in \mathcal{L}$. Then, the expected number of queries required to identify the unknown object using the given tree is equal to

$$\begin{aligned}\mathbb{E}[K(T)] &= \sum_{i=1}^{M} \Pr(\theta = \theta_i)\mathbb{E}[K(T)|\theta = \theta_i] \\ &= \sum_{i=1}^{M} \pi_i \left[\sum_{j \in \mathcal{L}^i} \tilde{p}_j d_j\right]\end{aligned}$$

Theorem 6: The expected number of queries required to identify an object using a tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$ is given by

$$\begin{aligned}\mathbb{E}[K(T)] = H(\Pi) + \sum_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \Bigg[1 \\ - \sum_{q \in Q^{z_a}} p_{z_a}(q) H(\rho_a(q))\Bigg]\end{aligned} \tag{11}$$

Proof: Special case of Theorem 7 below. ∎

Note from the above theorem, that given an object identification problem with group queries $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, the expected number of queries required to identify an object is lower bounded by its entropy $H(\Pi)$. Also, this lower bound can be achieved iff the reduction factors of all the queries in a query group at each internal node of the tree is equal to 0.5. In fact, Theorem 1 is a special case of the above theorem where each query group has just one query.

Given $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, the problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$ can be formulated as the following optimization problem:

$$\min_{T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})} \sum\nolimits_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \left[1 - \sum_{q \in Q^{z_a}} p_{z_a}(q) H(\rho_a(q))\right].$$

Note that here the reduction factors $\rho_a(q), \forall q \in Q^{z_a}$ and the prior probability function $p_{z_a}(q)$ depends on the query group $z_a \in \{1, \ldots, n\}$ chosen at node 'a' in the tree. The above optimization problem being a generalized version of the optimization problem in (2) is NP-complete. A greedy top-down local optimization of the above objective function yields a suboptimal solution where we choose a query group that minimizes the term $\Delta_a(j) := \left[1 - \sum_{q \in Q^j} p_j(q) H(\rho_a(q))\right]$ at each internal node, starting from the root node. The algorithm as summarized in Algorithm 3 below is referred to as GQSA (Group Queries Splitting Algorithm) in the rest of this paper.

Algorithm 3: Greedy decision tree algorithm for object identification with group queries

Group Queries Splitting Algorithm (GQSA)

Initialization : Let the leaf set consist of the root node

while some leaf node 'a' has $|\Theta_a| > 1$ **do**

for each query group with $\left|\overline{Q_a^j}\right| \geq 1$ **do**

Compute the prior probabilities of selecting queries within a group $p_j(q), \forall q \in Q^j$ at node 'a'

Compute the reduction factors for all the queries in the query group $\{\rho_a(q)\}_{q \in Q^j}$

Compute the cost $\Delta_a(j)$ of using query group j at node 'a'

end

Choose a query group j with the least cost $\Delta_a(j)$ at node 'a'

Form the left and the right child nodes for all queries with $p_j(q) > 0$ in the query group

end

Comment: In this section and the one following, we assume that the query groups are disjoint only for the sake of simplicity. However, we do not need this assumption for the results in Theorem 6, and Theorem 7 in the next section, to hold. Similarly, we assume that the prior probability of choosing a query from a query group depends only on the group membership. However, one could use a more complex prior distribution that not only depends on the group membership, but also on the previous queries and their responses. The results in Theorems 6 and 7 do not change by these generalizations, as long as the prior distribution is normalized and sums to 1 at each internal node in the tree. This can be readily observed from the proof of Theorem 7 in Appendix D.

V. Group Identification Under Group Queries

For the sake of completion, we consider here the problem of identifying the group of an unknown object $\theta \in \Theta$ under group queries. The input is a binary matrix $\mathbf{B}$ denoting the relationship between M objects and N queries, where the objects are grouped into m groups and the queries are grouped into n groups. The task is to identify the group of an unknown object through as few queries from Q as possible where, at each stage, the user is offered a query group from which a query is chosen.

As noted in Section III, a decision tree constructed for group identification can have multiple objects terminating in the same leaf node. Also, a decision tree constructed for group identification with a query group at each internal node has multiple leaves terminating in the same group. Hence a decision tree constructed in this section can have multiple objects terminating in the same leaf node and multiple leaves terminating in the same group. Also, we use most of the terminology defined in Sections III and IV here.

We denote a group identification problem with query groups by $(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$ where $\mathbf{y} = (y_1, \ldots, y_M)$ denotes the group

labels on the objects, $\mathbf{z} = (z_1, \ldots, z_N)$ denotes the group labels on the queries and $\mathbf{p} = (p_1(q), \ldots, p_n(q))$ denotes the *a priori* probability functions of selecting queries within query groups. Given a group identification problem under group queries $(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$, let $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$ denote the set of decision trees that can uniquely identify the groups of all objects in the set Θ with query groups at each internal node. For any decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$, let $\{\rho_a(q)\}_{q \in Q^{z_a}}$ denote the reduction factor set and let $\{\{\rho_a^i(q)\}_{i=1}^m\}_{q \in Q^{z_a}}$ denote the group reduction factor sets at each internal node $a \in \mathcal{I}$ in the tree, where $z_a \in \{1, \ldots, n\}$ denotes the query group selected at that node.

Also, for a tree with L leaves, let $\mathcal{L}^i \subset \mathcal{L} = \{1, \ldots, L\}$ denote the set of leaves terminating in object group i and let $d_j, \tilde{p}_j$ denote the depth of leaf node $j \in \mathcal{L}$ and the probability of reaching that node given $\theta \in \Theta_j$, respectively. Then, the expected number of queries required to identify the group of an unknown object using the given tree is equal to

$$\begin{aligned}\mathbb{E}[K(T)] &= \sum_{i=1}^{m} \Pr(\theta \in \Theta^i)\mathbb{E}[K(T)|\theta \in \Theta^i] \\ &= \sum_{i=1}^{m} \pi_{\Theta^i} \left[\sum_{j \in \mathcal{L}^i} \frac{\pi_{\Theta_j}}{\pi_{\Theta^i}} \tilde{p}_j d_j\right].\end{aligned}$$

Theorem 7: The expected number of queries required to identify the group of an unknown object using a tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$ is given by

$$\begin{aligned}\mathbb{E}[K(T)] = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \Bigg\{1 - \sum_{q \in Q^{z_a}} p_{z_a}(q) \\ \left[H(\rho_a(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i(q))\right]\Bigg\} \qquad (12)\end{aligned}$$

where $\Pi_{\mathbf{y}}$ denotes the probability distribution of the object groups induced by the labels $\mathbf{y}$, i.e., $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \ldots, \pi_{\Theta^m})$

Proof: The proof is given in Appendix D. ■

Note that Theorems 1, 2, and 6 are special cases of the above theorem. This theorem states that, given a group identification problem under group queries $(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$, the expected number of queries required to identify the group of an object is lower bounded by the entropy of the probability distribution of the object groups $H(\Pi_{\mathbf{y}})$. It also follows from the above theorem that this lower bound can be achieved iff the reduction factors and the group reduction factors of all the queries in a query group at each internal node are equal to 0.5 and 1, respectively.

The problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$ can be formulated as the following optimization problem:

$$\begin{aligned}\min_{T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})} \sum_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \Bigg\{1 - \sum_{q \in Q^{z_a}} p_{z_a}(q) \\ \left[H(\rho_a(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i(q))\right]\Bigg\}.\end{aligned}$$

Algorithm 4: Greedy decision tree algorithm for group identification under group queries

Group Identification under Group Queries

Splitting Algorithm (GIGQSA)

Initialization : Let the leaf set consist of the root node

while some leaf node 'a' has more than one group of objects **do**

for each query group with $\left|\overline{Q_a^j}\right| \geq 1$ **do**

Compute the prior probabilities of selecting queries within a group, $p_j(q), \forall q \in Q^j$ at node 'a'

Compute the reduction factors for all the queries in the query group $\{\rho_a(q)\}_{q \in Q^j}$

Compute the group reduction factors for all the queries in the query group $\{\rho_a^i(q)\}_{q \in Q^j}, \forall i = 1, \ldots, m$

Compute the cost $\Delta_a(j)$ of using query group j at node 'a'

end

Choose a query group j with the least cost $\Delta_a(j)$ at node 'a'

Form the left and the right child nodes for all queries with $p_j(q) > 0$ in the query group

end

Note that here the reduction factors $\{\rho_a(q)\}_{q \in Q^{z_a}}$, the group reduction factors $\{\rho_a^i(q)\}_{q \in Q^{z_a}}$ for all $i = 1, \ldots, m$, and the prior probability function $p_{z_a}(q)$ depends on the query group $z_a \in \{1, \ldots, n\}$ chosen at node 'a' in the tree. Once again, the above optimization problem being a generalized version of the optimization problem in (2) is NP-complete. A greedy top-down optimization of the above objective function yields a suboptimal solution where we choose a query group that minimizes the term $\Delta_a(j) := 1 - \sum_{q \in Q^j} p_j(q)\left[H(\rho_a(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i(q))\right]$ at each internal node, starting from the root node. The algorithm as summarized in Algorithm 4 above is referred to as Group Identification under Group Queries Splitting Algorithm (GIGQSA).

VI. Object Identification Under Persistent Noise

We now consider the problem of rapidly identifying an unknown object $\theta \in \Theta$ in the presence of persistent query noise, and relate this problem to group identification. Query noise refers to errors in the query responses, i.e., the observed query response is different from the true response of the unknown object. For example, a victim of toxic chemical exposure may not report a symptom because of a delayed onset of that symptom. Unlike the noise model often assumed in the literature, where repeated querying results in independent realizations of the noise, persistent query noise is a more stringent noise model where repeated queries results in the same response.

Before we address this problem, we need to introduce some additional notation. Given an object identification problem

prone to error	q_1 ×	q_2 ✓	q_3 ✓	Π
θ_1	0	0	0	$\frac{1}{4}$
θ_2	1	1	1	$\frac{3}{4}$

(a)

	q_1	q_2	q_3	$\widetilde{\Pi}_1(p=0.5)$	$\widetilde{\Pi}_2(p=0.25)$
Θ^1	0	0	0	$\frac{1}{12}$	$\frac{3}{20}$
	1	0	0	0	0
	0	1	0	$\frac{1}{12}$	$\frac{1}{20}$
	0	0	1	$\frac{1}{12}$	$\frac{1}{20}$
Θ^2	1	1	1	$\frac{1}{4}$	$\frac{9}{20}$
	0	1	1	0	0
	1	0	1	$\frac{1}{4}$	$\frac{3}{20}$
	1	1	0	$\frac{1}{4}$	$\frac{3}{20}$

(b)

Fig. 5. For the toy example shown in (a) consisting of 2 objects and 3 queries with an $\epsilon = 1$. (b) Demonstrates the construction of matrix $\widetilde{\mathbf{B}}$. The probability distribution of the objects in $\widetilde{\mathbf{B}}$ are generated using the noise model described in Section VI-B , where only queries q_2 and q_3 are assumed to be prone to error.

