# Mode Estimation with Partial Feedback

Charles Arnal, Vivien Cabannes, Vianney Perchet

*Preprint*



# MODE ESTIMATION WITH PARTIAL FEEDBACK

BY CHARLES ARNAL[*1], VIVIEN CABANNES[*2] AND VIANNEY PERCHET[3]

[1]*Datashape, Inria, Saclay, France*

[2]*Fundamental Artificial Intelligence Research (FAIR), Meta AI, New York, USA*

[3]*Center for Research in Economics and Statistics (CREST), ENSAE, Palaiseau, France*

The combination of lightly supervised pre-training and online fine-tuning has played a key role in recent AI developments. These new learning pipelines call for new theoretical frameworks. In this paper, we formalize core aspects of weakly supervised and active learning with a simple problem: the estimation of the mode of a distribution using partial feedback. We show how entropy coding allows for optimal information acquisition from partial feedback, develop coarse sufficient statistics for mode identification, and adapt bandit algorithms to our new setting. Finally, we combine those contributions into a statistically and computationally efficient solution to our problem.

## CONTENTS



[*]These authors contributed equally to this work.

*MSC2020 subject classifications:* Primary 62L05, 62B86; secondary 62D10, 62B10.

*Keywords and phrases:* Active Learning, Partial Feedback, Entropy Coding, Coarse Search, Best Arm Identification.

**1. Introduction.** The mode of a distribution is a fundamental concept in statistics, serving as a key identifier for the most likely event to occur. For instance, identifying modes of conditional distributions is the main task of classification algorithms –classification consists in learning the mapping $f^*(x) = \arg\max_y p(y \,|\, x)$ for a joint distribution $p$ over inputs $x$ and classes $y$. Traditionally, datasets were small enough to fully annotate samples before learning the modes of the underlying distributions. However, with the increasing scale of machine learning problems, data collection has become a significant part of machine learning pipelines. This is illustrated by the substantial efforts dedicated to data preprocessing to train foundational AI models (see, e.g., OpenAI, 2023; Touvron et al., 2023). Moreover, it is foreseeable that future models will incorporate annotation feedback loops. Indeed, fine-tuning foundational models already relies on various active learning strategies, such as reinforcement learning with human feedback (Ziegler et al., 2020) and partial annotations (Zhu, Jordan and Jiao, 2023). This makes theories of weakly-supervised, and active learning highly relevant to the machine learning community. This paper introduces one of the simplest setups to combine active and weakly-supervised learning. It focuses on partial annotations, and searches for the best algorithms to identify the mode of a distribution given a budget of annotations.

1.1. *The Mode Estimation with Partial Feedback Problem.* The task at hand is the identification of the mode of a distribution $p \in P(\mathcal{Y})$, where $\mathcal{Y}$ is a set of $m$ elements or classes, through partial feedback. Let us denote as $\mathcal{Y} = \{y_1, \dots, y_m\}$ this set of classes, and assume that the $m$ elements $y_1, \dots, y_m$ are indexed by decreasing probability, i.e. $p(y_i) \geq p(y_{i+1})$ for any $i < m$. The goal is to find the most probable value $y_1$ of $\mathcal{Y}$, which we assume to be unique for simplicity, i.e.

$$y_1 = \arg\max_{y \in \mathcal{Y}} \mathbb{P}_{Y \sim p}(Y = y). \tag{1}$$

To estimate the mode, we assume the existence of independent samples $(Y_j)$ distributed according to $p$. However, the practitioner does not directly observe the samples; instead, they can sequentially acquire weak information on them. Formally, at each time $t \in \mathbb{N}$, the practitioner selects an index $j_t \in \mathbb{N}$ and a set of potential labels $S_t \subset \mathcal{Y}$, and ask whether the sample $Y_{j_t}$ belongs to $S_t$ or not, leading to the observation of the binary variable

$$\mathbf{1}_{Y_{j_t} \in S_t}. \tag{2}$$

We call the process of checking whether $Y_{j_t} \in S_t$ a *query*, leading to the following problem description:

PROBLEM 1. *Design a sequence of queries $(\mathbf{1}_{Y_{j_t} \in S_t})$ to efficiently estimate the mode of $p$, where each query $\mathbf{1}_{Y_{j_t} \in S_t}$ can depend on the previous observations of $(\mathbf{1}_{Y_{j_s} \in S_s})_{s<t}$.*

The overarching objective is to design and analyze efficient algorithms that output a good estimate of the mode using a minimal number of queries. Although real-world applications often come with peculiarities, Problem 1 is generic enough that its resolution should shed light on a large variety of online learning tasks with partial feedback.

Among possible variants of Problem 1, one could forbid the experimenter from asking more than one query per sample, i.e. forcing $j_t = t$, add a contextual variable $X \in \mathcal{X}$ that conditions the distribution of $\mathbf{1}_{Y \in S}$, assume that some queries are cheaper to make than others, or consider cases where random noise affects the observations $\mathbf{1}_{Y \in S}$. Rather than the mode $y_1$ of the distribution $p$, we may want to identify instead one or all the classes $y$ such that $p(y) \geq p(y_1) - \epsilon$ for some $\epsilon > 0$. Finally, more structure could be added to the set $\mathcal{Y}$, rather than let it be a collection of unrelated classes without any meaningful interactions. In particular, the queried sets $S$ could be restricted to belong to some predefined collection of sets $\mathcal{S}$ –e.g., each class in $\mathcal{Y}$ represents an animal species, and you can ask whether the animal $Y$ is a feline, but not whether it belongs to {cat, wolf, common snapping turtle}. More generally, the observation of $\mathbf{1}_{Y \in S}$ could be replaced by a random variable $F(Y, S)$ for some function $F$, so that $F(Y, S)$ need not be discrete or equal to $\sum_{y \in S} F(Y, S)$.

Problem 1 and its variations appear in many contexts. A natural illustration would be a content-providing service, such as a social network app with a scrolling-centric interface, that tries to identify which type of content the user is most likely to like from a collection $\mathcal{Y}$ of possible topics. The app shows a batch of posts or videos to the user (which corresponds to the choice of a set $S$), and receives some measure of user satisfaction in return (such as their scrolling time); from this information, the app must design future batches of content and find the user's favourite topics. Another natural example comes from advertising: a hotel can compose online ads using various combinations of pictures from a set $\mathcal{Y}$ of photos of a given room. Each combination corresponds to a set $S$, and the variable $\mathbf{1}_{Y \in S}$ is equal to 1 when online visitor $Y$ clicks on the ad. The hotel tries to identify which pictures maximize the chances of the ad being clicked. Similar situations can also arise in experimental sciences: a biologist tries to understand which genes $y$ in a collection $\mathcal{Y}$ of genes considered have the strongest effect on a certain property. To that end, they can test whether activating a set $S$ of genes results in the expression of the property.

1.2. *Related Work.* In terms of related problems, Problem 1 bears resemblances to the active labeling framework introduced by Cabannes et al. (2022). This framework was motivated by dynamic pricing (Cesa-Bianchi, Cesari and Perchet, 2019), active ranking (Jamieson and Nowak, 2011; Braverman, Mao and Peres, 2019), and hierarchical classification (Cesa-Bianchi, Gentile and Zaniboni, 2006; Gangaputra and Geman, 2006). Additionally, a clear connection can be drawn with the well-studied task of best-arm identification in multi-armed bandit settings (Bubeck, Munos and Stoltz, 2010),[1] particularly in the context of combinatorial bandits (Chen et al., 2016) and even more precisely of transductive linear bandits (Fiez et al., 2019).

In terms of techniques, our algorithms draw their inspirations from the entropic coding schemes of Huffman (1952) and Vitter (1987), the elimination algorithm of Even-Dar, Mannor and Mansour (2006), together with the doubling trick of Auer et al. (1995). Finally, we notice that the expectation-maximum (EM) algorithm of Dempster, Laird and Rubin (1977) was notably motivated by the estimation of probability from partial observations, although we did not pursue their approximate Bayesian approach.

Philosophically, our approach was motivated by the fact that Kolmogorov capacity, hence entropy coding, lies at the heart of statistical learning theory (see, e.g., Cucker and Smale (2002), as well as Grünwald (2007)), suggesting that a contextual version of our setup might provide insightful perspectives on active learning.

[1] The connection is apparent if we force $i_t = t$ and $S_t$ to be a singleton for every $t \in \mathbb{N}$.

| | | |
|---|---|---|
| Exhaustive Search Algorithm 3 | $\Delta_2^{-2} + \Delta_2^{-2} \log_2(m)$ | |
| ↓ | | |
| Adaptive Search Algorithm 4 | $\Delta_2^{-2} + \Delta_2^{-2} \left( \sum_{i \geq 1} p(y_i) \lvert \log_2 p(y_i) \rvert \right)$ | ⟶ Elimination Algorithm 8: $\Delta_2^{-2} + \sum_{i>1} p(y_i) \Delta_i^{-2} \lvert \log_2 p(y_i) \rvert$ |
| ↓ | | ↓ |
| Truncated Search Algorithm 7 | $\Delta_2^{-2} + \Delta_2^{-2} \lvert \log_2 p(y_1) \rvert$ | ⟶ Set Elimination Algorithm 9: $\Delta_2^{-2} + \left( \sum_{i \geq 1} p(y_i) \Delta_i^{-2} \right) \lvert \log_2 p(y_1) \rvert$ |

TABLE 1

*Summary of the coefficient $\alpha$ as per (4) found for different methods, up to universal constants when $p(y_1)$ is bounded away from 1, and with $\Delta_1 = \Delta_2$. The arrows indicate improvements.*

1.3. *Performance Metrics.* Various performance metrics can be applied to algorithms that output a guess $\hat{y}$ of the mode of $p$ after a certain number of queries. We will focus on the probability of error, i.e., on the minimal $\delta$ such that

$$\mathbb{E}[\mathbf{1}_{\hat{y} \neq y_1}] = \mathbb{P}(\hat{y} \neq y_1) \leq \delta, \tag{3}$$

as a function of the number of queries $T$. Here, the randomness is inherited from that of the samples $(Y_j)$ and of the algorithm. Equation (3) can be seen as the "risk" associated with the "zero-one loss", which is used in classification problems. Other reasonable metrics include the expected difference $\mathbb{E}[p(y_1) - p(\hat{y})]$, or the probability of $\epsilon$-approximate success $\mathbb{P}(p(\hat{y}) \geq p(y_1) - \epsilon)$. Despite those variations, the principles behind our algorithms can be readily adapted to different metrics.

Among additional nuances, in some contexts, a user might fix in advance a "confidence" level $\delta$, and be interested in algorithms that minimize the (expected) number of queries $T$ to achieve it. Reciprocally, one might fix a "budget" of queries $T$, and design algorithms that minimize the probability $\delta$ of false prediction using those queries. These settings typically lead to equations of the shape

$$\mathbb{E}[T] \leq \ln(C_1/\delta)\alpha_1, \qquad \text{or respectively} \qquad \delta \leq C_2 \exp(-T/\alpha_2), \tag{4}$$

where the constants $\alpha_i$ and $C_i$ depend on the unknown distribution $p$. Once again, algorithm design principles are often agnostic to the settings differentiation, usually allowing the conversion of one type of bound into the other. To offer quantitative comparisons, we focus on the joint asymptotic behavior with respect to $\delta$ and $T$, which is determined by the constants $\alpha$, removing $C$ from the picture. This provides a simple metric: the smaller $\alpha$, the better the algorithm.

Note that our proofs also yield guarantees on the sample complexity of our algorithms; however, the expected number of queries required is a more relevant quantity, as it captures both the sample efficiency of an algorithm and the efficiency with which it extracts the necessary information from each sample, which is a key aspect of our setting.

1.4. *Summary of Contributions.* Our first contribution is to *introduce a new problem, Problem 1, which captures key aspects of most online learning tasks with partial feedback.* It is simple enough to allow for rigorous theory to be developed, yet naturally generalizes to match practical scenarios from the real world. We introduce several important ideas to efficiently solve Problem 1, which lead to increasingly refined algorithms, whose performances can be compared in terms of the coefficient $\alpha$ introduced in (4). An outline is provided in Table 1, up to universal multiplicative constants. Though each algorithm is discussed in detail later in the article, let us introduce them here.

The most naive one, the *Exhaustive Search* Algorithm 3, consists in fully identifying each sample $Y_j$ through binary search. At any time $t$, it outputs the most frequent class among the identified samples as an estimation of the true mode. For a tolerated probability of error $\delta \in (0,1]$ and a class $y_i \in \mathcal{Y}$, a number of samples proportional to $\Delta_i^{-2} \ln(1/\delta)$ is needed to correctly identify $y_1$ as the mode among $\{y_1, y_i\}$, where

$$\Delta_i^2 := -\ln\Big(1 - \big(\sqrt{p(y_1)} - \sqrt{p(y_i)}\big)^2\Big). \tag{5}$$

Unsurprisingly, $\Delta_i^{-2}$ is increasing with $p(y_i) \in [0, p(y_1))$ and grows infinitely large when $p(y_i)$ get closer to $p(y_1)$. The probability of error of Algorithm 3 is dominated by the probability of mistakenly picking the second most likely candidate $y_2$ as the mode, leading to the asymptotic performance $\alpha = \Delta_2^{-2} \lceil \log_2(m) \rceil$, as detailed in Section 3. It can be improved by identifying new samples with an entropy-based dichotomic search that uses a learned empirical distribution $\hat{p}$ on $\mathcal{Y}$, which yields Algorithm 5. Asymptotically, it replaces the $\log_2(m)$ queries per sample of Algorithm 3 by an expected number of queries equal to the entropy $H(p) := \sum_{i \geq 1} p(y_i) \, |\log_2(p(y_i))| \leq \log_2(m)$ of $p$, plus one query per sample due to some boundary effects, as reflected in Table 1 and as explained in Subsection 3.3. Showing how *online learning with partial feedback benefits from entropy coding* is our second contribution.

