Extensions of Generalized Binary Search to Group Identification and Exponential Costs

Gowtham Bellala[1], **Suresh K. Bhavnani**[2,3,4], **Clayton Scott**[1]
[1]Department of EECS, University of Michigan, Ann Arbor, MI 48109
[2]Institute for Translational Sciences, [3]Dept. of Preventative Medicine and Community Health, University of Texas Medical Branch, Galveston, TX 77555
[4]School of Biomedical Informatics, University of Texas, Houston, TX 77030
gowtham@umich.edu, skbhavnani@gmail.com, clayscot@umich.edu

Abstract

Generalized Binary Search (GBS) is a well known greedy algorithm for identifying an unknown object while minimizing the number of "yes" or "no" questions posed about that object, and arises in problems such as active learning and active diagnosis. Here, we provide a coding-theoretic interpretation for GBS and show that GBS can be viewed as a top-down algorithm that greedily minimizes the expected number of queries required to identify an object. This interpretation is then used to extend GBS in two ways. First, we consider the case where the objects are partitioned into groups, and the objective is to identify only the group to which the object belongs. Then, we consider the case where the cost of identifying an object grows exponentially in the number of queries. In each case, we present an exact formula for the objective function involving Shannon or Rényi entropy, and develop a greedy algorithm for minimizing it.

1 Introduction

In applications such as active learning [1, 2, 3, 4], disease/fault diagnosis [5, 6, 7], toxic chemical identification [8], computer vision [9, 10] or the adaptive traveling salesman problem [11], one often encounters the problem of identifying an unknown object while minimizing the number of binary questions posed about that object. In these problems, there is a set $\Theta = \{\theta_1, \cdots, \theta_M\}$ of M different objects and a set $Q = \{q_1, \cdots, q_N\}$ 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 $\Pi = (\pi_1, \cdots, \pi_M)$, i.e., $\pi_i = \Pr(\theta = \theta_i)$, and the goal is to uniquely identify this unknown object 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. For example, in active learning, the objects are classifiers and the queries are the labels for fixed test points. In active diagnosis, objects may correspond to faults, and queries to alarms. This problem has been generically referred to as binary testing or object/entity identification in the literature [5, 12]. We will refer to this problem as object identification. Our attention is restricted to the case where Θ and Q are finite, and the queries are noiseless.

The goal in object identification is to construct an optimal binary decision tree, where each internal node in the tree is associated with a query from Q, and each leaf node corresponds to an object from Θ. Optimality is often with respect to the expected depth of the leaf node corresponding to the unknown object θ. In general the determination of an optimal tree is NP-complete [13]. Hence, various greedy algorithms [5, 14] have been proposed to obtain a suboptimal binary decision tree. A well studied algorithm for this problem is known as the *splitting algorithm* [5] or *generalized binary search* (GBS) [1, 2]. This is the greedy algorithm which selects a query that most evenly divides the probability mass of the remaining objects [1, 2, 5, 15].

GBS assumes that the end goal is to rapidly identify individual objects. However, in applications such as disease diagnosis, where Θ is a collection of possible diseases, it may only be necessary to identify the intervention or response to an object, rather than the object itself. In these problems, the object set Θ is partitioned into groups and it is only necessary to identify the group to which the unknown object belongs. We note below that GBS is not necessarily efficient for group identification.

To address this problem, we first present a new interpretation of GBS from a coding-theoretic perspective by viewing the problem of object identification as constrained source coding. Specifically, we present an exact formula for the expected number of queries required to identify an unknown object in terms of Shannon entropy of the prior distribution Π, and show that GBS is a top-down algorithm that greedily minimizes this cost function. Then, we extend this framework to the problem of group identification and derive a natural extension of GBS for this problem.

We also extend the coding theoretic framework to the problem of object (or group) identification where the cost of identifying an object grows exponentially in the number of queries, i.e., the cost of identifying an object using d queries is λ^d for some fixed $\lambda > 1$. Applications where such a scenario arises have been discussed earlier in the context of source coding [16], random search trees [17] and design of alphabetic codes [18], for which efficient optimal or greedy algorithms have been presented. In the context of object/group identification, the exponential cost function has certain advantages in terms of avoiding deep trees (which is crucial in time-critical applications) and being more robust to misspecification of the prior probabilities. However, there does not exist an algorithm to the best of our knowledge that constructs a good suboptimal decision tree for the problem of object/group identification with exponential costs. Once again, we show below that GBS is not necessarily efficient for minimizing the exponential cost function, and propose an improved greedy algorithm that generalizes GBS.

1.1 Notation

We denote an object identification problem by a pair $(\mathbf{B}, \Pi)$ where $\mathbf{B}$ is a known $M \times N$ binary matrix with b_{ij} equal to 1 if $\theta_i \in q_j$, and 0 otherwise. 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 Θ. For a decision tree with L leaves, the leaf nodes are indexed by the set $\mathcal{L} = \{1, \cdots, L\}$ and the internal nodes are indexed by the set $\mathcal{I} = \{L+1, \cdots, 2L-1\}$. At any node '$a$', let $Q_a \subseteq Q$ denote the set of queries that have been performed along the path from the root node up to that node. An object θ_i reaches node 'a' if it agrees with the true θ on all queries in Q_a, i.e., the binary values in $\mathbf{B}$ for the rows corresponding to θ_i and θ are same over the columns corresponding to queries in Q_a. At any internal node $a \in \mathcal{I}$, let $l(a), r(a)$ denote the "left" and "right" child nodes, and let $\Theta_a \subseteq \Theta$ denote the set of objects that reach node 'a'. Thus, the sets $\Theta_{l(a)} \subseteq \Theta_a, \Theta_{r(a)} \subseteq \Theta_a$ correspond to the objects in Θ_a 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 reaching node 'a' in the tree. Finally, we denote the Shannon entropy of a proportion $\pi \in [0, 1]$ by $H(\pi) := -\pi \log_2 \pi - (1-\pi)\log_2(1-\pi)$ and that of a vector $\Pi = (\pi_1, \cdots, \pi_M)$ by $H(\Pi) := -\sum_i \pi_i \log_2 \pi_i$, where we use the limit, $\lim_{\pi \to 0} \pi \log_2 \pi = 0$, to define the value of $0 \log_2 0$.

2 GBS Greedily Minimizes the Expected Number of Queries

We begin by noting that object identification reduces to the standard source coding problem [19] in the special case when Q is *complete*, meaning, for any $S \subseteq \Theta$ there exists a query $q \in Q$ such that either $q = S$ or $\Theta \setminus q = S$. Here, the problem of constructing an optimal binary decision tree is equivalent to constructing optimal variable length binary prefix codes, for which there exists an efficient optimal algorithm known as the Huffman algorithm [20]. It is also known that the expected length of any binary prefix code (i.e., expected depth of any binary decision tree) is bounded below by the Shannon entropy of the prior distribution Π [19].

For the problem of object identification, where Q is not complete, the entropy lower bound is still valid, but Huffman coding cannot be implemented. In this case, GBS is a greedy, top-down algorithm that is analogous to Shannon-Fano coding [21, 22]. We now show that GBS is actually greedily minimizing the expected number of queries required to identify an object.