$(\mathbf{B}, \Pi)$, let δ denote the minimum Hamming distance between any two rows of the matrix $\mathbf{B}$. Also, we refer to the bit string consisting of observed query responses as an input string. The input string can differ from the true bit string (corresponding to the row vector of the true object in matrix $\mathbf{B}$) due to persistent query noise. However, we further assume that the number of query responses in error cannot exceed $\epsilon := \lfloor \frac{\delta-1}{2} \rfloor$ for the unknown object to be uniquely identified in the presence of noise. Given this noise setting, the goal of object identification under persistent noise is to uniquely identify the unknown object θ using as few queries as possible.

This problem can be posed as a group identification problem as follows: Given an object identification problem $(\mathbf{B}, \Pi)$ with M objects and N queries that is susceptible to ϵ errors, create $(\widetilde{\mathbf{B}}, \widetilde{\Pi})$ with M groups of objects and N queries, where each object group in this new matrix is formed by considering all possible bit strings that differ from the original bit string in at most ϵ positions, i.e., the size of each object group in $\widetilde{\mathbf{B}}$ is $\sum_{e=0}^{\epsilon} \binom{N}{e}$. Fig. 5(b) demonstrates construction of $\widetilde{\mathbf{B}}$ for the toy example shown in Fig. 5(a) consisting of 2 objects and 3 queries with an $\epsilon = 1$.

Each bit string in the object set Θ^i of $\widetilde{\mathbf{B}}$ corresponds to one of the possible input strings when the true object is θ_i and at most ϵ errors occur. Also note that, by definition of ϵ, no two bit strings in the matrix $\widetilde{\mathbf{B}}$ can be the same. Thus, the problem of rapidly identifying an unknown object θ from $(\mathbf{B}, \Pi)$ in the presence of at most ϵ persistent errors, reduces to the problem of identifying the group of the unknown object from $(\widetilde{\mathbf{B}}, \widetilde{\Pi})$. The probability distribution $\widetilde{\Pi}$ of the bit strings in $\widetilde{\mathbf{B}}$ depends on the prior Π and the error model. In the following section, we describe one specific error model that arises commonly in applications such as active learning, image processing and computer vision, and demonstrate the computation of $\widetilde{\Pi}$ under that error model.

Given that this problem can be reduced to a group identification problem, the unknown object can be rapidly identified in the presence of persistent query noise using any group identification algorithm including GISA and modified GISA. In addition, the near-optimal property of modified GISA guarantees that the expected number of queries required to identify an unknown object under persistent noise is logarithmically close to that of an optimal algorithm, as stated in the result below.

Corollary 3: Let $(\mathbf{B}, \Pi)$ denote an object identification problem that is susceptible to ϵ persistent errors. Let $\widehat{K}$ denote the expected number of queries required to identify an unknown object under persistent noise using modified GISA, and let K^* denote the expected number of queries required by an optimal algorithm. Then it holds that

$$\widehat{K} \leq \left(2 \ln \left(\frac{1}{\sqrt{3}\widetilde{\pi}_{\min}}\right) + 1\right) K^*,$$

where $\widetilde{\pi}_{\min} = \min\{\widetilde{\pi} \in \widetilde{\Pi} : \widetilde{\pi} > 0\}$.

Proof: The result follows from Theorem 5. ■

A. Constant Noise Rate

We now consider a noise model that has been used in the context of pool-based active learning with a faulty oracle [30], [34], experimental design [31], computer vision, and image processing [47], where the responses to some queries are assumed to be randomly flipped.

We will describe a general version of this noise model. Given N queries, consider the case where a fraction ν of them are prone to error. The query response to each of these νN queries can be in error with a probability $0 \leq p \leq 0.5$, where the errors occur independently. Then, the probability of e errors occurring is given by

$$\Pr(e \text{ errors}) = \frac{\binom{N\nu}{e} p^e (1-p)^{N\nu - e}}{\sum_{e'=0}^{\epsilon'} \binom{N\nu}{e'} p^{e'} (1-p)^{N\nu - e'}}, \quad 0 \leq e \leq \epsilon'$$

where $\epsilon' := \min(\epsilon, N\nu)$ denotes the maximum number of persistent errors that could occur. Note that this probability model corresponds to a truncated binomial distribution.

Given an object identification problem $(\mathbf{B}, \Pi)$ that is susceptible to ϵ errors, let $\widetilde{\mathbf{B}}$ denote the extended binary matrix constructed as described in Section VI. The probability distribution $\widetilde{\Pi}$ of the objects in $\widetilde{\mathbf{B}}$ can be computed as follows. For an object belonging to group i in $\widetilde{\mathbf{B}}$, if its response to a query that is not prone to error differs from the true response of object θ_i in $\mathbf{B}$, then the probability $\widetilde{\pi}$ of that object in $\widetilde{\mathbf{B}}$ is 0. On the other hand, if its response differs in $e \leq \epsilon'$ queries that are prone to error, then its probability is given by

$$\frac{p^e (1-p)^{N\nu - e}}{\sum_{e'=0}^{\epsilon'} \binom{N\nu}{e'} p^{e'} (1-p)^{N\nu - e'}} \, \pi_i.$$

Fig. 5(b) shows the probability distribution of the objects in $\widetilde{\mathbf{B}}$ using the probability model described above with $p = 0.5$ $(\widetilde{\Pi}_1)$ and $p = 0.25$ $(\widetilde{\Pi}_2)$ for the toy example shown in Fig. 5(a) where only queries q_2 and q_3 are prone to error.

However, one possible concern with this approach for object identification under persistent noise could be a memory related

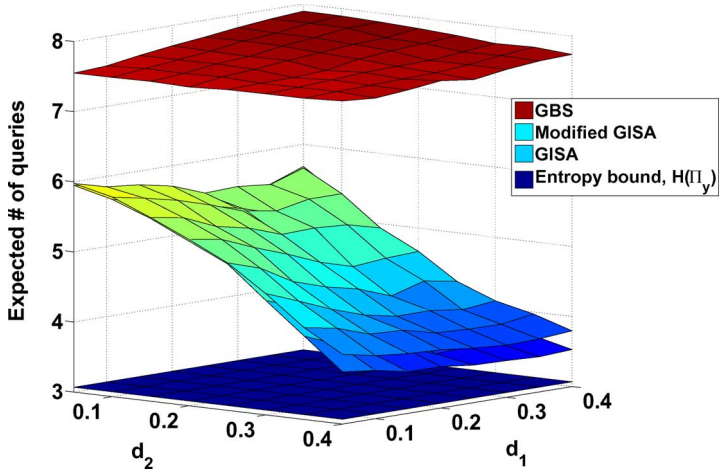

Fig. 6. Expected number of queries required to identify the group of an object using GBS, GISA and modified GISA on random datasets generated using the proposed random data model. Note that GISA and modified GISA achieve almost similar performance on these datasets, with GISA performing slightly better than modified GISA.

issue of explicitly maintaining the matrix $\widetilde{\mathbf{B}}$ due to the combinatorial explosion in its size. Interestingly, for the noise model described here, the relevant quantities for query selection in GBS, GISA, and modified GISA (i.e., the reduction factors) can be efficiently computed without explicitly constructing the matrix $\widetilde{\mathbf{B}}$, described in detail in Appendix E.

VII. Experiments

We perform three sets of experiments, demonstrating our algorithms for group identification, object identification using query groups, and object identification with persistent noise. In each case, we compare the performances of the proposed algorithms to standard algorithms such as the splitting algorithm, using synthetic data as well as a real dataset, the WISER database. The WISER database is a toxic chemical database describing the binary relationship between 298 toxic chemicals and 79 acute symptoms. The symptoms are grouped into 10 categories (e.g., neurological, cardio) as determined by NLM, and the chemicals are grouped into 16 categories (e.g., pesticides, corrosive acids) as determined by a toxicologist and a Hazmat expert.

A. Group Identification

Here, we consider a group identification problem $(\mathbf{B}, \Pi)$ where the objects are grouped into m groups given by $\mathbf{y} = (y_1, \ldots, y_M)$, $y_i \in \{1, \ldots, m\}$, with the task of identifying the group of an unknown object from the object set Θ through as few queries from Q as possible. First, we consider random datasets generated using a random data model and compare the performances of GBS, GISA, and modified GISA for group identification in these random datasets. Then, we compare the performance of these algorithms on the WISER database. In both these experiments, we assume a uniform *a priori* probability distribution on the objects.

1) Random Datasets: We consider random datasets of the same size as the WISER database, with 298 objects and 79 queries where the objects are grouped into 16 classes with the same group sizes as that in the WISER database. We associate each query in a random dataset with two parameters, $\gamma_w \in [0.5, 1]$ which reflects the correlation of the object responses *within* a group, and $\gamma_b \in [0.5, 1]$ which captures the correlation of the object responses *between* groups. When γ_w is close to 0.5, each object within a group is equally likely to exhibit 0 or 1 as its response to the query, whereas, when γ_w is close to 1, most of the objects within a group are highly likely to exhibit the same response to the query. Similarly, when γ_b is close to 0.5, each group is equally likely to exhibit 0 or 1 as its response to the query, where a group response corresponds to the majority vote of the object responses within a group, while, as γ_b tends to 1, most of the groups are highly likely to exhibit the same response.

Given a (γ_w, γ_b) pair for a query in a random dataset, the object responses for that query are created as follows.

1) Generate a Bernoulli random variable x
2) For each group $i \in \{1, \ldots, m\}$, assign a binary label b_i, where $b_i = x$ with probability γ_b
3) For each object in group i, assign b_i as the object response with probability γ_w

Given the correlation parameters $(\gamma_w(q), \gamma_b(q)) \in [0.5, 1]^2, \forall q \in Q$, a random dataset can be created by following the above procedure for each query. Conversely, we describe in Section VII-A.II on how to estimate these parameters for a given dataset.