There are two disjoint ways to further improve Algorithm 5. The first one exploits the main characteristic of our problem: the possibility of asking for partial information on samples at a lower cost than for complete identification. In particular, when searching for the mode by identifying samples with entropy-based techniques using Huffman trees, one can stop the identification procedure at roughly the depth of the mode in the tree, as classes that are deeper in the tree are unlikely candidates. This idea leads to the design of the *Truncated Search* Algorithm 7, detailed in Section 4. It reduces the asymptotic number of queries per sample from a constant plus the average depth of leaves in a Huffman tree $H(p)$ to a constant plus the minimal depth $|\log_2(p(y_1))|$, as seen in Table 1. The second amelioration comes from the adaptation of bandit algorithms to our setting. We develop in Section 5 the *Elimination* Algorithm 8, that discards mode candidates as soon as they seem unlikely to be the true mode. The class $y_i \in \mathcal{Y}$ can be eliminated after roughly $\Delta_i^{-2} \ln(1/\delta)$ samples, leading to an asymptotic performance $\alpha \simeq \Delta_2^{-2} + \sum_{i \geq 1} \Delta_i^{-2} \, p(y_i) \, |\log_2(p(y_i))|$, with the abuse of notation $\Delta_1^{-2} := \Delta_2^{-2}$.

While Algorithms 7 and 8 are both improvements over the Adaptive Exhaustive Search Algorithm 5, neither is strictly better than the other. When $p(y_1) - p(y_2)$ is very small compared to the other gaps $p(y_1) - p(y_i)$, the advantage goes to the Elimination Algorithm,[2] while it goes to the Truncated Search Algorithm when there are many $y \in \mathcal{Y}$ with small mass $p(y) \ll p(y_1)$ and $p(y_1) - p(y_2)$ is not too small.[3] We combine the ideas of each algorithm and get the best of both worlds with the *Set Elimination* Algorithm 9, presented in Section 6, which can be understood either as grouping together low mass classes before applying an elimination procedure to the resulting partitions, or as refining the truncated search procedure by taking into account confidence intervals. This is reflected in the corresponding coefficient $\alpha = \Delta_2^{-2} + \sum_{i \geq 1} \Delta_i^{-2} \, p(y_i) \, |\log_2(p(y_1))|$, where the number of samples $\Delta_i^{-2}$ is as for the Elimination Algorithm, while the expected number of queries needed for each sample $1 + |\log_2(p(y_1))|$ comes from the Truncated Search Algorithm, resulting in the best asymptotic performance among the proposed algorithms. *This sophisticated solution to Problem 1, together with its implementation available online, is our final contribution.*

[2] Consider the distribution $p(y_1) = 2/m$, $p(y_2) = 2/m - 1/m^2$ and $p(y) = (1 - p(y_1) - p(y_2))/(m-2)$ for all other $y \in \mathcal{Y} = \{y_1, \dots, y_m\}$. Then $\alpha_E = \sum_{i \geq 1} \Delta_{i,*}^{-2} \, p(y_i) \, |\log_2(p(y_i))| = O(\alpha_{TS}/m)$ as $m \to \infty$.

[3] Consider the distribution $p(y_1) = 1/2$ and $p(y) = 1/(2(m-1))$ for all other $y \in \mathcal{Y} = \{y_1, \dots, y_m\}$. Then $\alpha_E \to \infty$ while $\alpha_{TS} = \sum_{i \geq 1} \Delta_2^{-2} \, p(y_i) \, |\log_2(p(y_1))| = O(1)$ as $m \to \infty$.

In conclusion, our main contributions are summarized as follows.

1. Introducing the problem of mode estimation with partial feedback, Problem 1.
2. Unveiling links between adaptive entropy coding and active learning.
3. Combining adaptive entropy coding, coarse search procedures and bandits-inspired principles into Algorithm 9, an intuitive yet efficient solution to Problem 1.

Last but not least, we provide a code base to help researchers advance our knowledge of weakly supervised online learning, available at www.github.com/VivienCabannes/mepf.

**2. The Empirical Mode Estimator.** In the first part of this article, we will design mode estimation algorithms that *fully identify each sample* $Y_j$ one after the other. Given the identification of $n$ samples, those algorithms *estimate the mode of* $p$ *as the empirical mode among the* $n$ *samples,*

$$\hat{y}_n := \arg\max_{y\in\mathcal{Y}} \sum_{j\in[n]} \mathbf{1}_{Y_j=y}, \tag{6}$$

with ties broken arbitrarily. A tight characterization of the performance of the estimator (6) is offered by the following theorem.

THEOREM 1 (Empirical mode performance). *Let* $(Y_j)_{j\in[n]}$ *be* $n$ *independent variables sampled according to* $p \in P(\mathcal{Y})$*. Then the probability of error of* (6) *satisfies*

$$\exp\left(-n\Delta_2^2 - m\ln(n+1) - c_p\right) \le \mathbb{P}(\hat{y}_n \neq y_1) \le \exp\left(-n\Delta_2^2\right). \tag{7}$$

*where* $\Delta_2$ *is defined in Eq.* (5)*, and* $c_p$ *is some constant that depends on* $p$*.*

When accessing $n$ samples, the performance of empirical mode $\hat{y}_n$ cannot be bested without additional information on $p$, which leads to the following corollary.

COROLLARY 2 (Minimax lower bound). *For any distribution* $p_0 \in P(\mathcal{Y})$*, and any algorithm* $\mathcal{A}$ *that predicts* $\hat{y} := \mathcal{A}((Y_j)_{j\in[n]})$ *based on* $n$ *observations* $(Y_j)$*, there exists a permutation* $\sigma \in \mathfrak{S}_m$ *such that, when the data are generated according to* $p \in P(\mathcal{Y})$ *defined through the formula* $p(y) = p_0(\sigma(y))$*, the lower bound* (7) *holds for this algorithm.*

As one needs at least a single query per sample to gain any meaningful information, Corollary 2 states that the number of queries $T$ need to be greater than $\Delta_2^{-2}\ln(1/\delta)$, up to higher order terms, to reach precision $\delta$ as defined by Equation (3). The main challenge is thus to get as close to this lax lower bound as possible.

2.1. *Proof of Theorem 1.* In this subsection, we prove Theorem 1. We follow a proof of Sanov's theorem due to Csiszár and Körner (2011), together with an explicit computation of an information projection. Let us partition all possible sequences of observations $(Y_j)_{j\in[n]}$ according to their empirical distribution defined as

$$\hat{p}_{(Y_j)}(y) := n^{-1} \sum_{j\in[n]} \mathbf{1}_{Y_j=y}.$$

To do so, we define for any such empirical distribution $q \in P(\mathcal{Y}) \cap n^{-1}\cdot\mathbb{N}^{\mathcal{Y}}$ the type class

$$\mathcal{T}(q) = \left\{(y_t) \in \mathcal{Y}^n \,\middle|\, \forall\, y \in \mathcal{Y};\quad \hat{p}_{(y_t)}(y) = q(y)\right\}.$$

The event $\{\hat{y}_n \neq y_1\}$ is the union over all the values that $\hat{p}$ can take of the events "$\hat{p}$ does not have the right mode", which we can write as a union of disjoint events according to the type of $(Y_j)$, which leads to the bound

$$\sum_{q \in \mathcal{Q}_{n,-}} \mathbb{P}_{(Y_j)\sim p}\big((Y_j) \in \mathcal{T}(q)\big) \leq \mathbb{P}_{(Y_j)_{j\in[n]}}(\hat{y} \neq y_1) \leq \sum_{q \in \mathcal{Q}_{n,+}} \mathbb{P}_{(Y_j)\sim p}\big((Y_j) \in \mathcal{T}(q)\big),$$

where we account for the cases where $\hat{p}$ has several modes by differentiating

$$\mathcal{Q}_{n,-} = \left\{q \in P(\mathcal{Y}) \cap n^{-1} \cdot \mathbb{N}^{\mathcal{Y}} \,\middle|\, y_1 \notin \arg\max q(y)\right\}, \tag{8}$$

and

$$\mathcal{Q}_{n,+} = \left\{q \in P(\mathcal{Y}) \cap n^{-1} \cdot \mathbb{N}^{\mathcal{Y}} \,\middle|\, \arg\max q(y) \neq \arg\max p(y)\right\}. \tag{9}$$

We would like to enumerate the different classes in the previous sums, as well as their probability. A few lines of derivations lead to the equality,

$$\mathbb{P}((Y_j) = (z_j)) = 2^{-n(H(\hat{p}_{(z_j)}) + D(\hat{p}_{(z_j)} \| p))}.$$

Here, $D$ is the Kullback-Leibler divergence, and $H$ the entropy

$$D(q\|p) = \mathbb{E}_{Y\sim q}[\log_2(\frac{q(Y)}{p(Y)})], \qquad H(q) = \mathbb{E}_{Y\sim q}[-\log_2(q(Y))].$$

As a consequence, using the exchangeability of the $(Y_j)$,

$$\mathbb{P}((Y_j) \in \mathcal{T}(q)) = \mathbb{P}(\hat{p}_{(Y_j)} = q) = |\mathcal{T}(q)|\, 2^{-n(H(q)+D(q\|p))}.$$

We are left with the computation of the cardinality of each $\mathcal{T}(q)$. This is nothing but

$$|\mathcal{T}(q)| = \binom{n}{(nq(y))_{y\in\mathcal{Y}}} = \frac{n!}{\prod_{y\in\mathcal{Y}}(nq(y))!}.$$

This cardinality can be bounded with probabilistic arguments, as shown in Theorem 11.1.3 (which is 12.1.3 in the 2nd edition) of Cover and Thomas (1991),[4]

$$(n+1)^{-m} 2^{nH(q)} \leq |\mathcal{T}(q)| \leq 2^{nH(q)}.$$

Collecting the different pieces so far, we reach the conclusion,

$$(n+1)^{-m} \sum_{q \in \mathcal{Q}_{n,-}} 2^{-nD(q\|p)} \leq \mathbb{P}(\hat{y} \neq y_1) \leq \sum_{q \in \mathcal{Q}_{n,+}} 2^{-nD(q\|p)}.$$

From there, using the fact that the cardinality of $\mathcal{Q}_{n,+}$ is at most $(n+1)^m$, we get the rough bound

$$(n+1)^{-m} \max_{q \in \mathcal{Q}_{n,-}} 2^{-nD(q\|p)} \leq \mathbb{P}(\hat{y} \neq y_1) \leq (n+1)^{m} \max_{q \in \mathcal{Q}_{n,+}} 2^{-nD(q\|p)}. \tag{10}$$

It is useful to define the "limit" $\mathcal{Q} = \{q \in P(\mathcal{Y}) \,|\, \arg\max q(y) \neq \arg\max p(y)\}$ of the sets $\mathcal{Q}_{n,-}$ and $\mathcal{Q}_{n,+}$. We are left with the computation of the so-called "information projections" $\min_{q\in\mathcal{Q}_{n,-}} D(q\|p)$ and $\min_{q\in\mathcal{Q}_{n,+}} D(q\|p)$.

[4] Slightly tighter bounds can be derived from the fact that, as proven by Robbins (1955),

$$k! = \sqrt{2\pi} k^{k+1/2} e^{-k+r_k}, \quad \text{with} \quad 0 \leq \frac{1}{12k+1} \leq r_k \leq \frac{1}{12k} \leq 1.$$

LEMMA 1. *For any distribution $p \in P(\mathcal{Y})$, there exists a constant $c_p$ such that for any $n \in \mathbb{N}$ with $\mathcal{Q}_{n,-}$ and $\mathcal{Q}_{n,+}$ defined by Equations (8) and (9),*

$$\frac{\Delta_2^2}{\ln(2)} + \frac{c_p}{n\ln(2)} \geq \min_{q \in \mathcal{Q}_{n,-}} D(q\|p) \geq \min_{q \in \mathcal{Q}_{n,+}} D(q\|p) \geq \min_{q \in \mathcal{Q}} D(q\|p) = \frac{\Delta_2^2}{\ln(2)}.$$

This lemma is proved in Appendix A.1. Combining it with Equation (10), we find that

$$\exp\left(-n\Delta_2^2 - c_p - m\ln(n+1)\right) \leq \mathbb{P}(\hat{y} \neq y_1) \leq \exp\left(-n\Delta_2^2 + m\ln(n+1)\right).$$

Finally, we note that an argument of Dinwoodie (1992) shows that $\ln(\mathbb{P}(\hat{y}_n \neq y))/n$ is always below its limit proved by Sanov (1957), which yields the stronger bound

$$\mathbb{P}(\hat{y} \neq y_1) \leq \exp(-n\Delta_2^2)$$

which we reported in Theorem 1.