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 object.

Definition 1 (Reduction factor)**.** *Let T be a decision tree constructed on the pair $(\mathbf{B}, \Pi)$. The* reduction factor *at any internal node 'a' in the tree is defined by $\rho_a = \max\{\pi_{\Theta_{l(a)}}, \pi_{\Theta_{r(a)}}\}/\pi_{\Theta_a}$.*

Note that $0.5 \leq \rho_a \leq 1$. 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. We assume that the rows of $\mathbf{B}$ are distinct so that $\mathcal{T}(\mathbf{B}, \Pi) \neq \emptyset$. 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 number of queries required to identify object θ_i in the tree. Then the expected number of queries required to identify an unknown object using a tree (or, the expected depth of a tree) is $L_1(\Pi) = \sum_i \pi_i d_i$. Note that the cost function depends on both Π and $\mathbf{d} = (d_1, \cdots, d_M)$. However, we do not show the dependence on $\mathbf{d}$ explicitly.

Theorem 1. *For any $T \in \mathcal{T}(\mathbf{B}, \Pi)$, the expected number of queries required to identify an unknown object is given by*

$$L_1(\Pi) = H(\Pi) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a}[1 - H(\rho_a)]. \tag{1}$$

Theorems 1, 2 and 3 are special cases of Theorem 4, whose proof is sketched in the Appendix. Complete proofs are given in the Supplemental Material. Since $H(\rho_a) \leq 1$, this theorem recovers the result that $L_1(\Pi)$ is bounded below by the Shannon entropy $H(\Pi)$. It presents the exact formula for the gap in this lower bound. It also follows from the above result that a tree attains the entropy bound iff the reduction factors are equal to 0.5 at each internal node in the tree. Using this result, minimizing $L_1(\Pi)$ can be formulated as the following optimization problem:

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

Since Π is fixed, this optimization problem reduces to minimizing $\sum_{a \in \mathcal{I}} \pi_{\Theta_a}[1 - H(\rho_a)]$ over $\mathcal{T}(\mathbf{B}, \Pi)$. As mentioned earlier, finding a global optimal solution for this optimization problem is NP-complete [13]. Instead, we may take a top down approach and minimize the objective function by minimizing the term $C_a := \pi_{\Theta_a}[1 - H(\rho_a)]$ at each internal node, starting from the root node. Note that the only term that depends on the query chosen at node 'a' in this cost function is ρ_a. Hence the algorithm reduces to minimizing ρ_a (i.e., choosing a split as balanced as possible) at each internal node $a \in \mathcal{I}$.

In other words, greedy minimization of (2) is equivalent to GBS. In the next section, we show how this framework can be extended to derive greedy algorithms for the problems of group identification and object identification with exponential costs.

3 Extensions of GBS

3.1 Group Identification

In group identification[1], the goal is not to determine the unknown object $\theta \in \Theta$, rather the group to which it belongs, in as few queries as possible. Here, in addition to $\mathbf{B}$ and Π, the group labels for the objects are also provided, where the groups are assumed to be disjoint.

We denote a group identification problem by $(\mathbf{B}, \Pi, \mathbf{y})$, where $\mathbf{y} = (y_1, \cdots, y_M)$ denotes the group labels of the objects, $y_i \in \{1, \cdots, K\}$. Let $\{\Theta^k\}_{k=1}^K$ be the partition of Θ, where $\Theta^k = \{\theta_i \in \Theta : y_i = k\}$. It is important to note here that the group identification problem cannot be simply reduced to an object identification problem with groups $\{\Theta^1, \cdots, \Theta^K\}$ as "meta objects," since the objects within a group need not respond the same to each query. For instance, consider the toy example shown in Figure 1 where the objects θ_1, θ_2 and θ_3 belonging to group 1 cannot be collapsed into a 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 produce 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 Figure 1 where 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. Here, we propose a natural extension of GBS to the problem of group identification.

[1] Golovin et.al. [23] simultaneously studied the problem of group identification in the context of object identification with persistent noise. Their algorithm is an extension of that in [24].

	q_1	q_2	q_3	Group label, y	Π
θ_1	0	1	1	1	0.25
θ_2	1	1	0	1	0.25
θ_3	0	1	0	1	0.25
θ_4	1	0	0	2	0.25

Figure 1: Toy Example

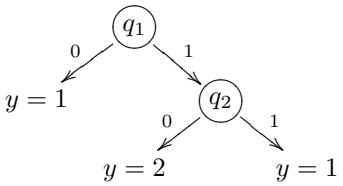

Figure 2: Decision tree constructed using GBS

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 $\mathcal{L}^k \subset \mathcal{L} = \{1, \cdots, L\}$ the set of leaves that terminate in group k. Similar to $\Theta^k \subseteq \Theta$, we denote by $\Theta^k_a \subseteq \Theta_a$ the set of objects belonging to group k that reach node 'a' in a tree. Also, in addition to the reduction factor defined in Section 2, we define a new parameter called the *group reduction factor* for each group $k \in \{1, \cdots, K\}$ at each internal node.

Definition 2 (Group reduction factor)**.** *Let T be a decision tree constructed on a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$. The* group reduction factor *for any group k at an internal node 'a' is defined by $\rho^k_a = \max\{\pi_{\Theta^k_{l(a)}}, \pi_{\Theta^k_{r(a)}}\}/\pi_{\Theta^k_a}$.*

Given $(\mathbf{B}, \Pi, \mathbf{y})$, 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 d_j denote the depth of leaf node $j \in \mathcal{L}$. Let random variable X denote the number of queries required to identify the group of an unknown object θ. Then, the expected number of queries required to identify the group of an unknown object using the given tree is equal to

$$L_1(\Pi) = \sum_{k=1}^{K} \Pr(\theta \in \Theta^k) \, \mathrm{E}[X | \theta \in \Theta^k] = \sum_{k=1}^{K} \pi_{\Theta^k} \left[\sum_{j \in \mathcal{L}^k} \frac{\pi_{\Theta_j}}{\pi_{\Theta^k}} d_j \right] \tag{3}$$

Theorem 2. *For any $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$, the expected number of queries required to identify the group of an unknown object is given by*

$$L_1(\Pi) = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_{\Theta^k_a}}{\pi_{\Theta_a}} H(\rho^k_a) \right] \tag{4}$$

where $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \cdots, \pi_{\Theta^K})$ denotes the probability distribution of the object groups induced by the labels $\mathbf{y}$ and $H(\cdot)$ denotes the Shannon entropy.

Note that the term in the summation in (4) is non-negative. Hence, the above result implies that $L_1(\Pi)$ is bounded below by the Shannon entropy of the probability distribution of the groups. It also follows from this result that this lower bound is achieved iff the reduction factor ρ_a is equal to 0.5 and the group reduction factors $\{\rho^k_a\}_{k=1}^K$ are equal to 1 at every internal node in the tree. Also, note that the result in Theorem 1 is a special case of this result where each group is of size 1 leading to $\rho^k_a = 1$ for all groups at every internal node.