Fig. 6 compares the mean $\mathbb{E}[K(T)]$ for GBS, GISA, and modified GISA in 100 randomly generated datasets (for each value of d_1 and d_2), where the random datasets are created such that the query parameters are uniformly distributed in the rectangular space governed by d_1, d_2 as shown in Fig. 7. This demonstrates the improved performance of GISA and modified GISA over GBS in group identification. Especially, note that $\mathbb{E}[K(T)]$ tends close to the entropy bound $H(\Pi_{\mathbf{y}})$ using both GISA and modified GISA as d_2 increases.

This is due to the increment in the number of queries in the fourth quadrant of the parameter space as d_2 increases. Specifically, as the correlation parameters γ_w, γ_b tends to 1 and 0.5, respectively, choosing that query eliminates approximately half

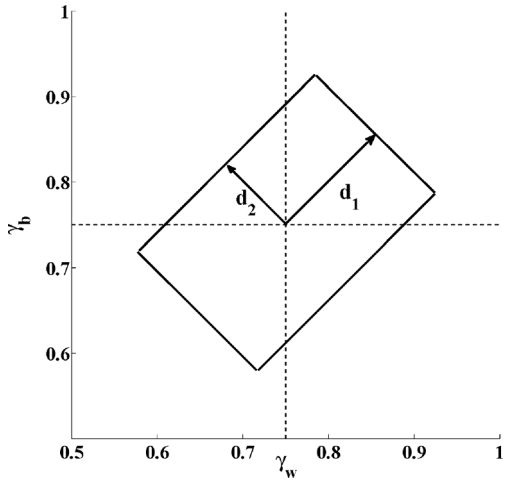

Fig. 7. Random data model – The query parameters $(\gamma_w(q), \gamma_b(q))$ are restricted to lie in the rectangular space.

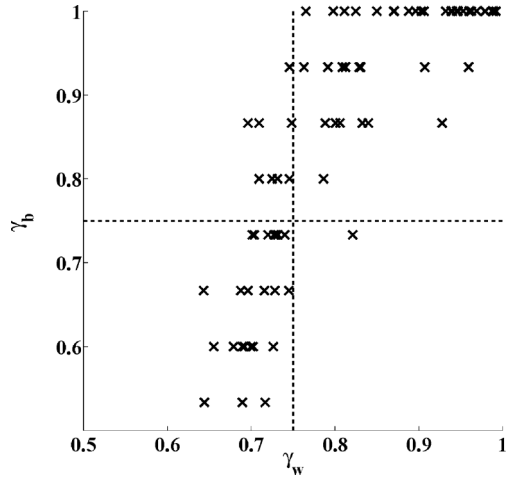

Fig. 8. Scatter plot of the query parameters in the WISER database.

TABLE I
EXPECTED NUMBER OF QUERIES REQUIRED TO IDENTIFY THE GROUP OF AN OBJECT IN WISER DATABASE

Algorithm	$\mathbb{E}[K(T)]$
modified GISA	7.291 $\pm$ 0.001
GISA	7.792 $\pm$ 0.001
GBS	7.948 $\pm$ 0.003
Random Search	16.328 $\pm$ 0.177

the groups with each group being either completely eliminated or completely included, i.e., the group reduction factors tend to 1 for these queries. Such queries are preferable in group identification with both GISA and modified GISA being specifically designed to search for those queries leading to their strikingly improved performance over GBS as d_2 increases.

2) Wiser Database: Table I compares the expected number of queries required to identify the group of an unknown object in the WISER database using GISA, modified GISA, GBS, and random search, where the group entropy in the WISER database is given by $H(\Pi_{\mathbf{y}}) = 3.068$. The table reports the 95% symmetric confidence intervals based on random trails, where the randomness in GISA, modified GISA and GBS is due to the presence of multiple best splits at each internal node.

However, the improvement of both GISA and modified GISA over GBS on WISER is less than was observed for many of the random datasets discussed above. To understand this, we developed a method to estimate the correlation parameters of the queries for a given dataset $\mathbf{B}$. For each query in the dataset, the correlation parameters can be estimated as follows.

1) For every group $i \in \{1, \ldots, m\}$, let b_i denote the group response given by the majority vote of object responses in the group and let $\widehat{\gamma_w^i}$ denote the fraction of objects in the group with similar response as b_i
2) Denote by a binary variable x, the majority vote of the group responses $\mathbf{b} = [b_1, \ldots, b_m]$
3) Then, $\widehat{\gamma_b}$ is given by the fraction of groups with similar response as x, and $\widehat{\gamma_w} = \frac{1}{m}\sum_i \widehat{\gamma_w^i}$

Now, we use the above procedure to estimate the query parameters for all queries in the WISER database, shown in Fig. 8. Note from this figure that there is just one query in the fourth quadrant of the parameter space and there are no queries with γ_w close to 1 and γ_b close to 0.5. In words, chemicals in the same group tend to behave differently and chemicals in different groups tend to exhibit similar response to the symptoms. This is a manifestation of the non-specificity of the symptoms in the WISER database as reported by Bhavnani *et al.* [42].

B. Object Identification Under Query Classes

In this section, we consider an object identification problem under group queries $(\mathbf{B}, \Pi)$ where the queries are *a priori* grouped into n groups given by $\mathbf{z} = (z_1, \ldots, z_N), z_i \in \{1, \ldots, n\}$, with the task of identifying an unknown object from the set Θ through as few queries from Q as possible, where the user is presented with a query group at each stage to choose from. Note that this approach is midway between a complete active search strategy and a complete passive search strategy. Hence, we primarily compare the performance of GQSA to a completely active search strategy such as GBS and a completely passive search strategy like random search where the user randomly chooses the queries from the set Q to answer. In addition, we also compare GQSA to other possible heuristics where we choose a query group i that minimizes $\min_{q\in Q^i} p_i(q)\rho_a(q)$ or $\max_{q\in Q^i} p_i(q)\rho_a(q)$ at each internal node 'a.'

First, we compare the performances of these algorithms on random datasets generated using a random data model. Then, we compare them in the WISER database. In both these experiments, we assume uniform *a priori* probability distribution on the objects as well as on queries within a group. The latter probability distribution corresponds to the probability of a user selecting a particular query q from a query group, $p_i(q), \forall i = 1, \ldots, n$.

1) Random Datasets: Here, we consider random datasets of the same size as the WISER database, with 298 objects and 79 queries where the queries are grouped into 10 groups with the same group sizes as that in the WISER database. We associate a random dataset with a parameter $\gamma_{max} \in [0.5, 1]$, where γ_{max} corresponds to the maximum permissible value of γ_b for a query in the random dataset. Given a γ_{max}, a random dataset is created as follows.

1) For each query group, generate a $\gamma_b \in [0.5, \gamma_{max}]$

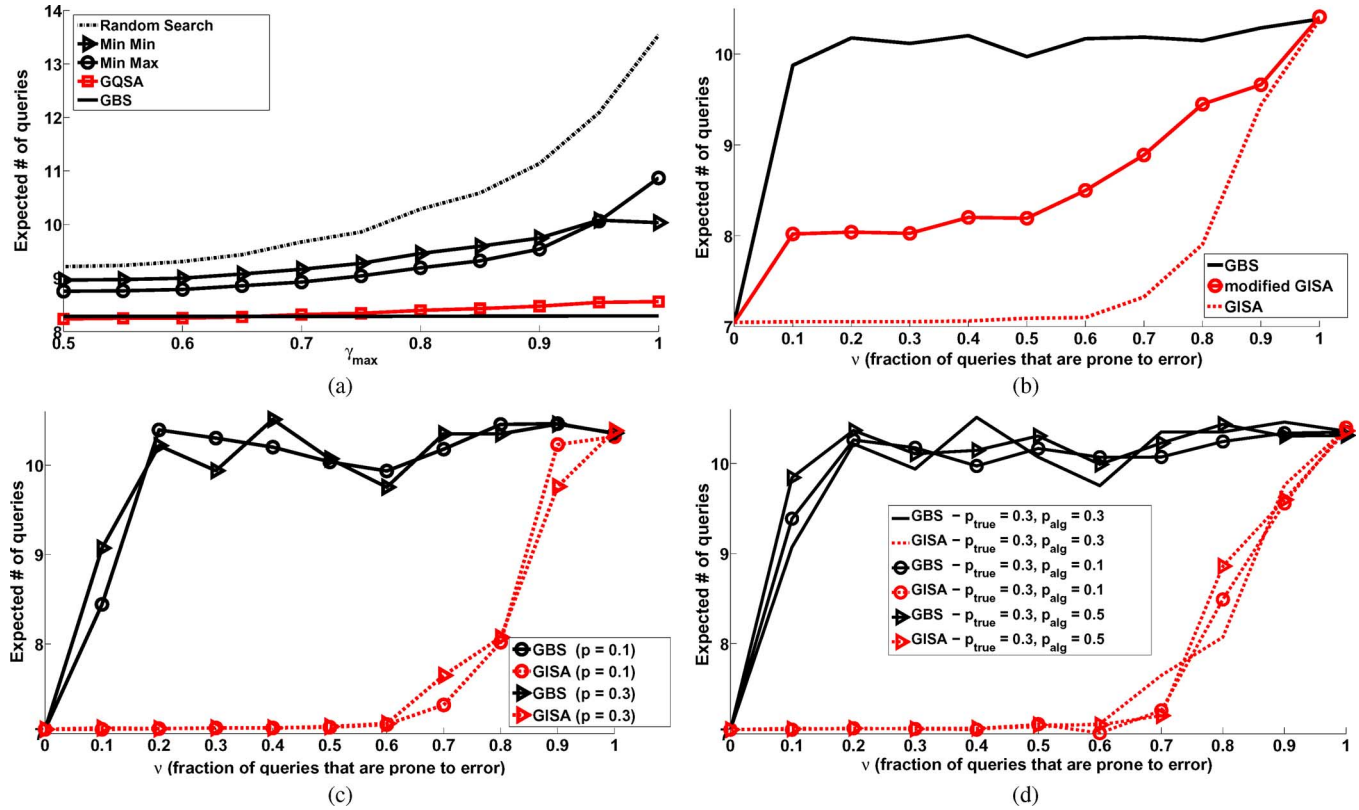

Fig. 9. (a) Compares the average query complexity of different algorithms for object identification under group queries in random datasets. (b) Compares the performance of GBS, modified GISA and GISA in identifying the true object in the presence of persistent query noise described in Section VI-B with $p = 0.5$. (c) Compares the same for different values of p. (d) Compares the performance of GBS and GISA under persistent noise in the presence of discrepancies between the true value of p, p_{true} and the value used in the algorithm p_{alg}.