2.2. *Minimax Optimality.* This subsection proves Corollary 2. To turn Theorem 1 into the minimax Corollary 2, we can use a prior on $p$ such that the optimal Bayesian algorithm is the mode estimation algorithm. To do so, let us endow the simplex $P(\mathcal{Y})$ with a distribution $D$ and try to minimize

$$\mathcal{E}(\mathcal{A}) = \mathbb{E}_{p\sim D}\left[\mathbb{E}_{(Y_j)}\left[\mathbf{1}_{\mathcal{A}((Y_j))\neq y_p^*} \,\middle|\, p\right]\right],$$

where $\mathcal{A}((Y_j))$ is the mode predicted by an algorithm $\mathcal{A}$ upon observing the independent samples $(Y_j)$ generated according to $p \in P(\mathcal{Y})$, and $y_p^*$ is the mode of $p$. To find the optimal algorithm $\mathcal{A}^*$, we can invert the expectation as

$$\mathcal{E}(\mathcal{A}) = \mathbb{E}_{(Y_j)}\left[\mathbb{E}_{p\sim D}\left[\mathbf{1}_{\mathcal{A}((Y_j))\neq y_p^*} \,\middle|\, (Y_j)\right]\right].$$

This leads to the optimal algorithm

$$\mathcal{A}^*((Y_j)) = \arg\min_{y\in\mathcal{Y}} \mathbb{E}_p\left[\mathbf{1}_{y\neq y_p^*} \,\middle|\, (Y_j)\right] = \arg\max_{y\in\mathcal{Y}} \mathbb{P}_{p\sim D}\left(y_p^* = y \,\middle|\, (Y_j)\right).$$

It follows from $\mathcal{E}(\mathcal{A}) \geq \mathcal{E}(\mathcal{A}^*)$ that there exists at least one distribution $p$ in the support of $D$ such that the error made by any algorithm $\mathcal{A}$ on this distribution cannot be better that the one made for $\mathcal{A}^*$, i.e.,

$$\mathbb{E}_{(Y_j)}\left[\mathbf{1}_{\mathcal{A}((Y_j))\neq y_p^*} \,\middle|\, p\right] \geq \mathbb{E}_{(Y_j)}\left[\mathbf{1}_{\mathcal{A}^*((Y_j))\neq y_p^*} \,\middle|\, p\right].$$

One can define a prior such that $\mathcal{A}^*$ corresponds to the empirical mode, i.e., $\mathcal{A}^*((Y_j)) = \hat{y}_n$. In particular, Corollary 2 follows from taking $D$ as the uniform distribution over the set of permutations of the given distribution $p_0$, i.e., $\mathfrak{S}_m \cdot p_0 = \{p \,|\, \exists\, \sigma \in \mathfrak{S}_m, p = \sigma_{\#} p_0\}$. Since any algorithm has to have its expected performance over the distribution $p$ bounded by the one of $\mathcal{A}^*$, for any algorithm, there exists at least one distribution $p \in \mathfrak{S}_m \cdot p_0$ such that its performance is no better than the one of $\mathcal{A}^*$.

*Aside on User-Knowable Bounds.* For a user that does not know $\Delta_y$, the bound of Theorem 1 is not practically helpful, and the practitioner might be interested in stronger formal guarantees. In particular, Valiant (1984) introduced the notion of $(1-\delta)$-probably $\epsilon$-approximately correct estimator, for some $\epsilon, \delta > 0$, which reads $\mathbb{P}(p(\hat{y}) < p(y_1) - \epsilon) \leq \delta$. Proofs based on concentration inequalities can usually be reworked to derive such $(\epsilon, \delta)$-PAC bounds by replacing some quantity $\Delta$, related to differences $p(y) - p(y_1)$, by $\epsilon$ in the bound on the error $\delta$. This is notably the case for Theorem 1. One can also estimate $\hat{\Delta}_y$ and use plug-in techniques. The second half of this paper will provide algorithms, namely the Elimination and Set Elimination Algorithms, such that the user knows, and furthermore can choose, the probability $\delta$ with which the algorithm will fail to output the true mode –in the bandit literature, such algorithms are called $(0, \delta)$-PAC.

**3. Exhaustive Dichotomic Search Procedures.** This section introduces baseline algorithms based on exhaustive search procedures. To leverage the empirical mode estimator, a naive baseline consists in fully identifying each sample $Y_j$ one after the other. Using a binary search, this can be done with $\lceil \log_2(m) \rceil$ queries (2) per sample. We can thus fully identify $n$ samples with $T = n \lceil \log_2(m) \rceil$ queries. Together with Theorem 1, we deduce that this naive baseline gets to precision $\delta$ as defined by Equation (3) with a number of queries bounded by $T = \Delta_2^{-2} \lceil \log_2(m) \rceil \ln(1/\delta)$. Our first proposed improvement over this baseline is to refine binary searches through entropy codes in order to *minimize the average number of binary questions asked to fully identify each sample* $Y_j$, reducing the number of queries per sample from $\lceil \log_2(m) \rceil$ to a quantity close to the entropy $H(p)$ of $p$ (Shannon, 1948).

3.1. *Entropy Coding.* In order to identify each sample $Y_j$ with the least amount of queries, we will use binary search algorithms stemming from coding theory. This section fixes terminology and adds self-contained details on the matter.

DEFINITION 1 (Binary Tree *etc.*). For a set of elements $\mathcal{Y}$, we define leaves as abstract objects $V_y$ for all $y \in \mathcal{Y}$. From leaves, we define nodes $V$ as abstract objects associated with a right child $V_1 = r(V)$ and a left child $V_2 = l(V)$ which are either nodes or leaves. A vertex is an abstract object $V$ which is either a node or a leaf. A vertex $V_1$ is a descendant of $V_2$, denoted by $V_1 \lhd V_2$, if it can be built from the composition $V_1 = s_1 \circ s_2 \circ s_3 \circ \cdots \circ s_d(V_2)$ for $d \in \mathbb{N}$ and $s_i \in \{r, l\}$ The descendants of a node $V$ are all the vertices that can be built from the composition $V' \lhd V$. A binary tree $\mathcal{T} = \mathcal{T}(R)$ is defined from a root node $R$ with a finite number of descendants forming a collection of vertices, such that each $V$ in this collection of vertices is defined by a unique path $V = s_1 \circ s_2 \circ s_3 \circ \cdots \circ s_d(R)$, with $d = D_{\mathcal{T}}(V) \in \mathbb{N}$ being known as the depth of $V$ in $\mathcal{T}$.

A binary tree is associated with a prefix code.

DEFINITION 2 (Vertex code). The code $c_{\mathcal{T}}(V) \in \{0,1\}^{D_{\mathcal{T}}(V)}$ of vertex $V$ in $\mathcal{T}$ is defined as the unique path to go from the root node of $\mathcal{T}$ to $V$ by reading $c_{\mathcal{T}}(V)$ and recursively advancing to the right child if reading a 1 and to the left child if reading a 0.

The concept of binary tree is graphically intuitive. Let us provide an example of a tree, and annotate each node in the tree with its code as per Definition 2. In the following illustration, each left (resp. right) child of a node are represented below the node on the left (resp. on the right). We emphasize the separation between right and left with a vertical bar. For example, if $V = r \circ l \circ l \circ r(R)$, then $c_{\mathcal{T}}(V) = 1001$.

```
    Root Node
Leaf: 0 |                                  Node: 1
        |                      Node: 10          | Leaf: 11
        |       Node: 100           | Leaf: 101 |
        | Leaf: 1000 | Leaf: 1001 |              |
```

Prefix codes have been heavily studied in information theory, where the goal is to transform a set $\mathcal{Y}$ in order to describe its elements $y$ with a sequence of bits $c(y)$. In particular, the minimal number of bits transmitted when encoding sequences of tokens $(Y_j) \in \mathcal{Y}^n$ into the concatenation of the codes $(c(Y_j))_{j \in [n]}$ is linked with the entropy of the empirical distribution of the $y \in \mathcal{Y}$ in $(Y_j)$. There is a well known fundamental limit, for any $p \in \mathcal{Y}$,

$$\min_{\mathcal{T}} \mathbb{E}_{Y \sim p}[D_{\mathcal{T}}(Y)] \geq H(p) := \mathbb{E}_{Y \sim p}[-\log_2 p(Y)],$$

where $D_{\mathcal{T}}(y)$ is seen as the length of the code $c(y)$, the expectation is seen as the average code length, and the minimization as the search for the best coding algorithm to minimize message length.

A simple algorithm to obtain optimal codes was found by Huffman (1952) –we reproduce it in Algorithm 1. It takes as input a set of elements $\mathcal{Y}$ such that some positive value $v(y)$ is associated to each $y \in \mathcal{Y}$. It outputs a tree whose leaves are the elements of $\mathcal{Y}$ and whose vertices also come associated to values, and which satisfies certain conditions with respect to those values. When the value $v(y)$ of element $y \in \mathcal{Y}$ is equal to $p(y)$ for some distribution $p \in \mathbb{P}(\mathcal{Y})$, which is assumed in Algorithm 1, then the prefix code of the resulting tree on the elements of $\mathcal{Y}$ is optimal with respect to the distribution $p$, i.e. it minimizes $\min_{\mathcal{T}} \mathbb{E}_{Y\sim p}[D_{\mathcal{T}}(Y)]$ (see Huffman, 1952).

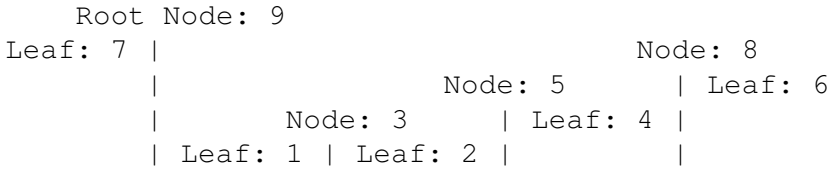
To describe Algorithm 1 succinctly, we introduce two notions of vertex orderings. The first comes from the value $v(y)$ associated by hypothesis to each $y \in \mathcal{Y}$, which would typically be $p(y)$, $N(y)$ or $N(y)/n$, respectively the probability of $y$ for some distribution $p$, the empirical occurrence counts of $y$ for some set of samples, or its empirical probability. Given a tree whose leaves $V_y$ are in bijection with the elements of $\mathcal{Y}$, let us associate to $V_y$ the value of the corresponding $y \in \mathcal{Y}$, i.e. $v(V_y) = v(y)$. The value of a node is then defined recursively as the sum of its children value $v(V) = v(r(V)) + v(l(V))$. This leads to the following notion of ordering.

DEFINITION 3 (Node ordering). Two vertices $V_1$ and $V_2$ satisfy the partial ordering $V_1 \vdash V_2$ if $v(V_1) < v(V_2)$ or if $v(V_1) = v(V_2)$ and $V_2$ is a node while $V_1$ is a leaf. The two vertices are equivalent, which we write $V_1 \sim_{\vdash} V_2$, if $V_1 \nvdash V_2$ and $V_2 \nvdash V_1$.

The second ordering orders the nodes from bottom to top, and left to right.

DEFINITION 4 (Code ordering). Two vertices $V_1$ and $V_2$ in a tree $\mathcal{T}$ satisfy the total ordering $V_1 <_{\mathcal{T}} V_2$ if $D_{\mathcal{T}}(V_1) > D_{\mathcal{T}}(V_2)$ or if $D_{\mathcal{T}}(V_1) = D_{\mathcal{T}}(V_2)$ and $c_{\mathcal{T}}(V_1) < c_{\mathcal{T}}(V_2)$ with respect to the lexicographical order.

Once again, this is a highly visual concept. Let us number the vertices of the previous tree according to the total ordering from Definition 4.




**Data:** Set of elements $\mathcal{Y}$ endowed with a probability distribution $p \in P(\mathcal{Y})$.
Create vertices $V_y$ for each $y \in \mathcal{Y}$ with value $v(V_y) = p(y)$;
Sort all the node into a heap $\mathcal{S}$ according to the comparison $\vdash$ (Definition 3);
**while** $\mathcal{S}$ *has more than one element* **do**
  Pop $V_1, V_2$ the respective smallest elements in $\mathcal{S}$;
  Merge them into a parent node $V$ with $V_1$ as the left child and $V_2$ the right one;
  Insert $V$ into the heap $\mathcal{S}$ with its value $v(V) = v(V_1) + v(V_2)$;
**end**
Set the remaining node $V$ in the heap $\mathcal{S}$ as the root node of $\mathcal{T} = \mathcal{T}(V)$;
**Result:** Huffman tree $\mathcal{T}$

**Algorithm 1:** Huffman Scheme

Let us illustrate Algorithm 1 with the count vector $N = (69, 14, 8, 6, 3)$. At the first iteration, we set all nodes in a heap. Let us represent this heap as a sorted list with comma-separated elements.

```
Leaf 3; Leaf 6; Leaf 8; Leaf 14; Leaf 69;
```

Then we merge the two smallest ones and add the result in the heap. Let us picture descendants below the nodes.

```
Leaf 8; Node 9; Leaf 14; Leaf 69;
   Leaf 3 | Leaf 6
```

We iterate the process.

```
Leaf 14; Node 17; Leaf 69
    Leaf 8 | Node 9
        Leaf 3 | Leaf 6
```

Now the heap is only made of a node with value 31 and a leaf with value 69.

```
    Node 31; Leaf 69;
Leaf 14 | Node 17
     Leaf 8 | Node 9
         Leaf 3 | Leaf 6
```

We end up with a binary tree.

```
                                        Node: 100
       Node: 31                           | Leaf: 69
Leaf: 14 |       Node: 17                 |
         | Leaf: 8 |       Node: 9        |
         |         | Leaf: 3 | Leaf: 6 |
```

In this paper, we introduce algorithms that build and adapt Huffman trees with respect to a distribution $p$ that is simultaneously being learnt online. This requires updating the trees on the fly. We can do so with the rebalancing algorithm of Vitter (1987), which we present in Algorithm 2. This algorithm uses integer values (which corresponds to empirical counts), and updates the tree in reaction to an increase in value of $+1$ for a single leaf. Of course, it can be repeatedly applied to account for greater changes.

**Data:** A Huffman tree $\mathcal{T}$ with counts $v(V) \in \mathbb{N}$. A node $V$.
Swap $V$ with the biggest $V_1$ in the sense of $<_{\mathcal{T}}$ such that $V \sim_{\vdash} V_1$;
Update $v(V) := v(V) + 1$;
Swap $V$ with the smallest $V_2$ in the sense of $<_{\mathcal{T}}$ such that $V \vdash V_2$;
**if** $v(V) = v(V_2)$ **then**
  Update the new parent of $V_2$ by calling this algorithm on $(\mathcal{T}, V_2)$;
**else**
  Update the new parent of $V$ by calling this algorithm on $(\mathcal{T}, V)$;
**end**
**Result:** Updated Huffman tree $\mathcal{T}$.

**Algorithm 2:** Vitter Rebalancing

One can prove the following properties of the Huffman tree construction and rebalancing.