Using this result, the problem of finding a decision tree with minimum $L_1(\Pi)$ can be formulated as:

$$\min_{T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})} \textstyle\sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_{\Theta^k_a}}{\pi_{\Theta_a}} H(\rho^k_a) \right]. \tag{5}$$

This optimization problem being a generalized version of that in (2) is NP-complete. Hence, we may take a top-down approach and minimize the objective function greedily by minimizing the term $\pi_{\Theta_a}[1 - H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_{\Theta^k_a}}{\pi_{\Theta_a}} H(\rho^k_a)]$ at each internal node, starting from the root node. Note that the terms that depend on the query chosen at node 'a' are ρ_a and ρ^k_a. Hence the algorithm reduces to minimizing $C_a := 1 - H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_{\Theta^k_a}}{\pi_{\Theta_a}} H(\rho^k_a)$ at each internal node 'a'.

Group-GBS (GGBS)

Initialize: $\mathcal{L} = \{\textit{root node}\}$, $Q_{root} = \emptyset$
while *some* $a \in \mathcal{L}$ *has more than one group*
  Choose query $q^* = \arg\min_{q \in Q \setminus Q_a} \; C_a(q)$
  Form child nodes $l(a), r(a)$
  Replace 'a' with $l(a), r(a)$ in $\mathcal{L}$
end

$C_a = 1 - H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_{\Theta_a^k}}{\pi_{\Theta_a}} H(\rho_a^k)$

Figure 3: Greedy algorithm for group identification

λ-GBS

Initialize: $\mathcal{L} = \{\textit{root node}\}$, $Q_{root} = \emptyset$
while *some* $a \in \mathcal{L}$ *has more than one object*
  Choose query $q^* = \arg\min_{q \in Q \setminus Q_a} \; C_a(q)$
  Form child nodes $l(a), r(a)$
  Replace 'a' with $l(a), r(a)$ in $\mathcal{L}$
end

$C_a = \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)})$

Figure 4: Greedy algorithm for object identification with exponential costs

Note that this objective function 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 term $\sum_k \frac{\pi_{\Theta_a^k}}{\pi_{\Theta_a}} H(\rho_a^k)$ favors queries that transfer an entire group of objects to one of its child nodes. This algorithm, which we refer to as Group Generalized Binary Search (GGBS), is summarized in Figure 3. Finally, as an interesting connection with greedy decision-tree algorithms for multi-class classification, it can be shown that GGBS is equivalent to the decision-tree splitting algorithm used in the C4.5 software package, based on the entropy impurity measure [25].

3.2 Exponential Costs

Now assume the cost of identifying an object is defined by $L_\lambda(\Pi) := \log_\lambda(\sum_i \pi_i \lambda^{d_i})$, where $\lambda > 1$ and d_i corresponds to the depth of object θ_i in a tree. In the limiting case where λ tends to 1 and ∞, this cost function reduces to the average depth and worst case depth, respectively. That is,

$$L_1(\Pi) = \lim_{\lambda \to 1} L_\lambda(\Pi) = \sum_{i=1}^{M} \pi_i d_i, \qquad L_\infty(\Pi) := \lim_{\lambda \to \infty} L_\lambda(\Pi) = \max_{i \in \{1, \cdots, M\}} d_i.$$

As mentioned in Section 2, GBS is tailored to minimize $L_1(\Pi)$, and hence may not produce a good suboptimal solution for the exponential cost function with $\lambda > 1$. Thus, we derive an extension of GBS for the problem of exponential costs. Here, we use a result by Campbell [26] which states that the exponential cost $L_\lambda(\Pi)$ of any tree T is bounded below by the α-Rényi entropy, given by $H_\alpha(\Pi) := \frac{1}{1-\alpha} \log_2 (\sum_i \pi_i^\alpha)$, where $\alpha = \frac{1}{1+\log_2 \lambda}$. We consider a general object identification problem and derive an explicit formula for the gap in this lower bound. We then use this formula to derive a family of greedy algorithms that minimize the exponential cost function $L_\lambda(\Pi)$ for $\lambda > 1$. Note that the entropy bound reduces to the Shannon entropy $H(\Pi)$ and $\log_2 M$, in the limiting cases where λ tends to 1 and ∞, respectively.

Theorem 3. *For any $\lambda > 1$ and any $T \in \mathcal{T}(\mathbf{B}, \Pi)$, the exponential cost $L_\lambda(\Pi)$ is given by*

$$\lambda^{L_\lambda(\Pi)} = \lambda^{H_\alpha(\Pi)} + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[(\lambda - 1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right]$$

where d_a denotes the depth of any internal node 'a' in the tree, Θ_a denotes the set of objects that reach node 'a', $\pi_{\Theta_a} = \sum_{\{i : \theta_i \in \Theta_a\}} \pi_i$, $\alpha = \frac{1}{1+\log_2 \lambda}$ and $\mathcal{D}_\alpha(\Theta_a) := \left[\sum_{\{i : \theta_i \in \Theta_a\}} \left(\frac{\pi_i}{\pi_{\Theta_a}} \right)^\alpha \right]^{1/\alpha}$.

The term in summation over internal nodes $\mathcal{I}$ in the above result corresponds to the gap in the Campbell's lower bound. This result suggests a top-down greedy approach to minimize $L_\lambda(\Pi)$, which is to minimize the term $(\lambda - 1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)})$ at each internal node, starting from the root node. Noting that the terms that depend on the query chosen at node 'a' are $\pi_{\Theta_{l(a)}}$, $\pi_{\Theta_{r(a)}}$, $\mathcal{D}_\alpha(\Theta_{l(a)})$ and $\mathcal{D}_\alpha(\Theta_{r(a)})$, this reduces to minimizing $C_a := \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)})$ at each internal node. This algorithm, which we refer to as λ-GBS, can be summarized as shown in Figure 4. Also, it can be shown by the application of L'Hôpital's rule that in the limiting case where $\lambda \to 1$, λ-GBS reduces to GBS, and in the case where $\lambda \to \infty$, λ-GBS reduces to GBS with uniform prior $\pi_i = 1/M$. The latter algorithm is GBS but with the true prior Π replaced by a uniform distribution.