TABLE II
EXPECTED NUMBER OF QUERIES REQUIRED TO IDENTIFY AN OBJECT UNDER GROUP QUERIES IN WISER DATABASE

Algorithm	$\mathbb{E}[K(T)]$
GBS	8.283 ± 0.000
GQSA	11.360 ± 0.096
$\min_i \min_{q \in Q^i} p_i(q)\rho_a(q)$	13.401 ± 0.116
$\min_i \max_{q \in Q^i} p_i(q)\rho_a(q)$	18.697 ± 0.357
Random Search	20.251 ± 0.318

2) For each query in the query group, generate a Bernoulli random variable x and give each object the same query label as x with probability γ_b

Fig. 9(a) compares the mean $\mathbb{E}[K(T)]$ for the respective algorithms in 100 randomly generated datasets, for each value of γ_{max}. The $\min\min$ corresponds to the heuristic where we minimize $\min_{q \in Q^i} p_i(q)\rho_a(q)$ at each internal node and the $\min\max$ corresponds to the heuristic where we minimize $\max_{q \in Q^i} p_i(q)\rho_a(q)$. Note from the figure that in spite of not being a completely active search strategy, the performance of GQSA is comparable to that of GBS and better than the other algorithms.

2) Wiser Database: Table II compares the expected number of queries required to identify an unknown object under group queries in the WISER database using the respective algorithms, where the entropy of the objects in the WISER database is given by $H(\Pi) = 8.219$. The table reports the 95% symmetric confidence intervals based on random trials, where the randomness in GBS is due to the presence of multiple best splits at each internal node.

Once again, it is not surprising that GBS outperforms GQSA as GBS is fully active, i.e., it always chooses the best split, whereas GQSA does not always pick the best split, since a human is involved. Yet, the performance of GQSA is not much worse than that of GBS. In fact, if we were to fully model the time-delay associated with answering a query, then GQSA might have a smaller "time to identification," because presumably it would take less time to answer the queries on average.

C. Object Identification Under Persistent Noise

In Section VI, we showed that identifying an unknown object in the presence of persistent query noise can be reduced to a group identification problem. Hence, any group identification algorithm can be adopted to solve this problem. Here, we compare the performance of GBS, GISA, and modified GISA under the noise model described in Section VI-B.

Note that this noise model requires the knowledge of the $N\nu$ queries from the set Q that are prone to error. We assume this knowledge in all our experiments in this section. Below, we show the procedure adopted to simulate the error model.

1) Select the fraction ν of the N queries that are prone to error.
2) Generate $e \in \{0, \ldots, \epsilon'\}$ according to the selected probability model (p value).
3) Choose e queries from the above $N\nu$ set of queries.
4) Flip the object responses of these e queries in the true object.

We compare the performance of GBS, GISA and modified GISA on a subset of the WISER database consisting of 131

toxic chemicals and 79 symptom queries with $\epsilon = 2$. Fig. 9(b) shows the expected number of queries required by GBS, GISA and modified GISA to identify the true object in the presence of a maximum of ϵ persistent errors for different values of ν, when the probability of query error p is 0.5. Note that except for the extreme cases where $\nu = 0$ and $\nu = 1$, GISA and modified GISA have great improvement over GBS. When $\nu = 0, 1$, GBS, GISA and modified GISA reduce to the same algorithm. Similar performance has been observed for different values of p as shown in Fig. 9(c). However, we do not show modified GISA in this figure to avoid cramping.

Also, note that to compute the probability distribution $\widetilde{\Pi}$ of the objects in the extended matrix $\widetilde{\mathbf{B}}$, we require the knowledge of p. Though this probability can be estimated with the help of external knowledge sources beyond the database such as domain experts, user surveys or by analyzing past query logs, the estimated value of p can vary slightly from its true value. Hence, we tested the sensitivity of the three algorithms to error in the value of p and noted that there is not much change in their performance to discrepancies in the value of p as shown in Fig. 9(d). Once again, we do not show the results of modified GISA to avoid cramping.

VIII. Conclusions and Future Work

In this paper, we developed algorithms that broaden existing methods for object identification to incorporate factors that are specific to a given task and environment. These algorithms are greedy algorithms derived in a common, principled framework, and extend Shannon-Fano coding to group-based settings. While our running example has been toxic chemical identification, the methods presented are applicable to a much broader class of applications, such as other forms of emergency response, pool-based active learning, disease diagnosis or network failure diagnosis.

In a series of experiments on synthetic data and a toxic chemical database, we demonstrated the effectiveness of our algorithms relative to the standard splitting algorithm, also known as generalized binary search (GBS), which is the most commonly studied algorithm for object identification. In some settings, our algorithms outperform GBS by drastic amounts. Furthermore, in the case of group identification, we propose a near-optimal greedy algorithm that achieves a logarithmic approximation to the optimal solution.

While this work is a step toward making object identification algorithms better suited to real-world identification tasks, there are many other issues that deserve to be examined in future work. These include challenges such as multiple objects present, probabilities of query response or query noise, or user confidence. In the problem of object identification under persistent noise, our approach can only recover from a restricted number of query errors, depending on the minimum Hamming distance between objects. While this assumption is required if we desire unique identification of the unknown object, it would be interesting to loosen this assumption by pursuing a slightly less ambitious goal. Additionally, instead of minimizing the expected number of queries required for object/group identification, it would be valuable to develop a similar framework that minimizes the number of queries in the worst case, thereby eliminating dependence on the prior probabilities (see [27]).

Appendix A
Proof of Theorem 3

Let T_a denote a subtree from any node 'a' in the tree T and let $\mathcal{L}_a$ denote the set of leaf nodes in this subtree. Then, let μ_a denote the expected depth of the leaf nodes in this subtree, given by

$$\mu_a = \sum_{j \in \mathcal{L}_a} \frac{\pi_{\Theta_j}}{\pi_{\Theta_a}} d_j^a$$

where d_j^a corresponds to the depth of leaf node j in the subtree T_a, and let H_a denote the entropy of the probability distribution of the classes at the root node of the subtree T_a, i.e.,

$$H_a = -\sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}.$$

Now, we show using induction that for any subtree T_a in the tree T, the following relation holds:

$$\pi_{\Theta_a}\mu_a - \pi_{\Theta_a}H_a = \sum_{s \in \mathcal{I}_a} \pi_{\Theta_s}\Big[1 - H(\rho_s) + \sum_{i=1}^{m} \frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}} H(\rho_s^i)\Big] - \sum_{s \in \mathcal{L}_a} \pi_{\Theta_s} I(\Theta_s)$$

where $\mathcal{I}_a, \mathcal{L}_a$ denotes the set of internal nodes and the set of leaf nodes in the subtree T_a respectively.

The relation holds trivially for any subtree rooted at a leaf node of the tree T with both the left hand side and the right hand side of the expression equal to $-\pi_{\Theta_a} I(\Theta_a)$ (Note from (6) that $I(\Theta_a) = H_a$). Now, assume the above relation holds for the subtrees rooted at the left and right child nodes of node 'a.' Then, using Lemma 1 we have

$$\begin{aligned}
&\pi_{\Theta_a}[\mu_a - H_a] \\
&\quad = \pi_{\Theta_{l(a)}}[\mu_{l(a)} - H_{l(a)}] + \pi_{\Theta_{r(a)}}[\mu_{r(a)} - H_{r(a)}] \\
&\qquad + \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right] \\
&\quad = \sum_{s \in \mathcal{I}_{l(a)}} \pi_{\Theta_s}\left[1 - H(\rho_s) + \sum_{i=1}^{m} \frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}} H(\rho_s^i)\right] \\
&\qquad + \sum_{s \in \mathcal{I}_{r(a)}} \pi_{\Theta_s}\left[1 - H(\rho_s) + \sum_{i=1}^{m} \frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}} H(\rho_s^i)\right] \\
&\qquad + \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right] \\
&\qquad - \sum_{s \in \mathcal{L}_{l(a)}} \pi_{\Theta_s} I(\Theta_s) - \sum_{s \in \mathcal{L}_{r(a)}} \pi_{\Theta_s} I(\Theta_s) \\
&\quad = \sum_{s \in \mathcal{I}_a} \pi_{\Theta_s}\left[1 - H(\rho_s) + \sum_{i=1}^{m} \frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}} H(\rho_s^i)\right] \\
&\qquad - \sum_{s \in \mathcal{L}_a} \pi_{\Theta_s} I(\Theta_s)
\end{aligned}$$

thereby completing the induction. Finally, the result follows by applying the relation to the tree T whose probability mass at the root node $\pi_{\Theta_a} = 1$.

Lemma 1:

$$\begin{aligned}&\pi_{\Theta_a}[\mu_a - H_a]\\ &= \pi_{\Theta_{l(a)}}[\mu_{l(a)} - H_{l(a)}] + \pi_{\Theta_{r(a)}}[\mu_{r(a)} - H_{r(a)}]\\ &\quad + \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right]\end{aligned}$$

Proof: We first note that $\pi_{\Theta_a}\mu_a$ for a subtree T_a can be decomposed as

$$\begin{aligned}\pi_{\Theta_a}\mu_a &= \sum_{j\in\mathcal{L}_a} \pi_{\Theta_j} d_j^a\\ &= \sum_{j\in\mathcal{L}_{l(a)}} \pi_{\Theta_j} d_j^a + \sum_{j\in\mathcal{L}_{r(a)}} \pi_{\Theta_j} d_j^a\\ &= \sum_{j\in\mathcal{L}_{l(a)}} \pi_{\Theta_j}(d_j^a - 1) + \sum_{j\in\mathcal{L}_{r(a)}} \pi_{\Theta_j}(d_j^a - 1)\\ &\quad + \sum_{j\in\mathcal{L}_a} \pi_{\Theta_j}\\ &= \pi_{\Theta_{l(a)}}\mu_{l(a)} + \pi_{\Theta_{r(a)}}\mu_{r(a)} + \pi_{\Theta_a}. \end{aligned} \tag{13}$$

Similarly, $\pi_{\Theta_a} H_a$ can be decomposed as

$$\begin{aligned}\pi_{\Theta_a} H_a &= \sum_{i=1}^{m} \pi_{\Theta_a^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_a^i}}\\ &= \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_a^i}} + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_a^i}}\\ &= \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_{l(a)}^i}} + \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_{l(a)}^i}}{\pi_{\Theta_a^i}}\\ &\quad + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_{r(a)}^i}} + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_{r(a)}^i}}{\pi_{\Theta_a^i}}\\ &\quad + \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{l(a)}}} + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{r(a)}}}\\ &= \pi_{\Theta_{l(a)}} H_{l(a)} + \pi_{\Theta_{r(a)}} H_{r(a)}\\ &\quad - \sum_{i=1}^{m} \left[\pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_{l(a)}^i}} + \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_{r(a)}^i}}\right]\\ &\quad + \left[\pi_{\Theta_{l(a)}} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{l(a)}}} + \pi_{\Theta_{r(a)}} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{r(a)}}}\right]\\ &= \pi_{\Theta_{l(a)}} H_{l(a)} + \pi_{\Theta_{r(a)}} H_{r(a)}\\ &\quad - \sum_{i=1}^{m} \pi_{\Theta_a^i} H(\rho_a^i) + \pi_{\Theta_a} H(\rho_a). \end{aligned} \tag{14}$$

The result follows from (13) and (14) above. ■

Appendix B
Proof of Theorem 4

From (14) in Lemma 1, we have

$$\begin{aligned}H_a - &\left[\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} H_{l(a)} + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} H_{r(a)}\right]\\ &= -\left[-H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right].\end{aligned}$$

Thus, maximizing the impurity based objective function with entropy function as the impurity function is equivalent to minimizing the cost function $\Delta_a := 1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)$.