PROPOSITION 2 (Vitter (1987)). *When $\mathcal{T}$ is built with Algorithm 1 with some initial value $v(V_y) = \sum_{j \in [t_0]} \mathbf{1}_{Y_j = y} > 0$ (for some set of samples $(Y_j)_j \subset \mathcal{Y}$) and is updated incrementally at each timestep $t \in \mathbb{N}$ with respect to Algorithm 2 and the observation of new*

*samples $Y_t \in \mathcal{Y}$, the total ordering $<_{\mathcal{T}}$ (Definition 4) is always compatible with the partial ordering $\vdash$ (Definition 3).*

Note that some subtleties are to be taken into account at initialization to deal with leaves that have not yet been observed. In coherence with the construction of Vitter (1987), we consider a special "not yet observed" node which defines a balanced subtree containing all the unobserved leaves as its descendants. When observing a new element $y$, we remove it from this subtree, re-balance the subtree, and create a new node whose left child is the "not yet observed" node and whose right one is the leaf corresponding to this $y$. This new node is set at the former place of the "not yet observed" node.

Finally, the following code property will be useful to derive statistical guarantees for our algorithms.

DEFINITION 5. A coding scheme $\mathcal{A}$ that associates a binary tree $\mathcal{T}_{\mathcal{A}}(p)$ to a probability vector $p \in P(\mathcal{Y})$ is said to be $C$-balanced if $|c_{\mathcal{T}_{\mathcal{A}}(p)}(y)| \leq C \lceil \log_2(p(y)) \rceil$ for all $y \in \mathcal{Y}$.

Shannon codes are built explicitly to be 1-balanced (Shannon, 1948). However, in contrast with Huffman coding, Shannon coding does not provide the lowest expected. Huffman codes are not always 1-balanced, but the following lemma, which we prove in Appendix A.2, states that they are at least 2-balanced –in fact, one can prove a slightly better constant than 2.

LEMMA 3. *Let $\mathcal{T}$ be a Huffman tree with respect to a value function $v$ on its vertices such that $v(R) = 1$, where $R$ is the root of $\mathcal{T}$. Then for any vertex $V$ of $\mathcal{T}$, we have*

$$D_{\mathcal{T}}(V) \leq 2 \lceil \log_2(1/v(V)) \rceil .$$

*In other words, Huffman codes are 2-balanced.*

3.2. *Exhaustive Search with Fixed Coding.* As mentioned at the start of the section, our first, and rather naive, proposed method of solving Problem 1 is to fully identify each sample $Y_i$ using a fixed search procedure, i.e., a set of $q$ predefined questions $(\mathbf{1}_{y \in S_i})_{i \in [q]}$ such that any element $y \in \mathcal{Y}$ is fully identified by the values of the functions $\mathbf{1}_{y \in S_i}$. Such a search procedure an be mapped to a prefix code $c : \mathcal{Y} \to \{0,1\}^q$ that associates any $y$ to the binary code $c(y) = (\mathbf{1}_{y \in S_i})_{i \in [q]}$. It can also be mapped to a binary tree, the code describing branching properties to reach the leaf $y$ from the root node. Each question can be seen as eliciting one bit in the code of $Y_j$, or equivalently going down one node in the corresponding tree. Reciprocally, a code $(c(y)_j)_{j \in [q]}$ is associated with the search procedure considering $S_j = \{y \in \mathcal{Y} \,|\, c(y)_j = 1\}$. In simple terms, $S_j$ enumerates all elements whose code $c$ contains a 1 in the $j$-th position. This strategy for solving Problem 1 is formalized in Algorithm 3. The average number of questions asked by such a search procedure reads $\sum_{y \in \mathcal{Y}} p(y)|c(y)|$ where $|c(y)|$ is the length of the code of $y$, or equivalently the depth of $y$ in the associated binary tree. Consequently, if the predefined search procedure is a simple binary search, the length of the code of any element is at most $\lceil \log_2(m) \rceil$, and Theorem 1 guarantees that we need at most $T = \ln(1/\delta)\alpha$ queries with $\alpha = \Delta_2^{-2} \lceil \log_2(m) \rceil$ to bound the probability of error by $\delta$.

3.3. *Exhaustive Search with Adaptive Coding.* Ideally, we would like to apply Algorithm 3 with a code $c$ that is optimal with respect to the distribution $p$. As $p$ is not known *a priori*, we need to learn the code on the fly. This leads to Algorithm 4, where a code is adapted iteratively to best reflect the current estimate $\hat{p}$ of $p$ based on past observations. As shown by the following theorem, the need to learn a code online does not impact too much the complexity of Algorithm 5 in comparison to that of Algorithm 3 with an optimal code.

**Data:** Set of classes $\mathcal{Y} = \{y_1, \dots, y_m\}$ endowed with $p \in P(\mathcal{Y})$. A code $c : \mathcal{Y} \to \{0,1\}^d$ (by default, we let $d = \lceil \log_2(m) \rceil$ and $c(y_i)$ be the binary representation of the number $i$ with zeros in front)
**for** $j \in [n]$ **do**
  Get new sample $Y_j \sim p$ and query $c_k(Y_j)$ for $k \in \{1, \dots, \text{length}(c(Y_j))\}$ until $Y_j$ is identified;
**end**
Set $\hat{y} = \arg\max_{y \in \mathcal{Y}} N(y)$, where $N(y) = \sum_{j \in [n]} \mathbf{1}_{Y_j = y}$;
**Result:** Estimated mode $\hat{y} = \arg\max \sum_{j \in [n]} \mathbf{1}_{Y_j = y}$ of $p$.

**Algorithm 3:** Fixed Coding Exhaustive Search

**Data:** Set of classes $\mathcal{Y} = \{y_1, \dots, y_m\}$ endowed with $p \in P(\mathcal{Y})$.
Initialize $N(y) = 0$ for all $y \in \mathcal{Y}$; A coding scheme $\mathcal{A}$, an initial tree $\mathcal{T}$;
**for** $j \in [n]$ **do**
  Get new sample $Y_j \sim p$ and identify it by querying the entries of $c_{\mathcal{T}}$;
  Update $N(Y_j) = N(Y_j) + 1$, and $\mathcal{T} = \mathcal{T}_{\mathcal{A}}(\hat{p})$ where $\hat{p}(y) = N(y)/j$;
**end**
Set $\hat{y} = \arg\max_{y \in \mathcal{Y}} N(y)$;
**Result:** Estimated mode $\hat{y} = \arg\max \sum_{j \in [n]} \mathbf{1}_{Y_j = y}$ of $p$.

**Algorithm 4:** Adaptive Coding Exhaustive Search

**Data:** Set of classes $\mathcal{Y} = \{y_1, \dots, y_m\}$ endowed with $p \in P(\mathcal{Y})$.
Initialize $N(y) = 0$ for all $y \in \mathcal{Y}$; Set $\mathcal{T}$ a Huffman tree with $V_y = N(y)$;
**for** $j \in [n]$ **do**
  Get new sample $Y_j \sim p$ and identify it by querying the entries of $c_{\mathcal{T}}(Y_j)$;
  Update $N(Y_j) = N(Y_j) + 1$, and the Huffman tree $\mathcal{T}$ with Algorithm 2;
**end**
Set $\hat{y} = \arg\max_{y \in \mathcal{Y}} N(y)$;
**Result:** Estimated mode $\hat{y} = \arg\max \sum_{j \in [n]} \mathbf{1}_{Y_j = y}$ of $p$.

**Algorithm 5:** Adaptive Coding Exhaustive Search with Huffman Coding

THEOREM 3 (Adaptive entropic search performance). *Given a $C$-balanced coding scheme (Definition 5), in order to fully identify $n$ samples $(Y_j)$ following the adaptive coding strategy of Algorithm 4, one needs on average $\mathbb{E}[T]$ queries, where*

$$nH(p) \le \mathbb{E}[T] \le Cn(H(p)+1) + 28Cm\log_2(n) + 6Cm + m^2, \tag{11}$$

*and $H(p) := -\sum_{y \in \mathcal{Y}} p(y)\log_2(p(y))$ is the entropy of the distribution $p \in P(\mathcal{Y})$. In particular, in the case of Huffman coding, Algorithm 5 yields*

$$nH(p) \le \mathbb{E}[T] \le 2n(H(p)+1) + o(n).$$

As shown by Theorem 3, to reach a probability of error smaller than $\delta$, Algorithm 4 needs $T = \ln(1/\delta)\alpha$ queries with $\alpha = C\Delta_2^{-2}(H(p)+1)$ up to higher-order terms, with $C = 1$ for Shannon coding and $C = 2$ for Huffman coding. This is an improvement in most cases over the coefficient $\alpha = \Delta_2^{-2}\lceil \log_2(m) \rceil$ of Algorithm 3. Nonetheless, both *Algorithms 3 and 4 precisely estimate $p(y)$ for each class $y$, which is more information than needed if one only wants the mode of $p$,* leaving room for improvements.

3.4. *Proofs.* This subsection is devoted to the proof of Theorem 3. It is well known from information theory that one has to ask at least $H(p)$ questions on average to be able to identify $Y \sim p$ (Shannon, 1948), which explains the lower bound. We need to prove the upper bound.

Recall that Algorithm 4 identifies each $Y_i$ by following a $C$-balanced tree with respect to the empirical distribution $\hat{p}_i$ of the previously observed samples $(Y_j)_{j\leq i} \subset \mathcal{Y}$ as values. Let us denote by $T_n$ the number of queries made to identify all the $(Y_i)$ for $i \leq n$. At round $i$, our search procedure associated with the probability distribution $\hat{p}_i$ identifies each element $y$ in $\mathcal{Y}$ such that $\hat{p}_i(y) \neq 0$ with at most $C\lceil -\log_2(\hat{p}_i(y))\rceil$ queries as per Definition 5. When $y$ is such that $\hat{p}_i(y) = 0$, we identify it with at most $m$ queries. Hence, at round $i+1 \in [n]$, $Y_{i+1} \sim p$ is identified with at most the following number of questions on average

$$C + \mathbb{E}_{y\sim p}\left[-\mathbf{1}_{\hat{p}_i(y)\neq 0} \cdot C\log_2(\hat{p}_i(y)) + \mathbf{1}_{\hat{p}_i(y)=0} \cdot m\right],$$

where $\hat{p}_i$ is our estimate of $p$ after the complete identification of $Y_i$, hence the one used for the queries on $Y_{i+1}$. Note that in this setting, for each $y \in \mathcal{Y}$, there is at most a single round where we need to identify $Y_i = y$ while $\hat{p}_{i-1}(y) = 0$. Recursively, this leads to

$$\mathbb{E}_{(Y_i)_{i\leq n}}[T_n] \leq Cn - C\sum_{i=1}^{n-1}\mathbb{E}_{(Y_j)}\left[\sum_{y\in\mathcal{Y}} p(y)\mathbf{1}_{\hat{p}_i(y)\neq 0} \cdot \log_2(\hat{p}_i(y))\right] + m^2.$$

Let us assume without loss of generality that $p(y)$ is never 0, as the terms for which $p(y) = 0$ do not contribute to the sum. Now we split this equation into two parts: a first part where one does not have a tight control of the empirical distribution, but that is really unlikely and will not contribute much to the full picture; and a part where the empirical distribution concentrates towards its real mean,

$$-\sum_{i=1}^{n-1}\mathbb{E}_{(Y_j)}\left[\sum_{y\in\mathcal{Y}} p(y)\mathbf{1}_{\hat{p}_i(y)\neq 0}\log_2(\hat{p}_i(y))\right] = -\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{E}_{(Y_j)}\left[\mathbf{1}_{N_{y,i}\neq 0}\log_2\left(\frac{N_{y,i}}{i}\right)\right]$$

$$= \sum_{i,y} p(y)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|\geq\frac{1}{2}\}}\mathbf{1}_{N_{y,i}\neq 0}\log_2\left(\frac{i}{N_{y,i}}\right) + 1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\log_2\left(\frac{i}{N_{y,i}}\right)\right],$$

where $N_{y,i} = \sum_{j\leq i}\mathbf{1}_{Yj=y}$ is the number of times we have seen $y$ in the first $i$ samples. The first term corresponds to a highly unlikely event, which we prove in Appendix A.2.

LEMMA 4. *With $N_{i,y}$ denoting the empirical count $\sum_{j\in[i]}\mathbf{1}_{Y_j=y}$,*

$$-\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|\geq\frac{1}{2}\}}\mathbf{1}_{N_{y,i}\geq 1}\log_2\left(\frac{N_{y,i}}{i}\right)\right] \leq 22m\log_2(n),$$

We now consider the second term. Let us extract the scaling in $H(p)$ inherent to entropy coding. To that end, we rewrite it as

$$-\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\left(\log_2(p(y)) + \log_2\left(\frac{N_{y,i}}{ip(y)}\right)\right)\right].$$

The first term presents the desired scaling

$$-\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\log_2(p(y))\right] \leq -\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\log_2(p(y)) \leq nH(p).$$

Finally, we deal with the rightmost logarithm, using the Taylor series of the logarithm to show concentration for the empirical mean of the logarithm. The details are provided in Appendix A.2.

LEMMA 5. *With $N_{i,y}$ denoting the empirical count $\sum_{j\in[i]} \mathbf{1}_{Y_j=y}$,*

$$-\ln(2)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\log_2\left(\frac{N_{y,i}}{ip(y)}\right)\right] \leq 4m(\ln(n)+1).$$

Collecting the different pieces together, we find the upper bound,

$$\begin{aligned}\mathbb{E}_{(Y_i)_{i\leq n}}[T_n] &\leq Cn - C\sum_{i=1}^{n-1}\mathbb{E}_{(Y_j)}\left[\sum_{y\in\mathcal{Y}} p(y)\mathbf{1}_{\hat{p}_i(y)\neq 0}\cdot\log_2(\hat{p}_i(y))\right] + m^2\\ &\leq Cn(1+H(p)) + m^2 + 22Cm\log_2(n) + \frac{4Cm}{\ln(2)}(\ln(n)+1) + m^2\\ &\leq Cn(1+H(p)) + 28Cm\log_2(n) + 6Cm + m^2.\end{aligned}$$

**4. Truncated Search.** In this section, we improve upon the previous search procedures using the following key observation. When estimating the empirical mode of a batch by identifying samples following a Huffman tree, one can stop the search procedure roughly when reaching the depth $D = |\log_2(p(y_1))|$ of the mode, resulting in about $|\log_2 p(y_1)|$ queries per sample on average, rather than $H(p)$ queries when trying to fully identify each sample.