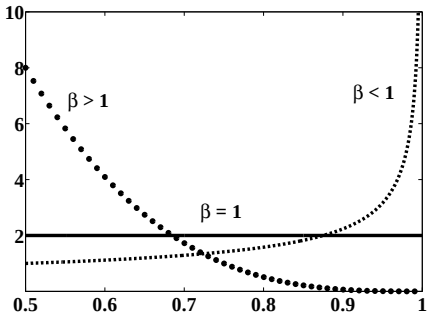

Figure 5: Beta distribution over the range $[0.5, 1]$ for different values of β when $\alpha = 1$

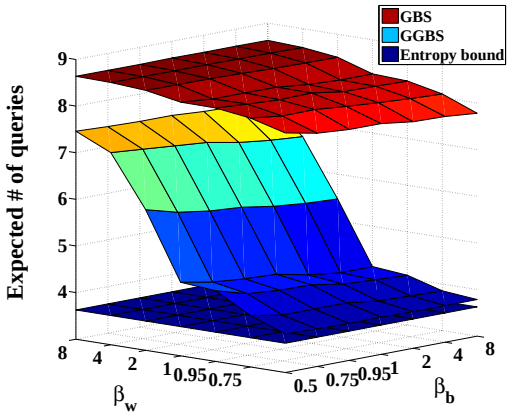

Figure 6: Expected number of queries required to identify the group of an object using GBS and GGBS

3.3 Group Identification with Exponential Costs

Finally, we complete our discussion by considering the problem of group identification with exponential costs. Here, the cost of identifying the group of an object given a tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$, is defined to be $L_\lambda(\Pi) = \log_\lambda \left(\sum_{j \in \mathcal{L}} \pi_{\Theta_j} \lambda^{d_j} \right)$, which reduces to (3) in the limiting case as $\lambda \to 1$, and to $\max_{j \in \mathcal{L}} d_j$, i.e., the worst case depth of the tree, in the case where $\lambda \to \infty$.

Theorem 4. *For any $\lambda > 1$ and any $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$, the exponential cost $L_\lambda(\Pi)$ of identifying the group of an object is given by*

$$\lambda^{L_\lambda(\Pi)} = \lambda^{H_\alpha(\Pi_{\mathbf{y}})} + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[(\lambda - 1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right]$$

where $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \cdots, \pi_{\Theta^K})$ denotes the probability distribution of the object groups induced by the labels $\mathbf{y}$, $\mathcal{D}_\alpha(\Theta_a) := \left[\sum_{k=1}^{K} \left(\frac{\pi_{\Theta_a^k}}{\pi_{\Theta_a}} \right)^\alpha \right]^{1/\alpha}$ with $\alpha = \frac{1}{1+\log_2 \lambda}$.

Note that the definition of $\mathcal{D}_\alpha(\Theta_a)$ in this theorem is a generalization of that in Theorem 3. As mentioned earlier, Theorems 1-3 are special cases of the above theorem, where Theorem 2 follows as $\lambda \to 1$ and Theorem 1 follows when each group is of size one in addition. This result also implies a top-down, greedy algorithm to minimize $L_\lambda(\Pi)$, which is to choose a query that minimizes $C_a := \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)})$ at each internal node. Once again, it can be shown by the application of L'Hôpital's rule that in the limiting case where $\lambda \to 1$, this reduces to GGBS, and in the case where $\lambda \to \infty$, this reduces to choosing a query that minimizes the maximum number of groups in the child nodes [27].

4 Performance of the Greedy Algorithms

We compare the performance of the proposed algorithms to that of GBS on synthetic data generated using different random data models.

4.1 Group Identification

For fixed $M = |\Theta|$ and $N = |Q|$, we consider a random data model where each query $q \in Q$ is associated with a pair of parameters $(\gamma_w(q), \gamma_b(q)) \in [0.5, 1]^2$. Here, $\gamma_w(q)$ reflects the correlation of the object responses *within* a group, and $\gamma_b(q)$ captures the correlation of object responses *between* groups. When $\gamma_w(q)$ is close to 0.5, each object within a group is equally likely to exhibit 0 or 1 as its response to query q, whereas, when it is close to 1, most of the objects within a group are highly likely to exhibit the same query response. Similarly, when $\gamma_b(q)$ 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 $\gamma_b(q)$ tends to 1, most of the

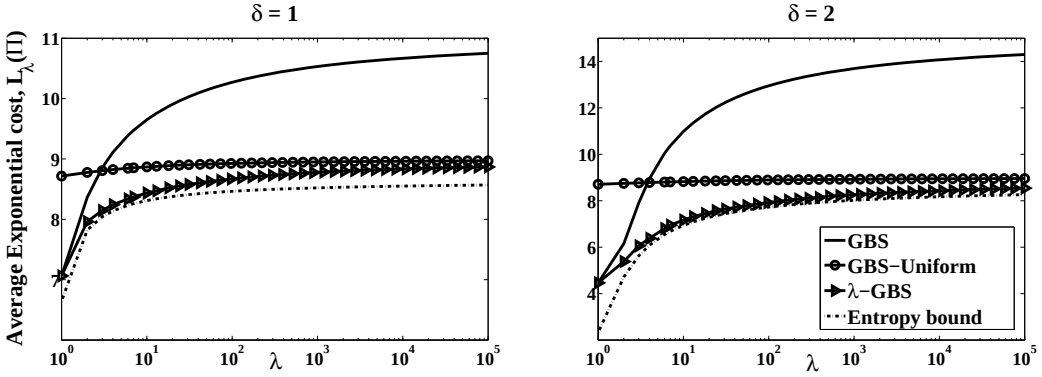

Figure 7: Exponential cost incurred in identifying an object using GBS and λ-GBS

groups are highly likely to exhibit the same response. Given these correlation values $(\gamma_w(q), \gamma_b(q))$ for a query q, the object responses to query q (i.e., the binary column of 0's and 1's corresponding to query q in $\mathbf{B}$) are generated as follows

1. Flip a fair coin to generate a Bernoulli random variable, x
2. For each group $k \in \{1, \cdots, K\}$, assign a binary label b_k, where $b_k = x$ with probability $\gamma_b(q)$
3. For each object in group k, assign b_k as the object response to q with probability $\gamma_w(q)$

Given the correlation parameters $(\gamma_w(q), \gamma_b(q))$, $\forall q \in Q$, a random dataset can be created by following the above procedure for each query.

We compare the performances of GBS and GGBS on random datasets generated using the above model. We demonstrate the results on datasets of size $N = 200$ (# of queries) and $M = 400$ (# of objects), where we randomly partitioned the objects into 15 groups and assumed a uniform prior on the objects. For each dataset, the correlation parameters are drawn from independent beta distributions over the range $[0.5, 1]$, i.e., $\gamma_w(q) \sim$ Beta$(1, \beta_w)$ and $\gamma_b(q) \sim$ Beta$(1, \beta_b)$ where $\beta_w, \beta_b \in \{0.5, 0.75, 0.95, 1, 2, 4, 8\}$. Figure 5 shows the density function (pdf) of Beta$(1, \beta)$ for different values of β. Note that $\beta = 1$ corresponds to a uniform distribution, while, for $\beta < 1$ the distribution is right skewed and for $\beta > 1$ the distribution is left skewed.

Figure 6 compares the mean value of the cost function $L_1(\Pi)$ for GBS and GGBS over 100 randomly generated datasets, for each value of (β_w, β_b). This shows the improved performance of GGBS over GBS in group identification. Especially, note that GGBS achieves performance close to the entropy bound as β_w decreases. This is due to the increased number of queries with $\gamma_w(q)$ close to 1 in the dataset. As the correlation parameter $\gamma_w(q)$ tends to 1, choosing that query keeps the groups intact, i.e., the group reduction factors ρ_a^k tend to 1 for these queries. Such queries offer significant gains in group identification, but can be overlooked by GBS.