Appendix C
Proof of Theorem 5

Before we prove the result in Theorem 5, we need to introduce some additional notation and review some definitions from [13]. Let $f : 2^Q \times \Theta \to \mathbb{R}_{\geq 0}$ be a utility/reward function that depends on the queries chosen and the unknown object $\theta \in \Theta$. For any $\mathcal{A} \subseteq \{1, \ldots, N\}$, let $Q_{\mathcal{A}}$ denote the subset of queries indexed by $\mathcal{A}$, and let $\mathbf{Z}_{\mathcal{A}}$ be a binary random vector denoting the responses to queries in $Q_{\mathcal{A}}$. In addition, given a tree T, let $Q(T, \theta_i)$ denote the queries made along the path from the root node to the leaf node terminating in object θ_i. Then, for any $S > 0$ that denotes the minimum desired reward, an optimal tree T^* is defined to be

$$T^* \in \arg\min_T \mathbb{E}[K(T)] \text{ such that } f(Q(T,\theta),\theta) \geq S,\ \forall \theta \in \Theta.$$

Finding an optimal tree T^* is NP-complete and hence we need to resort to greedy approaches.

Definition 3: ***(Conditional Expected Marginal Gain)*** Given the observed responses $\mathbf{z}_{\mathcal{A}}$ to queries in $Q_{\mathcal{A}}$, the conditional expected marginal gain of choosing a new query $q \notin Q_{\mathcal{A}}$ is given by

$$\Delta(q|\mathbf{z}_{\mathcal{A}}) := \mathbb{E}_\theta[f(Q_{\mathcal{A}} \cup \{q\}, \theta) - f(Q_{\mathcal{A}}, \theta)|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}] \tag{15}$$

where the expectation is taken with respect to Π.

A greedy algorithm to solve the above optimization problem is to construct a decision tree in a top-down manner, where at each internal node, a query that maximizes $\Delta(q|\mathbf{z}_{\mathcal{A}})$, i.e., $\arg\max_{q \notin Q_{\mathcal{A}}} \Delta(q|\mathbf{z}_{\mathcal{A}})$ is chosen, where $Q_{\mathcal{A}}$ denotes the queries leading to that node with $\mathbf{z}_{\mathcal{A}}$ being the responses.

Definition 4: ***(Strong Adaptive Monotonicity)*** A function $f : 2^Q \times \Theta \to \mathbb{R}_{\geq 0}$ is strongly adaptive monotone with respect to Π if, informally "selecting more queries never hurts" with respect to the expected reward. Formally, for all $Q_{\mathcal{A}} \subseteq Q$, all $q \notin Q_{\mathcal{A}}$ and all $z \in \{0, 1\}$ such that $\Pr(Z = z|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}) > 0$, we require

$$\begin{aligned}&\mathbb{E}_\theta[f(Q_{\mathcal{A}},\theta)|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}]\\ &\qquad \leq \mathbb{E}_\theta[f(Q_{\mathcal{A}} \cup \{q\}, \theta)|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}, Z = z].\end{aligned} \tag{16}$$

Definition 5: ***(Adaptive Submodular)*** A function $f : 2^Q \times \Theta \to \mathbb{R}_{\geq 0}$ is adaptive submodular with respect to distribution Π if the conditional expected marginal gain of any fixed query does not increase as more queries are selected and their responses are observed. Formally, f is adaptive submodular w.r.t. Π if for all $Q_{\mathcal{A}}$ and $Q_{\mathcal{B}}$ such that $Q_{\mathcal{A}} \subseteq Q_{\mathcal{B}} \subseteq Q$ and for all $q \notin Q_{\mathcal{B}}$, we have

$$\Delta(q|\mathbf{z}_{\mathcal{B}}) \leq \Delta(q|\mathbf{z}_{\mathcal{A}}). \tag{17}$$

Theorem 8: [13] Suppose $f : 2^Q \times \Theta \to \mathbb{R}_{\geq 0}$ is adaptive submodular and strongly adaptive monotone with respect to Π and there exists an S such that $f(Q, \theta) = S$ for all $\theta \in \Theta$. Let η be any value such that $f(Q_{\mathcal{A}}, \theta) > S - \eta$ implies $f(Q_{\mathcal{A}}, \theta) = S$ for all $Q_{\mathcal{A}} \subseteq Q$ and all θ. Let T^* be an optimal tree with the least expected depth and let $\widehat{T}$ be a suboptimal tree constructed using the greedy algorithm, then

$$\mathbb{E}[K(\widehat{T})] \leq \mathbb{E}[K(T^*)]\left(\ln\left(\frac{S}{\eta}\right) + 1\right). \tag{18}$$

1) Proof of Theorem 5: Let the utility function f be defined as $f(Q_{\mathcal{A}},\theta_i) := 1-\pi_a^2+\left(\pi_a^{k_i}\right)^2$, where π_a is the probability mass of the objects remaining after observing responses to queries in $Q_{\mathcal{A}}$ with θ_i as the unknown object, and k_i denoting the group to which θ_i belongs. As shown in Lemma 2 below, substituting this utility function in(15), we get the conditional expected marginal gain to be $3\pi_{l(a)}\pi_{r(a)}-\sum_{i=1}^m 3\frac{\pi_a^i}{\pi_a}\pi_{l(a)}^i\pi_{r(a)}^i$, which is the greedy criterion for choosing queries at each internal node.

Now, note that $f(Q,\theta)=1,\ \forall\theta\in\Theta$. Also, for any $Q_{\mathcal{A}}\subseteq Q$, if $f(Q_{\mathcal{A}},\theta_i) > 1-3\pi_{\min}^2$, it implies $f(Q_{\mathcal{A}},\theta_i)=1$, hence $\eta=3\pi_{\min}^2$. In addition, it follows from Lemma 2 and Lemma 3 below that the utility function f defined above is adaptive submodular and strongly adaptive monotone. Hence, the result follows from Theorem 8.

Lemma 2: The utility function f defined above is adaptive submodular.

Proof: Consider two subsets of Q such that $Q_{\mathcal{A}}\subseteq Q_{\mathcal{B}}$. Let $\mathbf{z}_{\mathcal{A}},\mathbf{z}_{\mathcal{B}}$ denote the responses to the queries in $Q_{\mathcal{A}}$ and $Q_{\mathcal{B}}$, respectively. Then, we need to show that for any $q\notin Q_{\mathcal{B}}$, $\Delta(q|\mathbf{z}_{\mathcal{A}})\geq\Delta(q|\mathbf{z}_{\mathcal{B}})$.

Let $\Theta_a\subseteq\Theta$ denote the set of objects whose responses to queries in $Q_{\mathcal{A}}$ are same as those in $\mathbf{z}_{\mathcal{A}}$. Then substituting $f(Q_{\mathcal{A}},\theta)=1-\pi_a^2+(\pi_a^i)^2$ in (15), we get

$$\begin{aligned}
&\Delta(q|\mathbf{z}_{\mathcal{A}})\\
&=\sum_{i=1}^m\frac{\pi_{l(a)}^i}{\pi_a}\left[\pi_a^2-\pi_{l(a)}^2-(\pi_a^i)^2+(\pi_{l(a)}^i)^2\right]\\
&\quad+\sum_{i=1}^m\frac{\pi_{r(a)}^i}{\pi_a}\left[\pi_a^2-\pi_{r(a)}^2-(\pi_a^i)^2+(\pi_{r(a)}^i)^2\right]\\
&=\frac{\pi_{l(a)}}{\pi_a}\pi_{r(a)}(\pi_a+\pi_{l(a)})-\sum_{i=1}^m\frac{\pi_{l(a)}^i}{\pi_a}\pi_{r(a)}^i(\pi_a^i+\pi_{l(a)}^i)\\
&\quad+\frac{\pi_{r(a)}}{\pi_a}\pi_{l(a)}(\pi_a+\pi_{r(a)})-\sum_{i=1}^m\frac{\pi_{r(a)}^i}{\pi_a}\pi_{l(a)}^i(\pi_a^i+\pi_{r(a)}^i)\\
&=3\pi_{l(a)}\pi_{r(a)}-\sum_{i=1}^m 3\frac{\pi_a^i}{\pi_a}\pi_{l(a)}^i\pi_{r(a)}^i.
\end{aligned}$$

Similarly, let $\Theta_b\subseteq\Theta$ denote the set of objects whose responses to queries in $Q_{\mathcal{B}}$ are equal to those in $\mathbf{z}_{\mathcal{B}}$. Then, substituting $f(Q_{\mathcal{B}},\theta)=1-\pi_b^2+(\pi_b^i)^2$ in (15), we get $\Delta(q|\mathbf{z}_{\mathcal{B}})=3\pi_{l(b)}\pi_{r(b)}-\sum_{i=1}^m 3\frac{\pi_b^i}{\pi_b}\pi_{l(b)}^i\pi_{r(b)}^i$.