4.1. *Coarse Sufficient Statistics.* We introduce the concept of admissible partitions, which provide *sufficient statistics for mode estimation that are weaker than the full empirical distribution $\hat{p}$,* as well as the concept of $\eta$-admissible partition, which will be useful to build statistically and computationally efficient algorithms. Recall that a partition $\mathcal{P}$ of $\mathcal{Y}$ is a subset of $2^{\mathcal{Y}}$, such that for any $\{S_1, S_2\} \subset \mathcal{P}$, $S_1 \cap S_2 = \emptyset$, and $\cup_{S\in\mathcal{P}} S = \mathcal{Y}$. Here and throughout the text, for $p \in P(\mathcal{Y})$ and $S \subset \mathcal{Y}$, we define $p(S) := \sum_{y\in S} p(y)$.

DEFINITION 6 (Admissible partitions). An *admissible partition* of $\mathcal{Y}$ with respect to $p \in P(\mathcal{Y})$ refers to a partition $\mathcal{P} \subset 2^{\mathcal{Y}}$ such that

$$\arg\max_{S\in\mathcal{P}} p(S) = \{y^*\}, \qquad \text{where} \qquad y^* = \arg\max_{y\in\mathcal{Y}} p(y). \tag{12}$$

For $\eta > 0$, an *$\eta$-admissible partition* of $\mathcal{Y}$ with respect to $p$ is similarly defined as an admissible partition $\mathcal{P} \subset 2^{\mathcal{Y}}$ for which all sets $S \in \mathcal{P}$ such that $p(S) \geq \eta$ are singletons, and at most a single set $S$ verifies $p(S) < \eta/2$. Formally,

$$|\{S \in \mathcal{P} \,|\, p(S) < \eta/2\}| \leq 1 \qquad \text{and} \qquad \forall S \in \mathcal{P}, \quad p(S) \geq \eta \quad \Rightarrow \quad |S| = 1. \tag{13}$$

The interest of admissible partitions comes from the fact that if $\mathcal{P}$ is an admissible partition with respect to both $p$ and $p' \in P(\mathcal{Y})$, then $p$ and $p'$ have the same modes. In particular, if $\mathcal{P}$ is an admissible partition for the empirical distribution $\hat{p}$ of some batch of samples, the set $S \in \mathcal{P}$ with the largest mass $\hat{p}(S)$ is a singleton containing the empirical mode of the batch. On the computational side, when $p$ is known, $\eta$-admissible partitions are easy to build, which contrasts with the NP-hardness of finding an admissible partition of minimal cardinality, or of finding a set $S$ that maximizes $p(S)$ under the constraint $p(S) \leq p(y^*)$.[5]

[5]The latter is equivalent to the knapsack problem (Mathews, 1896), while the partition problem (Korf, 1998) can be reduced to the former with the following construction. Consider a list $p_0$ of positive integers and include an element $p_*$ equal to half the sum of $p_0$ plus an infinitesimal quantity. Next, normalize all elements to transform the new list into a probability vector $p$ whose elements add up to one. Checking if $p_0$ can be partitioned into two lists $S_1$ and $S_2$ that sum to the same value is equivalent to determining whether there exists an admissible partition of $p$ of cardinality three.

4.2. *Adaptive Truncated Search.* Building upon Definition 6, Algorithm 6 efficiently constructs an $\eta$-admissible partition $\mathcal{P}$ of $\mathcal{Y}$ with respect to the empirical frequencies $\hat{p}$ of a batch of samples $(Y_j)_{j\in[n]}$. It uses a predefined binary tree $\mathcal{T}$, and takes two parameters $\gamma, \epsilon \in [0,1]$ that define $\eta = \gamma\hat{p}(\hat{y}) - \epsilon$, where $\hat{y}$ is the mode of $\hat{p}$. The algorithm starts with the trivial partition $\mathcal{P} = \{\mathcal{Y}\}$, and recursively refines it by splitting the set $S_*$ with the greatest empirical mass until $S_*$ is a singleton, which must then be equal to $\{\hat{y}\}$. It then keeps splitting non-singleton sets of mass strictly greater than $\gamma\hat{p}(\hat{y}) - \epsilon$ until there are no such sets left. The splitting is done using the tree $\mathcal{T}$ as follows: we identify each node $V$ in the tree with the set of all the elements that map to its descendent leaves $S(V) = \{y \in \mathcal{Y} \,|\, V_y \lhd V\}$. At each time step, the sets $S$ of the current partition correspond to nodes $(V_S)$ of $\mathcal{T}$, and the set $S_* = S(V_*)$ that has to be split is replaced in the partition by its two children $S(l(V_*)), S(r(V_2))$ in $\mathcal{T}$. This consumes $N(S_*) = \sum_{j\in[n]} \mathbf{1}_{Y_j \in S_*}$ queries to identify which sample belongs to $S_1$ and which to $S_2$. At the end, a Huffman scheme is applied to merge sets into an $\eta$-admissible partition, and re-balance the tree $\mathcal{T}$ so that all sets in the partition are at a similar depth, roughly equal to $\log_2(2/\eta)$, in the new tree. This re-balancing keeps the structure of the "sub-trees" below the nodes corresponding to the sets of the partition intact. In addition, the algorithm identifies which sample belongs to which set of $\mathcal{P}$, as well as the empirical mode $\hat{y}$.

**Data:** Set $\mathcal{Y} = \{y_1, \dots, y_m\}$, binary tree $\mathcal{T}$, $n$ samples $(Y_j)$, parameters $\gamma, \epsilon \in \mathbb{R}$.
Set $V_*$ the root of the tree $\mathcal{T}$, $S_* = \mathcal{Y}$, $N(S_*) = n$;
Set $\mathcal{S} = \{(V_* : n)\}$ built as a heap, $\mathcal{L} = \{\}$ an empty list, and $C = -\infty$;
For all $S$ and $V$, we denote $N(S) = \sum_{j\in[n]} \mathbf{1}_{Y_j \in S} \in \mathbb{N}$ and $S(V) = \{y \in \mathcal{Y} \,|\, V_y \lhd V\} \subset \mathcal{Y}$.
**while** *the heap $\mathcal{S}$ is non-empty and* $N(S_*) \geq C$ **do**
  Set $V_* = \arg\max_{V\in\mathcal{S}} N(S(V))$ by popping it out of the heap $\mathcal{S}$; set $S_* = S(V_*)$;
  **if** *$V_*$ is a leaf* **then**
    If it was the first encountered leaf, set $\{\hat{y}\} := S_*$, and refine $C := \gamma N(\{\hat{y}\}) - \epsilon n$;
    Add $V_*$ to the list $\mathcal{L}$;
  **else**
    Make $N(S_*)$ queries to get all the information on $(\mathbf{1}_{Y_j \in S(V)})_{j\in[n]}$ for $V \in \{l(V_*), r(V_*)\}$;
    Insert each child $V \in \{l(V_*), r(V_*)\}$ into the heap $\mathcal{S}$ with value $N(S(V))$;
  **end**
**end**
Add all the remaining elements of the heap to the list $\mathcal{L}$;
Apply Huffman's scheme, Algorithm 1, to the list $\mathcal{L}$ to rebalance the top of the tree $\mathcal{T}$;
**Result:** A tree $\mathcal{T}$ containing an $(\gamma\hat{p}(\hat{y}) - \epsilon)$-admissible partition for $\hat{p}$, and the empirical mode $\hat{y}$.

**Algorithm 6:** Batch Tree Rebalancing

This suggests a new way to tackle Problem 1: given a batch of samples, Algorithm 6 yields an admissible partition with respect to the empirical distribution $\hat{p}$, and in particular identifies its empirical mode $\hat{y}$, which is the best possible mode estimate. It does not need to fully identify each sample to do so, unlike Algorithms 3 and 5. The number of queries consumed by Algorithm 6 depends on the tree $\mathcal{T}$ that it takes as input: upon reaching the final partition $\mathcal{P}$, the algorithm will have required $D$ queries for each sample belonging to a given set $S \in \mathcal{P}$, where $D$ is the depth of the node associated with $S$ in the tree $\mathcal{T}$. To minimize the expected number of queries needed, $\mathcal{T}$ should be a partial Huffman tree with respect to the distribution $p$; this would result in $D \leq C\lceil -\log_2(2/p(y_1))\rceil$ for each $S \in \mathcal{P}$, with $C = 2$ as per Lemma 3. We do not have a priori access to the distribution $p$, but we can learn the structure of such a tree over several rounds, with a slack parameter $\epsilon_r$ to account for the gradually increasing precision of our estimates. This is Algorithm 7, whose guarantees are provided by Theorem 4.

**Data:** Set $\mathcal{Y} = \{y_1, \dots, y_m\}$, samples $(Y_s)$, scheduling $(n_r) \in \mathbb{N}^{\mathbb{N}}$ and $(\epsilon_r) \in \mathbb{R}^{\mathbb{N}}$.
Set $\mathcal{T}$ a binary search tree;
**for** $r \in \mathbb{N}$ **do**
  Get $n_r$ fresh samples $(Y_{j,r})_{j \in [n_r]}$ defining an empirical probability $\hat{p}_r$;
  Run Algorithm 6 with tree $\mathcal{T}$, $\gamma = 1$ and $\epsilon = \epsilon_r$ to obtain the empirical mode $\hat{y}_r$ and update $\mathcal{T}$;
**end**
**Result:** Return the last $\hat{y}_r$ as the estimated mode.

**Algorithm 7:** Truncated Search

THEOREM 4 (Truncated search performance). *For any $\delta > 0$, the number of queries $T$ needed by Algorithm 7 run with the schedulings $n_r := 2^r$ and $\epsilon_r := \frac{1}{4m}\left(\frac{2}{3}\right)^{\frac{r}{2}}$ to correctly identify the mode with probability at least $1 - \delta$ satisfies*

$$\mathbb{E}[T] \leq 8 \ln(1/\delta) \Delta_2^{-2} \left\lceil \log_2 \left( \frac{4}{p(y_1)} \right) \right\rceil + o(\ln(1/\delta)). \tag{14}$$

Algorithm 7 is a clear amelioration over the previous search algorithms: the probability of error remains the same as a function of the number of samples, while reducing the average number of queries per sample from $H(p)$ to roughly $|\log_2 p(y_1)|$, going from average code length to minimal code length. Nonetheless, there is still room for improvement. Algorithm 7 identifies the empirical mode of the entire batch with absolute certitude. This contrasts with the main takeaway from the bandit literature: *one should leverage confidence intervals to build statistically efficient algorithms so as to avoid spending queries solely to rule out highly unlikely events.* We explore this intuition in the next section. Note that replacing the Huffman code from Algorithm 6 by any $C$-balanced code would result in a modified bound

$$\mathbb{E}[T] \leq 4C \ln(1/\delta) \Delta_2^{-2} \left\lceil \log_2 \left( \frac{4}{p(y_1)} \right) \right\rceil + o(\ln(1/\delta)).$$

in Theorem 4 (in particular, using a Shannon code would yield $C = 1$).

4.3. *Proofs.* This subsection provides the key elements for the proof of Theorem 4. Our strategy is the following.

- Find a likely event $\mathcal{A}$ where the algorithm behaves as desired.
- Show that its complement ${}^c\mathcal{A}$ is unlikely enough that the additional queries it elicits are asymptotically negligible.

Let $r \geq 2$. We let $\mathcal{T}_r$ be the updated tree at the end of round $r$, and $\mathcal{P}_r$ be the corresponding $\eta_r$-admissible partition where $\eta_r = \hat{p}_r(\hat{y}_r) - \epsilon_r$. Here $\hat{p}_r$ is the empirical distribution of the samples at round $r$, and $\hat{y}_r$ is the corresponding empirical mode. Let us define the event

$$\mathcal{A} = \left\{ \forall S \subset \mathcal{Y}, \quad \max\left\{ |\hat{p}_{r-1}(S) - p(S)|, |\hat{p}_r(S) - p(S)| \right\} \leq \frac{\epsilon_{r-1} - \epsilon_r}{4} \right\}. \tag{15}$$

A union bound, detailed in Appendix A.3, shows that $\mathcal{A}$ is likely to happen as the number of rounds increases, as stated in the following lemma.