4.2 Object Identification with Exponential Costs

We consider the same random data model as above where we set $K = M$, i.e., each group is comprised of one object. Thus, the only correlation parameter that determines the structure of the dataset is $\gamma_b(q), q \in Q$. Figure 7 demonstrates the improved performance of λ-GBS over standard GBS, and GBS with uniform prior, over a range of λ values, for a dataset generated using the above random data model with $\gamma_b(q) \sim$ Beta$(1, 1) =$ unif$[0.5, 1]$. Each curve in the figure corresponds to the average value of the cost function $L_\lambda(\Pi)$ as a function of λ over 100 repetitions. In each repetition, the prior is generated according to Zipf's law, i.e., $(j^{-\delta}/\sum_{i=1}^{M} i^{-\delta})_{j=1}^{M}$, $\delta \geq 0$, after randomly permuting the objects. Note that in the special case when $\delta = 0$, this reduces to the uniform distribution and as δ increases, it tends to a skewed distribution with most of the probability mass concentrated on few objects.

Similar experiments have been performed on datasets generated using $\gamma_b(q) \sim$ Beta(α, β) for different values of α, β. In all our experiments, we observed λ-GBS to be consistently performing better than both the standard GBS, and GBS with uniform prior. In addition, the performance of λ-GBS has been observed to be very close to that of the entropy bound. Finally, Figure 7 also reflects that λ-GBS converges to GBS as $\lambda \to 1$, and to GBS with uniform prior as $\lambda \to \infty$.

5 Conclusions

In this paper, we show that generalized binary search (GBS) is a top-down algorithm that greedily minimizes the expected number of queries required to identify an object. We then use this interpretation to extend GBS in two ways. First, we consider the case where the objects are partitioned into groups, and the goal is to identify only the group of the unknown object. Second, we consider the problem where the cost of identifying an object grows exponentially in the number of queries. The algorithms are derived in a common framework. In particular, we prove the exact formulas for the cost function in each case that close the gap between previously known lower bounds related to Shannon and Rényi entropy. These exact formulas are then optimized in a greedy, top-down manner to construct a decision tree. We demonstrate the improved performance of the proposed algorithms over GBS through simulations. An important open question and the direction of our future work is to relate these greedy algorithms to the global optimizer of their respective cost functions.

Acknowledgements

G. Bellala and C. Scott were supported in part by NSF Awards No. 0830490 and 0953135. S. Bhavnani was supported in part by CDC/NIOSH grant No. R21OH009441.

6 Appendix: Proof Sketch for Theorem 4

Define two new functions $\widetilde{L}_\lambda$ and $\widetilde{H}_\alpha$ as

$$\widetilde{L}_\lambda := \frac{1}{\lambda - 1}\left[\sum_{j\in\mathcal{L}} \pi_{\Theta_j}\lambda^{d_j} - 1\right] = \sum_{j\in\mathcal{L}} \pi_{\Theta_j}\left[\sum_{h=0}^{d_j-1}\lambda^h\right] \quad \text{and} \quad \widetilde{H}_\alpha := 1 - \frac{1}{\left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}}},$$

where $\widetilde{L}_\lambda$ is related to the cost function $L_\lambda(\Pi)$ as $\lambda^{L_\lambda(\Pi)} = (\lambda-1)\widetilde{L}_\lambda + 1$, and $\widetilde{H}_\alpha$ is related to the α-Rényi entropy $H_\alpha(\Pi_{\mathbf{y}})$ as

$$H_\alpha(\Pi_{\mathbf{y}}) = \frac{1}{1-\alpha}\log_2\sum_{k=1}^K \pi_{\Theta^k}^\alpha = \frac{1}{\alpha\log_2\lambda}\log_2\sum_{k=1}^K \pi_{\Theta^k}^\alpha = \log_\lambda\left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}} \tag{6a}$$

$$\implies \lambda^{H_\alpha(\Pi_{\mathbf{y}})} = \left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}} = \left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}}\widetilde{H}_\alpha + 1 \tag{6b}$$

where we use the definition of α, i.e., $\alpha = \frac{1}{1+\log_2\lambda}$ in (6a). Now, we note from Lemma 1 that

$$\widetilde{L}_\lambda = \sum_{a\in\mathcal{I}} \lambda^{d_a}\pi_{\Theta_a} \implies \lambda^{L_\lambda(\Pi)} = 1 + \sum_{a\in\mathcal{I}}(\lambda-1)\lambda^{d_a}\pi_{\Theta_a} \tag{7}$$

where d_a denotes the depth of internal node 'a' in the tree T. Similarly, we note from (6b) and Lemma 2 that

$$\lambda^{H_\alpha(\Pi_{\mathbf{y}})} = 1 + \sum_{a\in\mathcal{I}}\left[\pi_{\Theta_a}\mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}}\mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}}\mathcal{D}_\alpha(\Theta_{r(a)})\right]. \tag{8}$$

Finally, the result follows from (7) and (8) above.

Lemma 1. *The function $\widetilde{L}_\lambda$ can be decomposed over the internal nodes in a tree T, as $\widetilde{L}_\lambda = \sum_{a\in\mathcal{I}}\lambda^{d_a}\pi_{\Theta_a}$, where d_a denotes the depth of internal node $a \in \mathcal{I}$ and π_{Θ_a} is the probability mass of the objects at that node.*

Lemma 2. *The function $\widetilde{H}_\alpha$ can be decomposed over the internal nodes in a tree T, as*

$$\widetilde{H}_\alpha = \frac{1}{\left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}}}\sum_{a\in\mathcal{I}}\left[\pi_{\Theta_a}\mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}}\mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}}\mathcal{D}_\alpha(\Theta_{r(a)})\right]$$

where $\mathcal{D}_\alpha(\Theta_a) := \left[\sum_{k=1}^K\left(\frac{\pi_{\Theta_a^k}}{\pi_{\Theta_a}}\right)^\alpha\right]^{\frac{1}{\alpha}}$ and π_{Θ_a} denotes the probability mass of the objects at any internal node $a \in \mathcal{I}$.

The above two lemmas can be proved using induction over subtrees rooted at any internal node 'a' in the tree. The details may be found in the Supplemental Material.

References

[1] S. Dasgupta, "Analysis of a greedy active learning strategy," *Advances in Neural Information Processing Systems*, 2004.

[2] R. Nowak, "Generalized binary search," *Proceedings of the 46th Allerton Conference on Communications, Control and Computing*, pp. 568–574, 2008.

[3] ——, "Noisy generalized binary search," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1366–1374, 2009.

[4] D. Golovin and A. Krause, "Adaptive Submodularity: A new approach to active learning and stochastic optimization," *In Proceedings of International Conference on Learning Theory (COLT)*, 2010.

[5] D. W. Loveland, "Performance bounds for binary testing with arbitrary weights," *Acta Informatica*, 1985.

[6] F. Yu, F. Tu, H. Tu, and K. Pattipati, "Multiple disease (fault) diagnosis with applications to the QMR-DT problem," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1187–1192, October 2003.