To prove f is adaptive submodular, we need to show that

$$\begin{aligned}
&\pi_{l(a)}\pi_{r(a)}-\sum_{i=1}^m\frac{\pi_a^i}{\pi_a}\pi_{l(a)}^i\pi_{r(a)}^i\\
&\qquad\geq\pi_{l(b)}\pi_{r(b)}-\sum_{i=1}^m\frac{\pi_b^i}{\pi_b}\pi_{l(b)}^i\pi_{r(b)}^i,\\
\Longrightarrow\ &\pi_a\pi_b\pi_{l(a)}\pi_{r(a)}-\sum_{i=1}^m\pi_a^i\pi_b\pi_{l(a)}^i\pi_{r(a)}^i\\
&\qquad\geq\pi_a\pi_b\pi_{l(b)}\pi_{r(b)}-\sum_{i=1}^m\pi_b^i\pi_a\pi_{l(b)}^i\pi_{r(b)}^i.
\end{aligned}$$

Note that since $Q_{\mathcal{A}}\subseteq Q_{\mathcal{B}}$, $\Theta_b\subseteq\Theta_a$ and hence $\pi_b\leq\pi_a$, $\pi_b^i\leq\pi_a^i,\ \forall i\in\{1,\dots,m\}$. For any query $q\notin Q_{\mathcal{B}}$, let $\Theta_{l(a)}$ and $\Theta_{r(a)}$ correspond to the objects in Θ_a that respond 0 and 1 to query q, respectively. Similarly, let $\Theta_{l(b)}$ and $\Theta_{r(b)}$ correspond to the objects in Θ_b that respond 0 and 1 to query q, respectively. Then, $\pi_{l(b)}\leq\pi_{l(a)}$, $\pi_{l(b)}^i\leq\pi_{l(a)}^i$, $\forall i$, and $\pi_{r(b)}\leq\pi_{r(a)}$, $\pi_{r(b)}^i\leq\pi_{r(a)}^i$, $\forall i$. Hence

$$\begin{aligned}
&\pi_a\pi_b\pi_{l(a)}\pi_{r(a)}-\sum_{i=1}^m\pi_a^i\pi_b\pi_{l(a)}^i\pi_{r(a)}^i\\
&=\pi_a\pi_b\sum_{i=1}^m\pi_{l(a)}^i\pi_{r(a)}^i+\pi_a\pi_b\sum_{i\neq j}\pi_{l(a)}^i\pi_{r(a)}^j\\
&\quad-\sum_{i=1}^m\pi_a^i\pi_b\pi_{l(a)}^i\pi_{r(a)}^i\\
&=\sum_{i=1}^m\pi_{l(a)}^i\pi_{r(a)}^i(\pi_a-\pi_a^i)\pi_b+\pi_a\pi_b\sum_{i\neq j}\pi_{l(a)}^i\pi_{r(a)}^j &\text{(19a)}\\
&\geq\sum_{i=1}^m\pi_{l(a)}^i\pi_{r(a)}^i(\pi_a-\pi_a^i)\pi_b+\pi_a\pi_b\sum_{i\neq j}\pi_{l(b)}^i\pi_{r(b)}^j &\text{(19b)}\\
&=\sum_{i=1}^m\pi_{l(a)}^i\pi_{r(a)}^i(\pi_a-\pi_a^i)(\pi_b-\pi_b^i)\\
&\quad+\sum_{i=1}^m\pi_{l(a)}^i\pi_{r(a)}^i(\pi_a-\pi_a^i)\pi_b^i+\pi_a\pi_b\sum_{i\neq j}\pi_{l(b)}^i\pi_{r(b)}^j &\text{(19c)}\\
&\geq\sum_{i=1}^m\pi_{l(b)}^i\pi_{r(b)}^i(\pi_a-\pi_a^i)(\pi_b-\pi_b^i)\\
&\quad+\sum_{i=1}^m\pi_{l(a)}^i\pi_{r(a)}^i(\pi_a-\pi_a^i)\pi_b^i+\pi_a\pi_b\sum_{i\neq j}\pi_{l(b)}^i\pi_{r(b)}^j &\text{(19d)}\\
&\geq\sum_{i=1}^m\pi_{l(b)}^i\pi_{r(b)}^i(\pi_a-\pi_a^i)(\pi_b-\pi_b^i)\\
&\quad+\sum_{i=1}^m\pi_{l(b)}^i\pi_{r(b)}^i(\pi_b-\pi_b^i)\pi_a^i+\pi_a\pi_b\sum_{i\neq j}\pi_{l(b)}^i\pi_{r(b)}^j &\text{(19e)}\\
&=\sum_{i=1}^m\pi_{l(b)}^i\pi_{r(b)}^i\pi_a(\pi_b-\pi_b^i)+\pi_a\pi_b\sum_{i\neq j}\pi_{l(b)}^i\pi_{r(b)}^j\\
&=\pi_a\pi_b\pi_{l(b)}\pi_{r(b)}-\sum_{i=1}^m\pi_a\pi_b^i\pi_{l(b)}^i\pi_{r(b)}^i
\end{aligned}$$

where (19e) follows from (19d) since

$$\begin{aligned}
&\sum_{i=1}^m\pi_{l(a)}^i\pi_{r(a)}^i(\pi_a-\pi_a^i)\pi_b^i\\
&=\sum_{i=1}^m\pi_{l(a)}^i\pi_{l(b)}^i\pi_{r(a)}^i(\pi_a-\pi_a^i)\\
&\qquad+\sum_{i=1}^m\pi_{r(a)}^i\pi_{r(b)}^i\pi_{l(a)}^i(\pi_a-\pi_a^i)\\
&\geq\sum_{i=1}^m\pi_{l(a)}^i\pi_{l(b)}^i\pi_{r(b)}^i(\pi_b-\pi_b^i)\\
&\qquad+\sum_{i=1}^m\pi_{r(a)}^i\pi_{r(b)}^i\pi_{l(b)}^i(\pi_b-\pi_b^i)\\
&=\sum_{i=1}^m\pi_{l(b)}^i\pi_{r(b)}^i(\pi_b-\pi_b^i)\pi_a^i
\end{aligned}$$

thus proving that f is adaptive submodular. ∎

Lemma 3: The utility function f as defined above is strongly adaptive monotone.

Proof: Consider any subset of queries $Q_{\mathcal{A}} \subseteq Q$, and let $\mathbf{z}_{\mathcal{A}}$ denote the responses to these queries. Let Θ_a denote the set of objects whose responses to queries in $Q_{\mathcal{A}}$ are equal to those of $\mathbf{z}_{\mathcal{A}}$. For any query $q \notin Q_{\mathcal{A}}$, let $\Theta_{l(a)}$ and $\Theta_{r(a)}$ correspond to the objects in Θ_a that respond 0 and 1 to query q, respectively.

For strong adaptive monotonicity, we need to show that

$$1-\pi_a^2+\sum_{i=1}^{m}\frac{\left(\pi_a^i\right)^3}{\pi_a} \leq 1-\pi_{l(a)}^2+\sum_{i=1}^{m}\frac{\left(\pi_{l(a)}^i\right)^3}{\pi_{l(a)}}, \quad \text{if } \pi_{l(a)}>0$$

$$\text{and } 1-\pi_a^2+\sum_{i=1}^{m}\frac{\left(\pi_a^i\right)^3}{\pi_a} \leq 1-\pi_{r(a)}^2+\sum_{i=1}^{m}\frac{\left(\pi_{r(a)}^i\right)^3}{\pi_{r(a)}}, \quad \text{if } \pi_{r(a)}>0.$$

We will show the first inequality, and the second inequality can be shown in a similar manner. Given $\pi_{l(a)} > 0$, we need to show that

$$\pi_a^3\pi_{l(a)}-\pi_{l(a)}^3\pi_a \geq \sum_{i=1}^{m}\left(\pi_a^i\right)^3\pi_{l(a)}-\left(\pi_{l(a)}^i\right)^3\pi_a.$$

Note that

$$\begin{aligned}
&\pi_a^3\pi_{l(a)}-\pi_{l(a)}^3\pi_a \\
&=\left(\pi_{l(a)}+\pi_{r(a)}\right)^3\pi_{l(a)}-\pi_{l(a)}^3\left(\pi_{l(a)}+\pi_{r(a)}\right) \\
&=\pi_{r(a)}^3\pi_{l(a)}+3\pi_{l(a)}^2\pi_{r(a)}^2+2\pi_{l(a)}^3\pi_{r(a)} && \text{(20a)}\\
&\geq \sum_{i=1}^{m}\left[\left(\pi_{r(a)}^i\right)^3\pi_{l(a)}+3\pi_{l(a)}\pi_{l(a)}^i\left(\pi_{r(a)}^i\right)^2\right] \\
&\quad +2\pi_{l(a)}^3\pi_{r(a)} && \text{(20b)}\\
&=\sum_{i=1}^{m}\left[\left(\pi_{r(a)}^i\right)^3\pi_{l(a)}+3\pi_{l(a)}\pi_{l(a)}^i\left(\pi_{r(a)}^i\right)^2\right. \\
&\quad \left.-\pi_{r(a)}\left(\pi_{l(a)}^i\right)^3\right]+2\pi_{l(a)}^3\pi_{r(a)} \\
&\quad +\sum_{i=1}^{m}\pi_{r(a)}\left(\pi_{l(a)}^i\right)^3 && \text{(20c)}\\
&\geq \sum_{i=1}^{m}\left[\left(\pi_{r(a)}^i\right)^3\pi_{l(a)}+3\pi_{l(a)}\pi_{l(a)}^i\left(\pi_{r(a)}^i\right)^2\right. \\
&\quad \left.-\left(\pi_{l(a)}^i\right)^3\pi_{r(a)}+3\left(\pi_{l(a)}^i\right)^2\pi_{r(a)}^i\pi_{l(a)}\right] && \text{(20d)}\\
&=\sum_{i=1}^{m}\left\{\pi_{l(a)}\left[\left(\pi_{l(a)}^i\right)^3+3\left(\pi_{l(a)}^i\right)^2\pi_{r(a)}^i\right.\right. \\
&\quad \left.+3\pi_{l(a)}^i\left(\pi_{r(a)}^i\right)^2+\left(\pi_{r(a)}^i\right)^3\right] \\
&\quad \left.-\left(\pi_{l(a)}^i\right)^3\pi_{l(a)}-\left(\pi_{l(a)}^i\right)^3\pi_{r(a)}\right\} \\
&=\sum_{i=1}^{m}\left(\pi_a^i\right)^3\pi_{l(a)}-\left(\pi_{l(a)}^i\right)^3\pi_a && \text{(20e)}
\end{aligned}$$