LEMMA 6. *The event $\mathcal{A}$ defined by (15) satisfies*

$$\mathbb{P}\left({}^c\mathcal{A}\right) \leq 2^{m+1} \exp\left( -\frac{n_{r-1}}{2} \left( \frac{\epsilon_{r-1} - \epsilon_r}{4} \right)^2 \right). \tag{16}$$

We claim that under the event $\mathcal{A}$, the $\eta_{r-1}$-admissible partition obtained at the end of round $r-1$ stays admissible at round $r$, namely, we have the implication

$$\mathcal{A} \quad \subset \quad \{\text{All sets } S \in \mathcal{P}_{r-1} \text{ such that } \hat{p}_r(S) > \hat{p}_r(\hat{y}_r) - \epsilon_r \text{ are singletons}\}\,. \tag{17}$$

PROOF OF EQUATION (17). Indeed, $\mathcal{A}$ implies that $|\hat{p}_{r-1}(S) - \hat{p}_r(S)| \leq (\epsilon_{r-1} - \epsilon_r)/2$ and as a consequence, for any set $S \in \mathcal{P}_{r-1}$ of cardinality at least 2, the fact that $\mathcal{P}_{r-1}$ is $(\hat{p}_{r-1}(\hat{y}_r) - \epsilon_{r-1})$-admissible with respect to $\hat{p}_{r-1}$ leads to

$$\begin{aligned}\hat{p}_r(S) &\leq \hat{p}_{r-1}(S) + \frac{\epsilon_{r-1} - \epsilon_r}{2} < \hat{p}_{r-1}(\hat{y}_{r-1}) - \epsilon_{r-1} + \frac{\epsilon_{r-1} - \epsilon_r}{2} \\ &\leq \hat{p}_r(\hat{y}_{r-1}) - \epsilon_r \leq \hat{p}_r(\hat{y}_r) - \epsilon_r,\end{aligned}$$

Hence, under $\mathcal{A}$, all sets $S \in \mathcal{P}_{r-1}$ such that $\hat{p}_r(S) > \hat{p}_r(\hat{S}_r) - \epsilon_r$ are singletons. □

Let us assume that $\mathcal{A}$ is realized –then all sets $S \in \mathcal{P}_{r-1}$ are either singletons or such that $\hat{p}_r(S) \leq \hat{p}_r(\hat{y}_r) - \epsilon_r$. Hence, when Algorithm 6 is applied to the $n_r$ samples of round $r$, a sample $Y_{i,r}$ belonging to some set $S \in \mathcal{P}_{r-1}$ only consumes a number of queries smaller[6] or equal to the depth $D(S)$ of $S$ in the tree $\mathcal{T}_{r-1}$. Since $\mathcal{T}_{r-1}$ was obtained by applying Huffman's scheme to the sets of $\mathcal{P}_{r-1}$ at the end of Algorithm 6 to re-balance the tree $\mathcal{T}_{r-2}$, we can leverage Lemma 3 to bound the depth $D(S)$. Since $\{y_1\} \subset \mathcal{Y}$, when $\mathcal{A}$ is realized

$$p(y_1) - \hat{p}_{r-1}(\hat{y}_{r-1}) \leq p(y_1) - \hat{p}_{r-1}(y_1) < \frac{\epsilon_{r-1} - \epsilon_r}{4} < \epsilon_r.$$

As $\mathcal{P}_{r-1}$ is $(\hat{p}_{r-1}(\hat{y}_{r-1}) - \epsilon_{r-1})$-admissible, necessarily each $S \in \mathcal{P}_{r-1}$ (except at most one) is such that

$$\hat{p}_{r-1}(S) \geq \frac{\hat{p}_{r-1}(\hat{y}_{r-1}) - \epsilon_{r-1}}{2} > \frac{p(y_1) - \epsilon_r - \epsilon_{r-1}}{2} \geq \frac{p(y_1) - 2\epsilon_{r-1}}{2} \geq \frac{p(y_1)}{4},$$

where we use the fact that $\epsilon_{r-1} < 1/4m \leq p(y_1)/4$. Applying Lemma 3, we deduce that

$$D(S) \leq 2\left\lceil \log_2(4/p(y_1)) \right\rceil. \tag{18}$$

If $S$ is such that $\hat{p}_{r-1}(S) < \frac{\hat{p}_{r-1}(\hat{y}_{r-1}) - \epsilon_{r-1}}{2}$ (and there can be only one such set), then it must share a parent in $\mathcal{T}_{r-1}$ with some other set $S'$, from which we deduce that its depth follows the same bound. Hence the total number of queries consumed at round $r$, assuming that $\mathcal{A}$ is realized, is at most $n_r 2\left\lceil \log_2(4/p(y_1)) \right\rceil$. If the event $\mathcal{A}$ is not realized, we can roughly upper bound the number of queries spent per sample by $m$, hence we can upper bound the total number of queries by $n_r m$. In the special case $r = 1$, we similarly need at most $n_r m$ queries. A few lines of derivations provided in Appendix A.3 lead to the following lemma.

LEMMA 7. *When running Algorithm 7 with the schedule of Theorem 4, the expected number of queries needed for round $r$ satisfies*

$$\mathbb{E}\left[T_r\right] \leq 2^r \left( \left\lceil \log_2\left(\frac{4}{p(y_1)}\right) \right\rceil + m 2^{m+1} \exp\left( -\left(\frac{4}{3}\right)^r \frac{C}{m^2} \right) \right)$$

*for some constant $C > 0$.*

[6] The number of queries needed can be strictly smaller than the depth of $S$ if some parent set $S'$ of $S$ is such that $\hat{p}_{r-1}(S') > \hat{p}_{r-1}(\hat{y}_{r-1}) - \epsilon_{r-1}$ but $\hat{p}_r(S') \leq \hat{p}_r(\hat{y}_r) - \epsilon_r$.

At the end of round $r$, the algorithm outputs the empirical mode $\hat{y}_r$ of the $n_r$ samples $(Y_{i,r})_{i\in[n_r]}$ defining the empirical probability $\hat{p}_r$. We know from Theorem 1 that its probability of error $\delta_r$ is bounded by

$$\delta_r \leq \exp\left(-n_r\Delta_2^2\right) = \exp\left(-2^r\Delta_2^2\right).$$

Now let $\delta > 0$, and define the round

$$r_\delta = \min\left\{r \,\middle|\, \exp(-n_r\Delta_2^2) \leq \delta\right\} = \min\left\{r \,\middle|\, n_r \geq \Delta_2^{-2}\ln(1/\delta)\right\}$$

By construction, $n_{r_\delta} = 2n_{r_\delta-1} \leq 2\ln(1/\delta)\Delta_2^{-2}$. At the end of round $r_\delta$, the probability of outputting the correct class is at least $1-\delta$, and the total number of expected queries used so far by Algorithm 7 is at most

$$\begin{aligned}
\sum_{r=1}^{r_\delta}\mathbb{E}[T_r] &\leq \sum_{r=1}^{r_\delta} n_r\left(2\left\lceil\log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + m2^{m+1}\exp\left(-\left(\frac{4}{3}\right)^r\frac{C}{m^2}\right)\right)\\
&\leq 2\left\lceil\log_2\left(\frac{4}{p(y_1)}\right)\right\rceil\sum_{r=1}^{r_\delta} n_r + \sum_{r\geq 1} 2^r m2^{m+1}\exp\left(-\left(\frac{4}{3}\right)^r\frac{C}{m^2}\right)\\
&\leq 4n_{r_\delta}\left\lceil\log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + \tilde{C} \leq 8\ln(1/\delta)\Delta_2^{-2}\left\lceil\log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + o(\ln(1/\delta)).
\end{aligned}$$

This completes the proof.

**5. Bandit-Inspired Elimination.** The core idea of this section is to successively eliminate candidate classes $y \in \mathcal{Y}$ in order to lower the required number of queries compared to the Exhaustive Search Algorithm from Section 3, which can be done by adapting and improving the seminal Successive Elimination Algorithm from Even-Dar, Mannor and Mansour (2006), resulting in Algorithm 8.

**Data:** Set $\mathcal{Y}$, probability $p \in P(\mathcal{Y})$, samples $\{Y_r\}_{r\in\mathbb{N}}$, schedule $\sigma : \mathbb{N}\times\mathbb{R}^{\mathcal{Y}} \to \mathbb{R}^{\mathcal{Y}}$.
Set $S_e = \emptyset$ the set of eliminated guesses, $r = 0$, $N(y) = 0 \in \mathbb{R}^m$;
**while** $\mathcal{Y}\backslash S_e$ *is not a singleton* **do**
  Set $r \leftarrow r+1$ and query $\mathbf{1}_{Y_r\in S_e}$;
  **if** $Y_r \notin S_e$ **then**
    Identify $Y_r$ with a Huffman tree adapted to empirical counts $N(y)$ on $\mathcal{Y}\setminus S_e$;
    Update $N(Y_r) = N(Y_r)+1$, and set the empirical distribution $\hat{p}_r(y) \propto N(y)$ accordingly;
    Set $S_e = S_e \cup \{y \in \mathcal{Y}\setminus S_e | \hat{p}_r(y) + \sigma_y(r,\hat{p}_r) < \max_{z\in\mathcal{Y}\setminus S_e}\hat{p}_r(z)\}$;
**end**
**Result:** Estimate mode $\hat{y}$ as the only element of $\mathcal{Y}\backslash S_e$.

**Algorithm 8:** Elimination

Algorithm 8 takes as input a generic parameter function $\sigma$ and works as follows: at any time, it maintains a set of eliminated classes $S_e \subset \mathcal{Y}$ that are deemed unlikely to be the mode. At the start of round $r$, the algorithm tests whether the new sample $Y_r$ belongs to $S_e$; if not, it is identified with a Huffman tree adapted to the empirical distribution $\hat{p}_r$ on $\mathcal{Y}\backslash S_e$, which is then updated.[7] Any class $y$ such that its empirical mass $\hat{p}_r(y)$ satisfies $p_r(y) +$

[7] There is a minor abuse of notations in the description of Algorithm 8: in the expression "$\sigma_y(r,\hat{p}_r)$", the distribution $\hat{p}_r$ on $\mathcal{Y}\backslash S_e$ is implicitly seen as a distribution on $\mathcal{Y}$ by setting $\hat{p}_r(y) = 0$ if $y \in S_e$ so that it is a valid argument for $\sigma$.

$\sigma_y(r, \hat{p}_r) < \max_{z \in \mathcal{Y} \setminus S_e} \hat{p}_r(z)$ is finally added to the set of eliminated classes $S_e$. For a well-chosen parameter function $\sigma$, Algorithm 8 has a high probability of identifying the true mode, as stated in Theorem 5.

THEOREM 5. *Let $\delta > 0$, and define the elimination scheduling $\sigma : \mathbb{N} \times \mathbb{R}^{\mathcal{Y}} \to \mathbb{R}^{\mathcal{Y}}$ as*

$$\sigma_y(r, \hat{p}) = \sqrt{\frac{c \max_z \hat{p}_r(z) \ln(\pi^2 m r^2 / \delta)}{r}}, \qquad \textit{with} \qquad c = 24. \tag{19}$$

*With probability $1 - \delta$, Algorithm 8 terminates, identifies the right mode and has consumed less than $T$ queries, where, conditionally to this successful event,*

$$\frac{\mathbb{E}[T]}{\ln(1/\delta)} \leq C_1 \frac{p(y_1)}{\nabla_2^2} + C_2 \sum_{i \in [m]} p(y_i) \frac{p(y_1)}{\nabla_i^2} \left\lceil \log_2 \left( \frac{1}{p(y_i)} \right) \right\rceil + o(1),$$

*for two universal constants $C_1, C_2$, $\nabla_i := p(y_1) - p(y_1)$ if $i \neq 1$ and $\nabla_1 = \nabla_2$.*

As shown in Proposition 8 below, whose proof can be found in Appendix A.4, $\nabla_i^{-2} p(y_1)$ is equal to $\Delta_i^{-2}$ up to a multiplicative factor as long as $p(y_1)$ is bounded away from 1 – this is no constraint in most interesting use cases, as Problem 1 gets easier the closer $p(y_1)$ gets to 1. Hence we see that Theorem 5 is a clear improvement upon the Exhaustive Search Algorithms 3 and 5. In essence, one requires roughly $\Delta_i^{-2} \log(1/\delta)$ samples to know with certainty $1 - \delta$ that class $y_i$ is not the mode. As the Exhaustive Search Algorithm from Section 3 never eliminates any class, the number of samples it requires to reach accuracy $1 - \delta$ is conditioned by the second most likely class $y_2$, the one that is hardest to correctly dismiss. Conversely, Algorithm 8 eliminates with high probability the class $y_i$ after having seen about $\log(1/\delta)\Delta_i^{-2}$ samples, which translates to an improved asymptotic expected number of required queries as a function of $\delta$. However, *Algorithm 8 does not leverage the coarse statistics* and search truncation mechanism presented in Section 4, leaving room for further improvement. We combine the elimination mechanism of Algorithm 8 and the search truncation technique of Algorithm 7 in Section 6, after the remainder of this section which is dedicated to the discussion of Algorithm 8 and the proof of Theorem 5.

PROPOSITION 8. *Let $p \in P(\mathcal{Y})$ with $p(y_1) > p(y_i)$ for all $i \in [m]$. We have the following inequality, with $\nabla_i := p(y_1) - p(y_i)$ and $\Delta_i^2 = -\ln(1 - (\sqrt{p(y_1)} - \sqrt{p(y_i)})^2)$,*

$$\frac{p(y_1)}{-\ln(1 - p(y_1))} p(y_1) \nabla_i^{-2} \leq \Delta_i^{-2} \leq 4 p(y_1) \nabla_i^{-2}.$$

*In particular, if $p(y_1)$ is bounded away from 1, then $-p(y_1)/\ln(1 - p(y_1)$ is bounded away from 0 and $\Delta_i^{-2} \simeq p(y_1) \nabla_i^{-2}$ up to a multiplicative constants.*

5.1. *Design of the elimination schedule.* Algorithm 8 was inspired by the seminal algorithm of Even-Dar, Mannor and Mansour (2006). Their Successive Elimination Algorithm uses Hoeffding's inequality to define $\sigma_y(r, p) = \sigma_r$ where $\sigma_r^2 \simeq \ln(1/\delta)/r$. It is known that Hoeffding's inequality can be refined for sub-gamma variables with Bernstein's inequalities, which was the main motivation behind the UCB-V algorithm of Audibert, Munos and Szepesvári (2009). Bernstein's inequality can be seen as a convenient weak formulation of Chernoff's bounds, which in the case of Bernoulli variables are expressed in terms of KL-divergences, and have motivated the KL-UCB algorithm of Cappé et al. (2013). KL-based confidence intervals are defined as

$$\sigma_y(r, p) = \inf \{ q \in [0, 1] \,|\, \exp(-r D(p(y) \| p(y) \pm q)) \leq \delta \}.$$

While they arguably provide the best asymptotic results, they introduce computational overhead that we were keen to avoid in our algorithms.