[7] J. Shiozaki, H. Matsuyama, E. O'Shima, and M. Iri, "An improved algorithm for diagnosis of system failures in the chemical process," *Computational Chemical Engineering*, vol. 9, no. 3, pp. 285–293, 1985.

[8] 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," *Proceedings of American Medical Informatics Association*, 2007.

[9] D. Geman and B. Jedynak, "An active testing model for tracking roads in satellite images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 1–14, 1996.

[10] M. J. Swain and M. A. Stricker, "Promising directions in active vision," *International Journal of Computer Vision*, vol. 11, no. 2, pp. 109–126, 1993.

[11] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi, "Approximation algorithms for optimal decision trees and adaptive TSP problems," 2010, available online at arXiv.org:1003.0722.

[12] M. Garey, "Optimal binary identification procedures," *SIAM Journal on Applied Mathematics*, vol. 23(2), pp. 173–186, 1972.

[13] L. Hyafil and R. Rivest, "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters*, vol. 5(1), pp. 15–17, 1976.

[14] S. R. Kosaraju, T. M. Przytycka, and R. S. Borgstrom, "On an optimal split tree problem," *Proceedings of 6th International Workshop on Algorithms and Data Structures, WADS*, pp. 11–14, 1999.

[15] R. M. Goodman and P. Smyth, "Decision tree design from a communication theory standpoint," *IEEE Transactions on Information Theory*, vol. 34, no. 5, 1988.

[16] P. A. Humblet, "Generalization of Huffman coding to minimize the probability of buffer overflow," *IEEE Transactions on Information Theory*, vol. IT-27, no. 2, pp. 230–232, March 1981.

[17] F. Schulz, "Trees with exponentially growing costs," *Information and Computation*, vol. 206, 2008.

[18] M. B. Baer, "Rényi to Rényi - source coding under seige," *Proceedings of IEEE International Symposium on Information Theory*, pp. 1258–1262, July 2006.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley, 1991.

[20] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the Institute of Radio Engineers*, 1952.

[21] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 379 – 423, July 1948.

[22] R. M. Fano, *Transmission of Information*. MIT Press, 1961.

[23] D. Golovin, D. Ray, and A. Krause, "Near-optimal Bayesian active learning with noisy observations," *to appear in the Proceedings of the Neural Information Processing Systems (NIPS)*, 2010.

[24] S. Dasgupta, "Coarse sample complexity bounds for active learning," *Advances in Neural Information Processing Systems*, 2006.

[25] G. Bellala, S. Bhavnani, and C. Scott, "Group-based query learning for rapid diagnosis in time-critical situations," Tech. Rep., 2009, available online at arXiv.org:0911.4511.

[26] L. L. Campbell, "A coding problem and Rényi's entropy," *Information and Control*, vol. 8, no. 4, pp. 423–429, August 1965.

[27] G. Bellala, S. Bhavnani, and C. Scott, "Query learning with exponential query costs," Tech. Rep., 2010, available online at arXiv.org:1002.4019.

7 Supplementary Material: Complete Proof of Theorem 4

Define two new functions $\widetilde{L}_\lambda$ and $\widetilde{H}_\alpha$ as

$$\widetilde{L}_\lambda := \frac{1}{\lambda - 1}\left[\sum_{j\in\mathcal{L}} \pi_{\Theta_j}\lambda^{d_j} - 1\right] = \sum_{j\in\mathcal{L}} \pi_{\Theta_j}\left[\sum_{h=0}^{d_j-1}\lambda^h\right]$$

$$\widetilde{H}_\alpha := 1 - \frac{1}{\left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}}},$$

where $\widetilde{L}_\lambda$ is related to the cost function $L_\lambda(\Pi)$ as

$$\lambda^{L_\lambda(\Pi)} = (\lambda - 1)\widetilde{L}_\lambda + 1, \tag{9}$$

and $\widetilde{H}_\alpha$ is related to the α-Rényi entropy $H_\alpha(\Pi_{\mathbf{y}})$ as

$$H_\alpha(\Pi_{\mathbf{y}}) = \frac{1}{1-\alpha}\log_2\sum_{k=1}^K \pi_{\Theta^k}^\alpha = \frac{1}{\alpha\log_2\lambda}\log_2\sum_{k=1}^K \pi_{\Theta^k}^\alpha = \log_\lambda\left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}} \tag{10a}$$

$$\implies \lambda^{H_\alpha(\Pi_{\mathbf{y}})} = \left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}} = \left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}}\widetilde{H}_\alpha + 1 \tag{10b}$$

where we use the definition of α, i.e., $\alpha = \frac{1}{1+\log_2\lambda}$ in (10a).

Now, we note from Lemma 3 that $\widetilde{L}_\lambda$ can be decomposed as

$$\widetilde{L}_\lambda = \sum_{a\in\mathcal{I}} \lambda^{d_a}\pi_{\Theta_a}$$

$$\implies \lambda^{L_\lambda(\Pi)} = 1 + \sum_{a\in\mathcal{I}}(\lambda-1)\lambda^{d_a}\pi_{\Theta_a} \tag{11}$$

where d_a denotes the depth of internal node 'a' in the tree T. Similarly, note from Lemma 4 that $\widetilde{H}_\alpha$ can be decomposed as

$$\widetilde{H}_\alpha = \frac{1}{\left(\sum_{k=1}^K \pi_{\Theta^k}^\alpha\right)^{\frac{1}{\alpha}}}\sum_{a\in\mathcal{I}}\left[\pi_{\Theta_a}\mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}}\mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}}\mathcal{D}_\alpha(\Theta_{r(a)})\right]$$

$$\implies \lambda^{H_\alpha(\Pi_{\mathbf{y}})} = 1 + \sum_{a\in\mathcal{I}}\left[\pi_{\Theta_a}\mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}}\mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}}\mathcal{D}_\alpha(\Theta_{r(a)})\right]. \tag{12}$$

Finally, the result follows from (11) and (12) above.

Lemma 3. *The function $\widetilde{L}_\lambda$ can be decomposed over the internal nodes in a tree T, as*

$$\widetilde{L}_\lambda = \sum_{a\in\mathcal{I}} \lambda^{d_a}\pi_{\Theta_a}$$

where d_a denotes the depth of internal node $a \in \mathcal{I}$ and π_{Θ_a} is the probability mass of the objects at that node.

Proof. Let T_a denote a subtree from any internal node 'a' in the tree T and let $\mathcal{I}_a, \mathcal{L}_a$ denote the set of internal nodes and leaf nodes in the subtree T_a, respectively. Then, define $\widetilde{L}_\lambda^a$ in the subtree T_a to be

$$\widetilde{L}_\lambda^a = \textstyle\sum_{j\in\mathcal{L}_a}\frac{\pi_{\Theta_j}}{\pi_{\Theta_a}}\left[\sum_{h=0}^{d_j^a-1}\lambda^h\right]$$

where d_j^a denotes the depth of leaf node $j \in \mathcal{L}_a$ in the subtree T_a.