where (20b) follows from (20a) as $\pi_{r(a)}^3\pi_{l(a)}$ and $3\pi_{l(a)}\pi_{l(a)}\pi_{r(a)}^2$ has more nonnegative terms than $\sum_{i=1}^{m}\left(\pi_{r(a)}^i\right)^3\pi_{l(a)}$, $\sum_{i=1}^{m}3\pi_{l(a)}\pi_{l(a)}^i\left(\pi_{r(a)}^i\right)^2$, respectively. Also (20d) follows from (20c) since

$$\begin{aligned}
&\pi_{r(a)}\left[2\pi_{l(a)}^3+\sum_{i=1}^{m}\left(\pi_{l(a)}^i\right)^3\right] \\
&=\pi_{r(a)}\left[\sum_{i=1}^{m}3\left(\pi_{l(a)}^i\right)^3+6\sum_{i\neq j}\left(\pi_{l(a)}^i\right)^2\pi_{l(a)}^j\right. \\
&\quad \left.+6\sum_{i\neq j\neq k}\pi_{l(a)}^i\pi_{l(a)}^j\pi_{l(a)}^k\right] \\
&=\left(\sum_{h=1}^{m}\pi_{r(a)}^h\right)\left[\sum_{i=1}^{m}3\left(\pi_{l(a)}^i\right)^3+6\sum_{i\neq j}\left(\pi_{l(a)}^i\right)^2\pi_{l(a)}^j\right. \\
&\quad \left.+6\sum_{i\neq j\neq k}\pi_{l(a)}^i\pi_{l(a)}^j\pi_{l(a)}^k\right] \\
&\geq 3\sum_{i=1}^{m}\left(\pi_{l(a)}^i\right)^3\pi_{r(a)}^i+3\sum_{i\neq j}\left(\pi_{l(a)}^i\right)^2\pi_{r(a)}^i\pi_{l(a)}^j \\
&=3\pi_{l(a)}\sum_{i=1}^{m}\left(\pi_{l(a)}^i\right)^2\pi_{r(a)}^i
\end{aligned}$$

thus proving that f is strongly adaptive monotone. ■

Appendix D
Proof of Theorem 7

Let T_a denote a subtree from any node 'a' in the tree T and let $\mathcal{L}_a$ denote the set of leaf nodes in this subtree. Then, let μ_a denote the expected number of queries required to identify the group of an object terminating in a leaf node of this subtree, given by

$$\mu_a = \sum_{j\in\mathcal{L}_a}\frac{\pi_{\Theta_j}}{\pi_{\Theta_a}}\tilde{p}_j^a d_j^a$$

where $d_j^a, \tilde{p}_j^a$ denotes the depth of leaf node j in the subtree T_a and the probability of reaching that leaf node given $\theta \in \Theta_j$, respectively, and let H_a denote the entropy of the probability distribution of the object groups at the root node of this subtree, i.e.,

$$H_a = -\sum_{i=1}^{m}\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}\log\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}.$$

Now, we show using induction that for any subtree T_a in the tree T, the following relation holds:

$$\pi_{\Theta_a}\mu_a-\pi_{\Theta_a}H_a=\sum_{s\in\mathcal{I}_a}\tilde{p}_s^a\pi_{\Theta_s}\left\{1-\sum_{q\in Q^{z_s}}p_{z_s}(q)\left[H(\rho_s(q))-\sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i(q))\right]\right\}$$

where $\mathcal{I}_a$ denotes the set of internal nodes in the subtree T_a.

The relation holds trivially for any subtree rooted at a leaf node of the tree T with both the left hand side and the right hand side of the expression being equal to 0. Now, assume the above relation holds for all subtrees rooted at the child nodes

of node 'a.' Note that node 'a' has a set of left and right child nodes, each set corresponding to one query from the query group selected at that node. Then, using the decomposition in Lemma 1 on each query from this query group, we have

$$\begin{aligned}
&1\cdot\pi_{\Theta_a}[\mu_a - H_a]\\
&= \sum_{q\in Q^{z_a}} p_{z_a}(q)\pi_{\Theta_a}[\mu_a - H_a]\\
&= \sum_{q\in Q^{z_a}} p_{z_a}(q)\Bigg\{\pi_{\Theta_{l^q(a)}}\left[\mu_{l^q(a)} - H_{l^q(a)}\right]\\
&\quad + \pi_{\Theta_{r^q(a)}}\left[\mu_{r^q(a)} - H_{r^q(a)}\right]\\
&\quad + \pi_{\Theta_a}\left[1 - H(\rho_a(q)) - \sum_{i=1}^{m}\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}H(\rho_a^i(q))\right]\Bigg\}\\
&= \sum_{q\in Q^{z_a}} p_{z_a}(q)\Bigg\{\pi_{\Theta_{l^q(a)}}\left[\mu_{l^q(a)} - H_{l^q(a)}\right]\\
&\quad + \pi_{\Theta_{r^q(a)}}\left[\mu_{r^q(a)} - H_{r^q(a)}\right]\Bigg\}\\
&\quad + \pi_{\Theta_a}\Bigg\{1 - \sum_{q\in Q^{z_a}} p_{z_a}(q)\Bigg[H(\rho_a(q))\\
&\qquad - \sum_{i=1}^{m}\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}H(\rho_a^i(q))\Bigg]\Bigg\}
\end{aligned}$$

where $l^q(a), r^q(a)$ correspond to the left and right child of node 'a' when query q is chosen from the query group and $\mu_{l^q(a)}, \pi_{\Theta_{l^q(a)}}, H_{l^q(a)}$ correspond to the expected depth of a leaf node in the subtree $T_{l^q(a)}$, probability mass of the objects at the root node of this subtree, and the entropy of the probability distribution of the objects at the root node of this subtree, respectively. Now, using the induction hypothesis, we get

$$\begin{aligned}
&\pi_{\Theta_a}\mu_a - \pi_{\Theta_a}H_a\\
&= \sum_{q\in Q^{z_a}} p_{z_a}(q)\Bigg\{\sum_{s\in\mathcal{I}_{l^q(a)}}\tilde{p}_s^{l^q(a)}\pi_{\Theta_s}\Bigg[1\\
&\qquad - \sum_{q\in Q^{z_s}} p_{z_s}(q)\left(H(\rho_s(q)) - \sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i(q))\right)\Bigg]\Bigg\}\\
&\quad + \sum_{q\in Q^{z_a}} p_{z_a}(q)\Bigg\{\sum_{s\in\mathcal{I}_{r^q(a)}}\tilde{p}_s^{r^q(a)}\pi_{\Theta_s}\Bigg[1\\
&\qquad - \sum_{q\in Q^{z_s}} p_{z_s}(q)\left(H(\rho_s(q)) - \sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i(q))\right)\Bigg]\Bigg\}\\
&\quad + \pi_{\Theta_a}\Bigg\{1 - \sum_{q\in Q^{z_a}} p_{z_a}(q)\Bigg[H(\rho_a(q))\\
&\qquad - \sum_{i=1}^{m}\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}H(\rho_a^i(q))\Bigg]\Bigg\}\\
&= \sum_{s\in\mathcal{I}_a}\tilde{p}_s^a\pi_{\Theta_s}\Bigg\{1 - \sum_{q\in Q^{z_s}} p_{z_s}(q)\Bigg[H(\rho_s(q))\\
&\qquad - \sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i(q))\Bigg]\Bigg\}
\end{aligned}$$

thereby completing the induction. Finally, the result follows by applying the relation to the subtree rooted at the root node of T, whose probability mass $\pi_{\Theta_a} = 1$.

Appendix E
Reduction Factor Calculation in the Persistent Noise Model

At any internal node $a \in \mathcal{I}$ in a tree, let δ_i^a denote the Hamming distance between the query responses up to this internal node (Q_a) and the true responses of object θ_i to those queries. Also, let n_a denote the number of queries from the set of $N\nu$ queries (that were prone to error) in the set $Q \setminus Q_a$ and for a query $q \in Q \setminus Q_a$, denote by $b_i(q)$ the binary response of object θ_i to that query. Denote by the set $I^a = \{i : \delta_i^a \le \epsilon'\}$, the object groups with nonzero number of objects at this internal node. All the formulas below come from routine calculations based on probability model 2.

For a query $q \in Q\setminus Q_a$, that is not prone to error, the reduction factor and the group reduction factors generated by choosing that query at node 'a' are as follows. The group reduction factor of any group $i \in I^a$ is equal to 1 and the reduction factor is given by

$$\begin{aligned}
\rho_a &= \frac{\max\{A,B\}}{\sum_{i\in I_0^a\bigcap I_1^a}\pi_i\left[\sum_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}\right]}\\
A &= \sum_{i\in I_0^a}\pi_i\left[\sum_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}\right]\\
B &= \sum_{i\in I_1^a}\pi_i\left[\sum_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}\right]
\end{aligned}$$

where $I_0^a = \{i \in I^a : b_i(q) = 0\}$, $I_1^a = \{i \in I^a : b_i(q) = 1\}$, and $\tau_i^a = \min(n_a, \epsilon' - \delta_i^a)$.