Rather than trying to find the tightest confidence intervals, our definition of $\sigma_y(r,p)$ in Theorem 5 comes from a different perspective: the optimization of elimination times. We know from Lemma 24, a variant of Theorem 1 provided in Appendix A.1 based on a method of Cramér (1938), that we can not safely eliminate $y_i$ as a mode candidate before having made $r_i \simeq \Delta_i^{-2}\ln(1/\delta)$ observations of both $\mathbf{1}_{Y_j=y_i}$ and $\mathbf{1}_{Y_i=y_1}$. The Proposition 8 below states that this is roughly the same as asking for

$$r_i \simeq p(y_1)\nabla_i^{-2}\ln(1/\delta) = p(y_1)(p(y_1)-p(y_i))^{-2}\ln(1/\delta).$$

Plugging this in the elimination criterion of Algorithm 8, we would like to define $\sigma$ so as to ensure that with high probability,

$$\hat{p}_r(y_1) - \hat{p}_r(y_i) - \sigma_{y_i}(r,\hat{p}) \geq 0$$

for any $r \leq r_i$. Assuming that the empirical probability $\hat{p}_r$ converges fast enough to $p$, a confidence parameter $\sigma$ defined as

$$\sigma_{y_i}(r,p) \simeq (p(y_1)-p(y_i))\sqrt{\frac{r_i}{r}} \simeq \sqrt{\frac{p(y_1)\ln(1/\delta)}{r}}$$

satisfies this constraint. Of course, we do not have a priori access to $p(y_1)$. Instead, we estimate it at round $r$ with $\max_{y\in\mathcal{Y}}\hat{p}_r(y)$. Up to a few constants needed to account for union bounds, this leads to $\sigma$ as defined in Theorem 5. In comparison to the results of Even-Dar, Mannor and Mansour (2006), whose algorithm requires $O(\sum_i \nabla_i^{-2}\log(1/\delta))$ samples, we gain a factor $p(y_1) < 1$, thanks to which we reach the ideal scaling in $\Delta_i^{-2}$ as long as $p(y_1)$ is bounded away from 1.

5.2. *Proofs.* In this section, we prove Theorem 5. Let $\hat{p}_r$ denotes the empirical probability of the first $r$ samples $(Y_i)_{i\leq r}$, and $S_r$ be the set $S_e$ of rejected classes, updated at each iteration $r$ by Algorithm 8. We define $\hat{y}_r = \arg\max_{y\in\mathcal{Y}\setminus S_r}\hat{p}_r(y)$, and $\delta_r = \delta/\pi^2 m r^2$, and we consider the elimination criterion

$$\hat{p}_r(y) + \sigma_r < \hat{p}_r(\hat{y}_r), \qquad \sigma_r = \sqrt{\frac{24\hat{p}_r(\hat{y}_r)\ln(1/\delta_r)}{r}}$$

from Algorithm 8 with the schedule from Theorem 5.

We have seen in the discussion of the previous subsection that the estimation of the unknown quantity $p(y_1)$ by $\hat{p}_r(\hat{y}_r)$ is a key component of the definition of $\sigma_r$. In fact, it turns out that estimating $p(y_1)$ is much easier than finding the mode $y_1$, yet can help us with that second, harder task. The next two lemmas deal with this subproblem, and are crucial to our proof of Theorem 5. Chernoff's bound for Bernoulli variables states that one needs roughly $\ln(1/\delta)/p(y)$ samples to get a good estimate of $p(y)$ up to multiplicative constants. It leads to the following lemma, proved in Appendix A.4.

LEMMA 9. *For any $c > 1$, let $\hat{p}_r$ be the empirical probability associated with $r$ random samples $(Y_i)_{i\leq r}$ independently distributed according to $p \sim P(\mathcal{Y})$. It holds that*

$$\forall\, r \geq \frac{c+1}{(c-1)^2}\frac{1}{p(y)}\ln(1/\delta), \qquad \mathbb{P}(\hat{p}_r(y) > cp(y)) \leq \delta$$

*and*

$$\forall\, r \geq \frac{c^2}{(c-1)^2}\frac{1}{p(y)}\ln(1/\delta), \qquad \mathbb{P}(\hat{p}_r(y) < c^{-1}p(y)) \leq \delta.$$

While Lemma 9 shows that one needs about $r \simeq \ln(1/\delta)/p(y_1)$ samples in order to get a good estimate of $p(y_1)$, it is not of much use by itself: as $p(y_1)$ is a priori unknown, we cannot compute this required number of samples. We could naively estimate it as the first round $r$ such that $r \max_{y \in \mathcal{Y}} \hat{p}_r(y) \geq \ln(1/\delta)$. The next lemma shows that this strategy actually works well.

LEMMA 10. *For any $r \in \mathbb{N}_{\geq 1}$, $\delta > 0$ and $c > 1$, let $\hat{y}_r = \arg\max_{y \in \mathcal{Y}} \hat{p}_r(y)$ and consider the event*

$$\mathcal{A}_r = \{r\hat{p}_r(\hat{y}_r) \leq c \ln(1/\delta)\}.$$

*Then*

$$\forall r \leq \frac{2c^2 - c}{2c + 1 + \sqrt{1 + 8c}} \frac{1}{p(y_1)} \ln(1/\delta), \qquad \mathbb{P}({}^c\mathcal{A}_r) \leq m\delta,$$

*and*

$$\forall r \geq \frac{c^2}{c + 1 - \sqrt{1 + 2c}} \frac{1}{p(y_1)} \ln(1/\delta), \qquad \mathbb{P}(\mathcal{A}_r) \leq \delta.$$

Note that by combining Lemmas 9 and 10, one can derive an efficient $(0, \delta)$-PAC algorithm to get a good estimate of $p(y_1)$ up to user-defined multiplicative constants: first, use Lemma 9 and the aforementioned constants to express the number $r_0$ of samples needed as a function of $p(y_1)$ and $\delta$. This number cannot be explicitly computed by the user, as they do not have access to $p(y_1)$; however, one can use Lemma 10 to define some event $\mathcal{A}$ associated to some constant $c$ such that once $\mathcal{A}$ does not hold any more, $r$ is bigger than $r_0$ will high probability. For any such $r$, $\hat{p}_r(\hat{y}_r)$ will be a good estimate of $p(y_1)$. Though we do not explicitly use such an algorithm, it is implicitly at the core of Theorem 4, as will become apparent in what follows.

As for the proof of Theorem 4, we define likely events that ease the analysis. Their definitions, and lower bounds on their probabilities, are provided by the following lemma, proven in Appendix A.4.

LEMMA 11. *Let us write $\tilde{\sigma}_r = \sqrt{3p(y_1)\ln(1/\delta_r)/r}$, and define the events*

- $A_1 = \{\hat{p}_r(y_1) \geq p(y_1)/2$ *for all* $r \geq 4\ln(1/\delta_r)/p(y_1)\}$,
- $A_2 = \{\hat{p}_r(\hat{y}_r) \leq 2p(y_1)$ *for all* $r \geq 4\ln(1/\delta_r)/p(y_1)\}$,
- $A_3 = \{|\hat{p}_r(y_i) - p(y_i)| \leq \tilde{\sigma}_r\}$ *for all* $r \geq 4\ln(1/\delta_r)/p(y_1)$ *and* $i \in [m]$,
- $A_4 = \{\sigma_r \geq \hat{p}_r(\hat{y}_r)$ *for all* $r \leq 4\ln(1/\delta_r)/p(y_1)\}$.

*Then $\mathbb{P}(A_i) \geq 1 - \delta/6$ for $i \in \{1, 2, 4\}$ and $\mathbb{P}(A_3) \geq 1 - \delta/3$.*

Those events will allow us to guarantee the validity of Algorithm 8 with high probability. Note that, due to a coincidence in our choice of constants, the event $A_2$ is in fact implied by $A_3$. We nonetheless keep $A_2$ as a separate event for increased readability. Intuitively, these events should be interpreted as follows: $A_1$ and $A_2$ ensure that $\hat{p}_r(\hat{y})$ is roughly equal to $p(y_1)$ once the threshold $r = \frac{4}{p(y_1)}\ln(1/\delta_r)$, which is given by Lemmas 9 and 10, is reached. Past this threshold, the quantity $\tilde{\sigma}_r = \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{r}}$ acts as a good deterministic proxy for $\sigma_r/2 = \sqrt{\frac{6\hat{p}_r(\hat{y})\ln(1/\delta_r)}{r}}$, and $A_3$ ensures that the empirical probability mass $\hat{p}_r(y_i)$ of each class $y_i$ is within an interval of width $\tilde{\sigma}_r$ centered around the true mass $p(y_i)$. Before this threshold, $A_4$ ensures that $\sigma_r$ is too large for the elimination criterion to be satisfied, hence

that no classes are eliminated before $r \leq \frac{4}{p(y_1)} \ln(1/\delta_r)$. All of this combined keeps $y_1$ from being wrongly eliminated, and yields upper bounds on the elimination times of the classes $y_i$ with $i > 1$. The next paragraphs will formalize these claims. We start by showing that the mode is never wrongfully eliminated.

LEMMA 12. *When the events $(A_i)_{i\in[4]}$ hold, the mode $y_1$ is never eliminated.*

PROOF. If $A_4$ holds, then for any $r \leq 4\ln(1/\delta_r)/p(y_1)$ we have $\hat{p}_r(y) + \sigma_r \geq \hat{p}_r(\hat{y}_r)$, hence, by definition of the elimination criterion, no classes can be eliminated at round $r$.

Let us now consider any round $r \geq 4\ln(1/\delta_r)/p(y_1)$. Assume that $y_1$ has not yet been eliminated at the start of round $r$. The event $A_3$ implies

$$\hat{p}_r(y_1) \geq p(y_1) - \tilde{\sigma}_r \geq p(y_i) - \tilde{\sigma}_r \geq \hat{p}_r(y_i) - 2\tilde{\sigma}_r,$$

while the event $A_1$ leads to $\hat{p}_r(\hat{y}_r) \geq \hat{p}_r(y_1) \geq p(y_1)/2$, hence

$$\sigma_r = \sqrt{24\hat{p}_r(\hat{y}_r)\ln(1/\delta_r)/r} \geq \sqrt{12p(y_1)\ln(1/\delta_r)/r} = 2\tilde{\sigma}_r.$$

This means that the criterion $\hat{p}_r(y_1) + \sigma_r < \hat{p}_r(\hat{y}_r)$ cannot be satisfied, and thus $y_1$ cannot be eliminated at round $r$ if it was not at round $r-1$. Recursively, we deduce that $y_1$ is never eliminated. □

Let us now focus on the elimination of the other classes $y \neq y_1$.

LEMMA 13. *When the event $(A_j)_{j\in[4]}$ hold, the class $y_i \in \mathcal{Y}\backslash\{y_1\}$ is eliminated no later than when*

$$r > 108\frac{p(y_1)}{\nabla_i^2}\ln(1/\delta_r).$$

*Since $\ln(\delta_r) = \ln(1/\delta) + 2\ln(r) + c$, this defines an elimination time*

$$r(y_i) = 108\frac{p(y_1)}{\nabla_i^2}\ln(1/\delta) + o(\ln(1/\delta)). \tag{20}$$

PROOF. Let $i > 1$, and assume that the events $(A_j)_{j\in[4]}$ hold. We know from Lemma 12 that our hypothesis implies that $y_1$ is never eliminated, hence that $\hat{p}_r(\hat{y}_r) \geq \hat{p}_r(y_1)$. Due to $A_2$, it also implies that

$$\tilde{\sigma}_r = \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{r}} \geq \sqrt{\frac{3\hat{p}_r(\hat{y}_r)\ln(1/\delta_r)}{2r}} = \frac{\sigma_r}{4}.$$

Furthermore, $A_3$ implies that

$$\hat{p}_r(\hat{y}_r) - \hat{p}_r(y_i) \geq p(y_1) - p(y_i) - 2\tilde{\sigma}_r.$$

If the following inequality holds

$$p(y_1) - p(y_i) > 6\tilde{\sigma}_r.$$

then

$$\hat{p}_r(\hat{y}_r) - \hat{p}_r(y_i) \geq p(y_1) - p(y_i) - 2\tilde{\sigma}_r > 4\tilde{\sigma}_r \ \geq \sigma_r.$$

The lemma reduces to the characterization of $r$ such that $p(y_1) - p(y_i) > 6\tilde{\sigma}_r$. □

The two previous lemmas have shown that when $(A_i)_{i\in[4]}$ hold, $y_1$ is never eliminated and $y_i$ is eliminated at the latest when $r = 108(p(y_1) - p(y_i))^{-2}p(y_1)\ln(1/\delta) + o(\ln(1/\delta))$. In particular, the algorithm ends and outputs the correct mode before $r_2$. Using Lemma 11, this happens with probability at least

$$\mathbb{P}(\cap_{i\in[4]}\mathcal{A}_i) \geq 1 - \mathbb{P}({}^cA_1 \cup {}^cA_2 \cup {}^cA_3 \cup {}^cA_4) \geq 1 - \sum_{i=1}^{4}\mathbb{P}({}^cA_i) \geq 1 - \delta, \tag{21}$$

Regarding the number of expected queries, we use one query for each sample to check if it belongs to the eliminated set. If it is not, i.e. $Y_i \notin S_r$, we expect, based on Theorem 3, to have to ask about $|\log_2(\tilde{p}(y))| \leq |\log_2(p(y))|$ queries to identify $Y_i = y$, where $\tilde{p}$ is the restriction of $p$ to the set $\mathcal{Y} \setminus S_r$.

This explains the number of queries in Theorem 5,

$$\begin{aligned}
\mathbb{E}\left[T \,\middle|\, (A_j)_{j\in[4]}\right] &\lesssim \sum_{r\leq r(y_2)}\sum_{i\in[m]} p(y_i)(1 + \mathbf{1}_{y_i\notin S_r}|\log_2(p(y_i))|) \\
&\lesssim r(y_2) + \sum_{i\in[m]} r(y_i)p(y_i)|\log_2(p(y_i))| \\
&\simeq \Big(\frac{p(y_1)}{\nabla_2^2} + \sum_{i\leq m} p(y_i)\frac{p(y_1)}{\nabla_i^2}|\log_2(p(y_i))|\Big)\ln(1/\delta).
\end{aligned}$$

The proof technicalities are provided in Appendix A.4, yielding the following lemma.