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

$$\pi_{\Theta_a}\widetilde{L}_\lambda^a = \sum_{s\in\mathcal{I}_a}\lambda^{d_s^a}\pi_{\Theta_s} \tag{13}$$

where d_s^a denotes the depth of internal node $s \in \mathcal{I}_a$ in the subtree T_a.

The relation holds trivially for any subtree T_a rooted at an internal node $a \in \mathcal{I}$ whose both child nodes terminate as leaf nodes, with both the left hand side and the right hand side of the expression equal to π_{Θ_a}. Now, consider a subtree T_a rooted at an internal node $a \in \mathcal{I}$ whose left child (or right child) alone terminates as a leaf node. Assume that the above relation holds true for the subtree rooted at the right child of node 'a'. Then,

$$\begin{aligned}
\pi_{\Theta_a} \widetilde{L}^a_\lambda &= \sum_{j \in \mathcal{L}_a} \pi_{\Theta_j} \left[\sum_{h=0}^{d^a_j - 1} \lambda^h \right] \\
&= \sum_{\{j \in \mathcal{L}_a : d^a_j = 1\}} \pi_{\Theta_j} + \sum_{\{j \in \mathcal{L}_a : d^a_j > 1\}} \pi_{\Theta_j} \left[\sum_{h=0}^{d^a_j - 1} \lambda^h \right] \\
&= \pi_{\Theta_{l(a)}} + \sum_{\{j \in \mathcal{L}_a : d^a_j > 1\}} \pi_{\Theta_j} \left[1 + \lambda \sum_{h=0}^{d^a_j - 2} \lambda^h \right] \\
&= \pi_{\Theta_a} + \lambda \sum_{j \in \mathcal{L}_{r(a)}} \pi_{\Theta_j} \left[\sum_{h=0}^{d^{r(a)}_j - 1} \lambda^h \right] \\
&= \pi_{\Theta_a} + \lambda \sum_{s \in \mathcal{I}_{r(a)}} \lambda^{d^{r(a)}_s} \pi_{\Theta_s}
\end{aligned}$$

where the last step follows from the induction hypothesis. Finally, consider a subtree T_a rooted at an internal node $a \in \mathcal{I}$ whose neither child node terminates as a leaf node. Assume that the relation in (13) holds true for the subtrees rooted at its left and right child nodes. Then,

$$\begin{aligned}
\pi_{\Theta_a} \widetilde{L}^a_\lambda &= \sum_{j \in \mathcal{L}_a} \pi_{\Theta_j} \left[\sum_{h=0}^{d^a_j - 1} \lambda^h \right] \\
&= \sum_{j \in \mathcal{L}_{l(a)}} \pi_{\Theta_j} \left[1 + \lambda \sum_{h=0}^{d^a_j - 2} \lambda^h \right] + \sum_{j \in \mathcal{L}_{r(a)}} \pi_{\Theta_j} \left[1 + \lambda \sum_{h=0}^{d^a_j - 2} \lambda^h \right] \\
&= \pi_{\Theta_a} + \lambda \sum_{j \in \mathcal{L}_{l(a)}} \pi_{\Theta_j} \left[\sum_{h=0}^{d^{l(a)}_j - 1} \lambda^h \right] + \lambda \sum_{j \in \mathcal{L}_{r(a)}} \pi_{\Theta_j} \left[\sum_{h=0}^{d^{r(a)}_j - 1} \lambda^h \right] \\
&= \pi_{\Theta_a} + \lambda \left[\sum_{s \in \mathcal{I}_{l(a)}} \lambda^{d^{l(a)}_s} \pi_{\Theta_s} + \sum_{s \in \mathcal{I}_{r(a)}} \lambda^{d^{r(a)}_s} \pi_{\Theta_s} \right] = \sum_{s \in \mathcal{I}_a} \lambda^{d^a_s} \pi_{\Theta_s}
\end{aligned}$$

thereby completing the induction. Finally, the result follows by applying the relation in (13) to the tree T whose probability mass at the root node, $\pi_{\Theta_a} = 1$. □

Lemma 4. *The function $\widetilde{H}_\alpha$ can be decomposed over the internal nodes in a tree T, as*

$$\widetilde{H}_\alpha = \frac{1}{\left(\sum_{k=1}^K \pi^\alpha_{\Theta^k} \right)^{\frac{1}{\alpha}}} \sum_{a \in \mathcal{I}} \left[\pi_{\Theta_a} \mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}} \mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right]$$

where $\mathcal{D}_\alpha(\Theta_a) := \left[\sum_{k=1}^K \left(\frac{\pi_{\Theta^k_a}}{\pi_{\Theta_a}} \right)^\alpha \right]^{\frac{1}{\alpha}}$ and π_{Θ_a} denotes the probability mass of the objects at any internal node $a \in \mathcal{I}$.

Proof. Let T_a denote a subtree from any internal node 'a' in the tree T and let $\mathcal{I}_a$ denote the set of internal nodes in the subtree T_a. Then, define $\widetilde{H}^a_\alpha$ in a subtree T_a to be

$$\widetilde{H}^a_\alpha = 1 - \frac{\pi_{\Theta_a}}{\left[\sum_{k=1}^K \pi^\alpha_{\Theta^i_a} \right]^{\frac{1}{\alpha}}}$$

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

$$\left[\sum_{k=1}^K \pi^\alpha_{\Theta^k_a} \right]^{\frac{1}{\alpha}} \widetilde{H}^a_\alpha = \sum_{s \in \mathcal{I}_a} \left[\pi_{\Theta_s} \mathcal{D}_\alpha(\Theta_s) - \pi_{\Theta_{l(s)}} \mathcal{D}_\alpha(\Theta_{l(s)}) - \pi_{\Theta_{r(s)}} \mathcal{D}_\alpha(\Theta_{r(s)}) \right] \tag{14}$$

Note that the relation holds trivially for any subtree T_a rooted at an internal node $a \in \mathcal{I}$ whose both child nodes terminate as leaf nodes. Now, consider a subtree T_a rooted at any other internal node $a \in \mathcal{I}$. Assume the above relation holds true for the subtrees rooted at its left and right child nodes. Then,