In addition, for a query $q \in Q \setminus Q_a$ that is prone to error, denote by $\delta_i^{l(a)}, \delta_i^{r(a)}$ the Hamming distance between the user responses to queries up to the left and right child node of node 'a' with query q chosen at node 'a,' and the true responses of object θ_i to those queries. In particular, $\delta_i^{l(a)} = \delta_i^a + |b_i(q) - 0|$ and $\delta_i^{r(a)} = \delta_i^a + |b_i(q) - 1|$. Then, the reduction factor and the group reduction factors generated by choosing this query at node 'a' are as follows. The group reduction factor of a group $i \in I^a$ whose $\delta_i^a = \epsilon'$ is equal to 1 and that of a group whose $\delta_i^a < \epsilon'$ is given by

$$\begin{aligned}
\rho_a^i &= \frac{\max\{A,B\}}{\sum_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}}\\
A &= \sum_{e=0}^{\tau_i^{l(a)}}\binom{n_a-1}{e}p^{e+\delta_i^{l(a)}}(1-p)^{N\nu-e-\delta_i^{l(a)}}\\
B &= \sum_{e=0}^{\tau_i^{r(a)}}\binom{n_a-1}{e}p^{e+\delta_i^{r(a)}}(1-p)^{N\nu-e-\delta_i^{r(a)}}
\end{aligned}$$

where $\tau_i^{l(a)} = \min(n_a - 1, \epsilon' - \delta_i^{l(a)})$ and $\tau_i^{r(a)} = \min(n_a - 1, \epsilon' - \delta_i^{r(a)})$, and the reduction factor is given by

$$\rho_a = \frac{\max\{A, B\}}{\sum_{i \in I^a} \pi_i \left[\sum_{e=0}^{\tau_i^a} \binom{n_a}{e} p^{e+\delta_i^{r(a)}} (1-p)^{N\nu - e - \delta_i^{r(a)}} \right]}$$

$$A = \sum_{i \in I^{l(a)}} \pi_i \left[\sum_{e=0}^{\tau_i^{l(a)}} \binom{n_a - 1}{e} p^{e+\delta_i^{l(a)}} (1-p)^{N\nu - e - \delta_i^{l(a)}} \right]$$

$$B = \sum_{i \in I^{r(a)}} \pi_i \left[\sum_{e=0}^{\tau_i^{r(a)}} \binom{n_a - 1}{e} p^{e+\delta_i^{r(a)}} (1-p)^{N\nu - e - \delta_i^{r(a)}} \right].$$

ACKNOWLEDGMENT

The authors would like to thank R. Nowak for helpful feedback and S. S. Pradhan for insightful discussions during the initial phase of this research. Also, the authors thank A. Ganesan, R. Richardson, P. Saxman, G. Vallabha, and C. Weber for their contributions. Finally, the authors would like to thank all the anonymous reviewers and the Associate Editor whose suggestions greatly helped improve the overall quality of the paper.

REFERENCES

[1] D. Angluin, "Queries revisited," *Theoret. Comput. Sci.*, vol. 313, pp. 175–194, 2004.
[2] S. Dasgupta, "Analysis of a greedy active learning strategy," *Adv. Neural Inf. Process. Syst.*, 2004.
[3] M. Garey, "Optimal binary decision trees for diagnostic identification problems," Ph.D., Univ. Wisconsin, Madison, 1970.
[4] M. Garey, "Optimal binary identification procedures," *SIAM J. Appl. Math.*, vol. 23(2, pp. 173–186, 1972.
[5] D. W. Loveland, "Performance bounds for binary testing with arbitrary weights," *Acta Informatica*, 1985.
[6] I. Koren and Z. Kohavi, "Diagnosis of intermittent faults in combinational networks," *IEEE Trans. Comput.*, vol. C-26, pp. 1154–1158, 1977.
[7] Ünlüyurt, "Sequential testing of complex systems: A review," *Discr. Appl. Math.*, vol. 142, no. 1-3, pp. 189–205, 2004.
[8] J. Shiozaki, H. Matsuyama, E. O'Shima, and M. Iri, "An improved algorithm for diagnosis of system failures in the chemical process," *Computat. Chem. Eng.*, vol. 9, no. 3, pp. 285–293, 1985.
[9] F. Yu, F. Tu, H. Tu, and K. Pattipati, "Multiple disease (fault) diagnosis with applications to the QMR-DT problem," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2003, vol. 2, pp. 1187–1192.
[10] D. Geman and B. Jedynak, "An active testing model for tracking roads in satellite images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 1–14, 1996.
[11] M. J. Swain and M. A. Stricker, "Promising directions in active vision," *Int. J. Comput. Vision*, vol. 11, no. 2, pp. 109–126, 1993.
[12] R. Nowak, "Generalized binary search," in *Proc. Allerton Conf.*, 2008.
[13] D. Golovin and A. Krause, "Adaptive Submodularity: A new approach to active learning and stochastic optimization," in *Proc. Int. Conf. Learn. Theory (COLT)*, 2010.
[14] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi, "Approximation algorithms for optimal decision trees and adaptive TSP problems," in *Proc. ICALP, LNCS*, 2010.
[15] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul. 1948.
[16] R. M. Fano, *Transmission of Information*. Cambridge, MA: MIT Press, 1961.
[17] D. A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proc. Inst. Radio Eng.*, 1952.
[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[19] L. Hyafil and R. Rivest, "Constructing optimal binary decision trees is np-complete," *Inf. Process. Lett.*, vol. 5, no. 1, pp. 15–17, 1976.
[20] S. Roy, H. Wang, G. Das, U. Nambiar, and M. Mohania, "Minimum-effort driven dynamic faceted search in structured databases," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 13–22.
[21] S. R. Kosaraju, T. M. Przytycka, and R. S. Borgstrom, "On an optimal split tree problem," in *Proc. 6th Int. Workshop on Algorithms and Data Structures (WADS)*, 1999, pp. 11–14.
[22] R. M. Goodman and P. Smyth, "Decision tree design from a communication theory standpoint," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, 1988.
[23] V. T. Chakaravarthy, V. Pandit, S. Roy, P. Awasthi, and M. Mohania, "Decision trees for entity identification: Approximation algorithms and hardness results," in *Proc. ACM SIGMOD Symp. Principles of Database Syst.*, 2007.
[24] V. T. Chakaravarthy, V. Pandit, S. Roy, and Y. Sabharwal, "Approximating decision trees with multiway branches," in *Proc. ICALP*, 2009, pp. 210–221.
[25] F. Cicalese, T. Jacobs, E. Laber, and M. Molinaro, "On greedy algorithms for decision trees," in *Proc. ISAAC*, 2010.
[26] M. Adler and B. Heeringa, "Approximating optimal binary decision trees," in *Proc. 11th Int. Workshop on Approx., Random. Combinator. Optimiz.*, 2008, pp. 1–9.
[27] G. Bellala, S. K. Bhavnani, and C. Scott, "Extensions of generalized binary search to group identification and exponential costs," *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 23, 2010.
[28] D. Golovin, D. Ray, and A. Krause, "Near-optimal Bayesian active learning with noisy observations," *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 23, 2010.
[29] M. Kääriäinen, "Active learning in the non-realizable case," *Algorithm. Learn. Theory*, pp. 63–77, 2006.
[30] R. Nowak, "Noisy generalized binary search," *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 21, 2009.
[31] A. Rényi, "On a problem of information theory," *MTA Mat. Kut. Int. Kozl.*, vol. 6B, pp. 505–516, 1961.
[32] S. A. Goldman, M. J. Kearns, and R. E. Schapire, "Exact identification of circuits using fixed points of amplification functions," in *Proc. 31st Ann. Symp. Found. Comput. Sci.*, 1990.
[33] J. Jackson, E. Shamir, and C. Shwartzman, "Learning with queries corrupted by classification noise," in *Proc. 5th Israel Symp. Theory of Comput. Syst.*, 1997, pp. 45–53.
[34] S. Hanneke, "Teaching dimension and the complexity of active learning," in *Proc. 20th Conf. Learn. Theory*, 2007.
[35] D. Angluin and D. K. Slonim, "Randomly fallible teachers: Learning monotone DNF with an incomplete membership oracle," *Mach. Learn.*, vol. 14, pp. 7–26, 1994.
[36] M. F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006.
[37] J. Czyzowicz, D. Mundici, and A. Pelc, "Ulam's searching games with lies," *J. Combinator. Theory*, vol. 52, pp. 62–76, 1989.
[38] A. Pelc, "Detecting errors in searching games," *J. Combinator. Theory*, vol. 51, pp. 43–54, 1989.
[39] A. Pelc, "Searching with known error probability," *Theoret. Comput. Sci.*, vol. 63, pp. 185–202, 1989.
[40] J. Spencer, "Ulam's searching game with a fixed number of lies," *Theoret. Comput. Sci.*, vol. 95, pp. 307–321, 1992.
[41] A. Pelc, "Searching games with errors – Fifty years of coping with liars," *Theoret. Comput. Sci.*, vol. 270, pp. 71–109, 2002.
[42] S. Bhavnani, A. Abraham, C. Demeniuk, M. Gebrekristos, A. Gong, S. Nainwal, G. Vallabha, and R. Richardson, "Network analysis of toxic chemicals and symptoms: Implications for designing first-responder systems," in *Proc. Amer. Med. Informat. Assoc.*, 2007.
[43] J. R. Quinlan, *C4.5: Programs for Machine Learning*. New York: Morgan Kaufmann , 1993.
[44] M. Kearns and Y. Mansour, "On the boosting ability of top-down decision tree learning algorithms," in *Proc. 28th Ann. ACM Symp. Theory of Comput.*, 1995.
[45] E. Takimoto and A. Maruoka, "Top-down decision tree learning as information based boosting," *Theoret. Comput. Sci.*, 2003.
[46] S. Dasgupta, "Coarse sample complexity bounds for active learning," *Adv. Neural Inf. Process. Syst.*, 2006.
[47] A. P. Korostelev and J. C. Kim, "Rates of convergence of the sup-norm risk in image models under sequential designs," *Statist. Probabil. Lett.*, vol. 46, pp. 391–399, 2000.

Gowtham Bellala received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, in 2006, and the M.S. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 2008.

He is currently working toward the Ph.D. degree in electrical engineering and computer science. His research interests include pattern recognition, machine learning, data mining, statistical signal processing, and applications.

Suresh K. Bhavnani received the Ph.D. degree in computational design and human-computer interaction from Carnegie Mellon University, Pittsburgh, PA, in 1998.

He is Associate Professor of biomedical informatics in the Institute for Translational Sciences (ITS), University of Texas (UT) Medical Branch, and holds a secondary appointment with the Department of Preventive Medicine and Community Health, and an Adjunct Associate Professor appointment with the School of Biomedical Informatics, UT Houston. His research interests include network visualization and analysis of biomedical data, with translation to the design of decision-support systems.

Dr. Bhavnani has received two distinguished paper awards in translational bioinformatics, and a distinguished paper award in medical informatics from the American Medical Informatics Association. In addition, he has received an Outstanding Research Mentorship Award from the University of Michigan, and the Rising STAR award from the University of Texas Systems. He is PI of the new Discovery and Innovation through Visual Analytics (DIVA) Lab at UTMB, and PI of a grant from the Center for Disease Control and Prevention.

Clayton Scott received the A.B. degree in mathematics from Harvard University, Cambridge, in 1998, and the M.S. and Ph.D. degrees in electrical engineering from Rice University, Houston, TX, in 2000 and 2004, respectively.

He was a Postdoctoral Fellow with the Department of Statistics at Rice University, and is currently an Assistant Professor with the Departments of Electrical Engineering and Computer Science and of Statistics, University of Michigan, Ann Arbor. His research interests include pattern recognition, machine learning, statistical signal processing, and applications.