$$\begin{aligned}
\left[\sum_{k=1}^{K} \pi_{\Theta_a^k}^{\alpha}\right]^{\frac{1}{\alpha}} \widetilde{H}_{\alpha}^{a} &= \left[\sum_{k=1}^{K} \pi_{\Theta_a^k}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_a} = \left[\sum_{k=1}^{K} \pi_{\Theta_a^k}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_{l(a)}} - \pi_{\Theta_{r(a)}} \\
&= \left[\sum_{k=1}^{K} \pi_{\Theta_a^k}^{\alpha}\right]^{\frac{1}{\alpha}} - \left[\sum_{k=1}^{K} \pi_{\Theta_{l(a)}^k}^{\alpha}\right]^{\frac{1}{\alpha}} - \left[\sum_{k=1}^{K} \pi_{\Theta_{r(a)}^k}^{\alpha}\right]^{\frac{1}{\alpha}} \\
&\quad + \left(\left[\sum_{k=1}^{K} \pi_{\Theta_{l(a)}^k}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_{l(a)}}\right) + \left(\left[\sum_{k=1}^{K} \pi_{\Theta_{r(a)}^k}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_{r(a)}}\right) \\
&= \left[\pi_{\Theta_a} \mathcal{D}_{\alpha}(\Theta_a) - \pi_{\Theta_{l(a)}} \mathcal{D}_{\alpha}(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}} \mathcal{D}_{\alpha}(\Theta_{r(a)})\right] \\
&\quad + \left[\sum_{k=1}^{K} \pi_{\Theta_{l(a)}^k}^{\alpha}\right]^{\frac{1}{\alpha}} \widetilde{H}_{\alpha}^{l(a)} + \left[\sum_{k=1}^{K} \pi_{\Theta_{r(a)}^k}^{\alpha}\right]^{\frac{1}{\alpha}} \widetilde{H}_{\alpha}^{r(a)} \\
&= \sum_{s \in \mathcal{I}_a} \left[\pi_{\Theta_s} \mathcal{D}_{\alpha}(\Theta_s) - \pi_{\Theta_{l(s)}} \mathcal{D}_{\alpha}(\Theta_{l(s)}) - \pi_{\Theta_{r(s)}} \mathcal{D}_{\alpha}(\Theta_{r(s)})\right]
\end{aligned}$$

where the last step follows from the induction hypothesis. Finally, the result follows by applying the relation in (14) to the tree T. □

8 Proof of Theorem 3

The result in Theorem 3 follows from the above result where each group is of size one, thereby reducing $\mathcal{D}_{\alpha}(\Theta_a)$ to

$$\mathcal{D}_{\alpha}(\Theta_a) = \left[\sum_{i=1}^{M} \left(\frac{\pi_i \mathbb{I}_{\{\theta_i \in \Theta_a\}}}{\pi_{\Theta_a}}\right)^{\alpha}\right]^{\frac{1}{\alpha}} = \left[\sum_{\{i:\theta_i \in \Theta_a\}} \left(\frac{\pi_i}{\pi_{\Theta_a}}\right)^{\alpha}\right]^{\frac{1}{\alpha}},$$

where $\mathbb{I}_{\{\theta_i \in \Theta_a\}}$ is the indicator function which takes the value one when $\theta_i \in \Theta_a$, and zero otherwise.

9 Proof of Theorem 2

The result in Theorem 2 is a special case of that in Theorem 4 when $\lambda \to 1$. It follows by taking the logarithm to the base λ on both sides of equation

$$\lambda^{L_{\lambda}(\Pi)} = \lambda^{H_{\alpha}(\Pi_{\mathbf{y}})} + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[(\lambda - 1)\lambda^{d_a} - \mathcal{D}_{\alpha}(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_{\alpha}(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_{\alpha}(\Theta_{r(a)})\right],$$

and then finding the limit as $\lambda \to 1$.

Using L'Hôpital's rule, the left hand side (LHS) of the equation reduces to

$$\lim_{\lambda \to 1} \log_{\lambda}(\text{LHS}) = \lim_{\lambda \to 1} L_{\lambda}(\Pi) = \sum_{j \in \mathcal{L}} \pi_{\Theta_j} d_j,$$

where $L_{\lambda}(\Pi) = \log_{\lambda}\left(\sum_{j \in \mathcal{L}} \pi_{\Theta_j} \lambda^{d_j}\right)$. Similarly, the right hand side (RHS) of the equation reduces to

$$\lim_{\lambda \to 1} \log_{\lambda}(\text{RHS}) = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - \left(H(\Theta_a) - \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} H(\Theta_{l(a)}) - \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} H(\Theta_{r(a)})\right)\right],$$

where $H(\Theta_a) = -\sum_{k=1}^{K} \frac{\pi_{\Theta_a^k}}{\pi_{\Theta_a}} \log_2\left(\frac{\pi_{\Theta_a^k}}{\pi_{\Theta_a}}\right)$.

Finally, the result follows by noticing that

$$
\begin{aligned}
H(\Theta_a) &- \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} H(\Theta_{l(a)}) - \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} H(\Theta_{r(a)}) \\
&= \frac{1}{\pi_{\Theta_a}} \left[\sum_{k=1}^{K} \pi_{\Theta_a^k} \log_2 \left(\frac{\pi_{\Theta_a}}{\pi_{\Theta_a^k}} \right) - \pi_{\Theta_{l(a)}^k} \log_2 \left(\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_{l(a)}^k}} \right) - \pi_{\Theta_{r(a)}^k} \log_2 \left(\frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_{r(a)}^k}} \right) \right] && \text{(15a)} \\
&= \frac{1}{\pi_{\Theta_a}} \left[\sum_{k=1}^{K} \pi_{\Theta_{l(a)}^k} \log_2 \left(\frac{\pi_{\Theta_a}}{\pi_{\Theta_{l(a)}}} \cdot \frac{\pi_{\Theta_{l(a)}^k}}{\pi_{\Theta_a^k}} \right) + \pi_{\Theta_{r(a)}^k} \log_2 \left(\frac{\pi_{\Theta_a}}{\pi_{\Theta_{r(a)}}} \cdot \frac{\pi_{\Theta_{r(a)}^k}}{\pi_{\Theta_a^k}} \right) \right] && \text{(15b)} \\
&= \frac{1}{\pi_{\Theta_a}} \left[\pi_{\Theta_{l(a)}} \log_2 \left(\frac{\pi_{\Theta_a}}{\pi_{\Theta_{l(a)}}} \right) + \pi_{\Theta_{r(a)}} \log_2 \left(\frac{\pi_{\Theta_a}}{\pi_{\Theta_{r(a)}}} \right) \right. \\
&\qquad \left. + \sum_{k=1}^{K} \pi_{\Theta_{l(a)}^k} \log_2 \left(\frac{\pi_{\Theta_{l(a)}^k}}{\pi_{\Theta_a^k}} \right) + \pi_{\Theta_{r(a)}^k} \log_2 \left(\frac{\pi_{\Theta_{r(a)}^k}}{\pi_{\Theta_a^k}} \right) \right] && \text{(15c)} \\
&= H(\rho_a) + \sum_{k=1}^{K} \frac{\pi_{\Theta_a^k}}{\pi_{\Theta_a}} H(\rho_a^k), && \text{(15d)}
\end{aligned}
$$

where (15b) follows from (15a) by using the relation $\pi_{\Theta_a^k} = \pi_{\Theta_{l(a)}^k} + \pi_{\Theta_{r(a)}^k}$, and (15d) follows from (15c) using the definitions of ρ_a and ρ_a^k.

10 Proof of Theorem 1

The result in Theorem 1 follows from the above result where each group is of size one, thereby having $\rho_a^k = 1$ $\forall k$ at each internal node $a \in \mathcal{I}$. It can also be derived as the limiting case of the relation in Theorem 3 by taking logarithm to the base λ on both sides of the relation and letting $\lambda \to 1$.