LEMMA 14. *In the setting of Theorem 5, with the event $(A_i)_{i\in[n]}$ defined in Lemma 11,*

$$\frac{\mathbb{E}\left[T|(A_j)_{j\in[4]}\right]}{\ln(1/\delta} \leq 324\frac{p(y_1)}{\nabla_2^2} + 216\sum_{i=1}^{m} p(y_i)\,|\log_2(p(y_i))|\,\frac{p(y_1)}{\nabla_i^2} + o(1)$$

Lemma 14 ends the proof of Theorem 5 and explicits the constants $C_1$ and $C_2$.

**6. Set Elimination.** The intuition behind Algorithm 7 is that given a Huffman tree with respect to $p$, one does not need to go far below the depth $-\log_2 p(y_1) + 1$ when searching for the mode (or equivalently, that classes with low probability can be grouped into sets of classes with probability roughly equal to $p(y_1)/2$). Meanwhile, the intuition behind Algorithm 8 is that one can quickly eliminate low-probability classes to focus on likely candidates. The combination of these ideas yields Algorithm 9, which outperforms both. Classes are partitioned into sets of mass roughly equal to $p(y_1)/2$, which are eliminated as soon as they appear unlikely to be the mode. Those partitions must sometimes be redefined (as the sets might be of greater mass than initially estimated), but these reorderings are rare and do not impact asymptotic performance. Hence, under the light of Proposition 8, Algorithm 9 requires roughly $\Delta_i^{-2}\log(1/\delta)$ samples to discard class $y_i$ as a mode candidate with confidence level $1-\delta$, and a sample consumes roughly $|\log_2 p(y_1)|$ queries. This explains Theorem 6.

THEOREM 6. *If Algorithm 9 is run with schedule $n_r := 2^r$ and $\epsilon_r := \frac{1}{4m}\left(\frac{2}{3}\right)^{\frac{r}{2}}$ and confidence interval parameter $\sigma$ defined from a confidence level parameter $\delta > 0$ as*[8]

$$\sigma(n,\hat{p}) = \sqrt{\frac{c\max_{z\in\mathcal{Y}\setminus S_e}\hat{p}(z)\ln(\pi^2 mn^2/\delta)}{n}}, \qquad \textit{with} \qquad c = 24, \tag{22}$$

[8]Note that although we do not have access to all the $\{\hat{p}_r(z)\}_{z\in\mathcal{Y}\setminus S_e}$ when running Algorithm 9, we have access to their maximizer (as an output of Algorithm 6), which ensures that $\sigma(n_r, \hat{p}_r)$ is computable by the user.

**Data:** $\mathcal{Y}$, $(Y_s) \sim p$, scheduling $(n_r) \in \mathbb{N}^{\mathbb{N}}$ and $(\epsilon_r) \in \mathbb{R}^{\mathbb{N}}$, confidence interval parameter $\sigma$.
Set the eliminated set $S_e = \emptyset$, $r = 0$, and $\mathcal{T}$ a binary search tree;
**while** $\mathcal{Y} \backslash S_e$ *is not a singleton* **do**
  Set $r \leftarrow r + 1$;
  Get $n_r$ fresh samples $(Y_{j,r})_{j \in [n_r]}$;
  Ask $(\mathbf{1}_{(Y_{j,r}) \in S_e})_{j \in [n_r]}$ to remove samples that belong to the eliminated set $S_e$;
  We call $\hat{p}_r$ the empirical distribution on $(Y_{j,r})_{j \in [n_r]}$ and $\hat{y}_r$ the mode of its restriction to $\mathcal{Y} \backslash S_e$;
  Run Algorithm 6 with tree $\mathcal{T}$ and parameters $\gamma = 1/2, \epsilon = \epsilon_r / \hat{p}_r(\mathcal{Y} \backslash S_e)$ on the samples $Y_{j,r} \notin S_e$;
  This updates $\mathcal{T}$ and yields a $\hat{p}_r(\hat{y}_r)/2 - \epsilon_r$-admissible partition $\mathcal{P}_r$ of $\mathcal{Y} \backslash S_e$ with respect to $\hat{p}_r$;
  Set $\hat{p}_+(S) = \hat{p}_r(S) + \sigma(n_r, \hat{p}_r)$ for $S \in \mathcal{P}_r$;
  Set $S_e = S_e \cup \{y \in S \mid S \in \mathcal{P}, \hat{p}_+(S) < \hat{p}_r(\hat{y}_r)\}$ to eliminate unlikely mode candidates;
**end**
**Result:** Return the mode estimate $\hat{y}_r$ of the last round $r$ as the estimated mode.

**Algorithm 9:** Set Elimination

*then with probability* $1 - \delta$ *Algorithm 9 terminates, identifies the right mode and consumes less than* $T$ *queries, where, conditionally to this successful event,*

$$\frac{\mathbb{E}[T]}{\log(1/\delta)} \leq C_1 \frac{p(y_1)}{\nabla_2^2} + C_2 \sum_{i \leq m} p(y_i) \frac{p(y_1)}{\nabla_i^2} \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil + o(1),$$

*for two universal constants* $C_1, C_2$, $\nabla_i := p(y_1) - p(y_1)$ *if* $i \neq 1$ *and* $\nabla_1 = \nabla_2$.

Note that the asymptotic expected number of queries required only depends on the classes $y$ that are close in probability mass to $y_1$, as the contribution to the sum of all the classes $y$ that are such that $p(y) \leq p(y_1)/2$ is smaller than $4C_2 \lceil \log_2(10/p(y_1)) \rceil \log(1/\delta)/p(y_1)$. In particular, we have freed ourselves from any direct dependence in the number $m$ of classes. On the other hand, we expect to have to precisely estimate the probability mass $p(y_i)$ of the classes for which $p(y_i)$ is close to $p(y_1)$. In line with this intuition, the dependence in those classes of the expected number of queries is roughly $C_2 \Delta_i^{-2} \lceil \log_2(10/p(y_1)) \rceil \log(1/\delta)$, which corresponds, up to a multiplicative factor, to the number of samples needed to disqualify $y_i$ as a mode candidate multiplied by the rough number of queries needed to precisely identify $y_i$ using a Huffman tree adapted to $p$. These semi-heuristic arguments suggest that there should be no easy way to improve upon Algorithm 9, besides tightening the various constants in Theorem 6 through more careful computations.

6.1. *Proofs.* This section provides the proof for Theorem 6, which is a combination of sorts of the proofs of Theorems 4 and 5. Let $r \geq 2$, $n_r = 2^r$, and $\hat{p}_r$ be the empirical probability distribution associated to the samples $(Y_{i,r})_{i \in [n_r]}$ used in the $r$-th round in Algorithm 9. We denote by $S_r$ the random set of all classes that have been eliminated in the previous rounds, and by $\hat{y}_r$ the mode of $\hat{p}_r$ restricted to $\mathcal{Y} \backslash S_r$, i.e. $\hat{y}_r = \arg\max_{y \notin S_r} \hat{p}_r(y)$. To simplify notation, we write $\sigma_r$ for $\sigma_y(n_r, \hat{p}_r)$, and set $\delta_r = \delta / \pi^2 m n_r^2$.

We start by introducing the same events as in the proof of Theorem 5, with the small nuance that the distributions $\hat{p}_r$ for $r \in \mathbb{N}$ are independent from each other, and that we only consider the subset of indices $\{n_r\}_r \subset \mathbb{N}$. The proof of the associated lemma is the same as that of Lemma 11.

LEMMA 15. *Let us write* $\tilde{\sigma}_r = \sqrt{3p(y_1) \ln(1/\delta_r)/n_r}$, *and define the events*

- $A_1 = \{\hat{p}_r(y_1) \geq p(y_1)/2$ *for all* $r$ *such that* $n_r \geq 4 \ln(1/\delta_r / p(y_1))\}$,
- $A_2 = \{\hat{p}_r(\hat{y}_r) \leq 2p(y_1)$ *for all* $r$ *such that* $n_r \geq 4 \ln(1/\delta_r)/p(y_1))\}$,
- $A_3 = \{|\hat{p}_r(y_i) - p(y_i)| \leq \tilde{\sigma}_r$ *for all* $r$ *such that* $n_r \geq 4 \ln(1/\delta_r)/p(y_1)$ *and* $i \in [m]\}$,

- $A_4 = \{\sigma_r \geq \hat{p}_r(\hat{y}_r)$ *for all* $r$ *such that* $n_r \leq 4\ln(1/\delta_r)/p(y_1)\}$.

*Then* $\mathbb{P}(\cap_{i\in[4]}\mathcal{A}_i) \geq 1-\delta$.

Once again $y_1$ cannot be eliminated when the events $(A_i)_{i\in[4]}$ hold.

LEMMA 16. *When the events* $(A_i)_{i\in[4]}$ *defined in Lemma 15 holds, Algorithm 9 does not eliminate the mode.*

PROOF. Assume that $y_1$ was not eliminated at the start of round $r$. The round iteration defines a $(\hat{p}_r(\hat{y}_r)/2-\epsilon_r)$-admissible partition $\mathcal{P}_r$ on $\mathcal{Y}\backslash S_r$ with respect to $\hat{p}_r$, and all samples $Y_{i,r}$ are identified along $\mathcal{P}_r \cup \{S_r\}$. Necessarily $\{\hat{y}_r\} \in \mathcal{P}_r$. At the end of this round, a class $y \in \mathcal{Y} \setminus S_r$ can only be added to the set $S_r$ of eliminated classes if it belongs to some set $S \subset \mathcal{Y}\backslash S_r$ such that $\hat{p}_r(S)+\sigma_r < \hat{p}_r(\hat{y}_r)$. In particular, this implies that $\hat{p}_r(y)+\sigma_r < p_r(\hat{y}_r)$, which is the criterion that was applied in the Elimination Algorithm 8 from Section 5. As in the proof of Theorem 5, the events $(A_i)_{i\in[4]}$ ensure that $\hat{p}_r(y_1)+\sigma_r < p_r(\hat{y}_r)$ is never true, which means that $y_1$ cannot be eliminated. □

Let us now estimate the expected number of queries required for round $r$, still assuming that events $(A_i)_{i\in[4]}$ are realized. We introduce the same event as in the proof of Theorem 4.

$$\mathcal{B}_r = \left\{\forall\, S \subset \mathcal{Y}, \quad \max\left\{|\hat{p}_{r-1}(S)-p(S)|, |\hat{p}_r(S)-p(S)|\right\} \leq \frac{\epsilon_{r-1}-\epsilon_r}{4}\right\}. \tag{23}$$

We have already shown that

$$\mathbb{P}({}^c\mathcal{B}_r) \leq 2^{m+1}\exp\Big(-\frac{n_{r-1}}{2}\Big(\frac{\epsilon_{r-1}-\epsilon_r}{4}\Big)^2\Big) \leq 2^{m+1}\exp\Big(-\Big(\frac{4}{3}\Big)^r\frac{C}{m^2}\Big), \tag{24}$$

for some constant $C>0$. When $\mathcal{B}_r$ holds, we can bound the number of queries similarly to what we have done for Theorem 4.

LEMMA 17. *Let* $\mathcal{T}_{r-1}$ *be the tree as updated by Algorithm 6 at the end of round* $r-1$ *of Algorithm 9. A sample* $Y_{i,r} \notin S_r$ *belonging to some set* $S \in \mathcal{P}_{r-1}$ *only consumes at round* $r$ *a number of queries smaller than or equal to the depth* $D(S)$ *of* $S$ *in the tree* $\mathcal{T}_{r-1}$. *When* $\mathcal{B}_r$ (23) *holds,*

$$D(S) \leq 2\left\lceil \log_2\Big(\frac{10}{p(y_1)}\Big)\right\rceil.$$

PROOF. For $r \geq 1$, $\mathcal{T}_r$ is built as a Huffman tree on the nodes corresponding to the sets of $\mathcal{P}_r$ with respect to $\hat{p}_r$. Similarly to the case of the Truncated Search Algorithm 7 and the proof of the associated Theorem 4, when $B_r$ is realized, all sets $S \in \mathcal{P}_{r-1}$ are either singletons or such that $\hat{p}_r(S) \leq \hat{p}_r(\hat{y}_r)/2-\epsilon_r$. Hence, when Algorithm 6 is applied to the $n_r$ samples of round $r$, a sample $Y_{i,r} \notin S_r$ belonging to some set $S \in \mathcal{P}_{r-1}$ only consumes a number of queries smaller than or equal to the depth $D(S)$ of $S$ in the tree $\mathcal{T}_{r-1}$. Using as in the proof of Theorem 4 the facts that $\epsilon_{r-1} < p(y_1)/4$ and that all $S \in \mathcal{P}_{r-1}$ (except at most one) satisfy

$$\hat{p}_{r-1}(S) \geq \frac{\hat{p}_{r-1}(\hat{y}_{r-1})/2-\epsilon_{r-1}}{2} > \frac{p(y_1)-\frac{\epsilon_{r-1}-\epsilon_r}{4}-2\epsilon_{r-1}}{4} > \frac{p(y_1)-\frac{9\epsilon_{r-1}}{4}}{4} > \frac{7p(y_1)}{64}.$$

We conclude that $D(S) \leq 2\lceil\log_2(64/7p(y_1))\rceil$ thanks to Lemma 3. □

If the event $\mathcal{B}_r$ is not realized, we can roughly upper bound the number of queries spent per sample by $m$, hence we can upper bound the total number of queries by $n_r m$. In the special case $r = 1$, we similarly need at most $n_r m$ queries.

Let $T_r$ be the total number of queries needed for round $r$ (it is a random variable). We spend $n_r$ queries at the start of the round to check whether $Y_{i,r} \in S_r$ for each sample $Y_{i,r}$, $i \in [n_r]$. We have shown that if $r \geq 2$, then

$$T_r \leq \sum_{i=1}^{n_r} 1 + 1_{\{Y_{i,r} \notin S_r\}} \left( 2 \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil + 1_{^c B_r} m \right), \tag{25}$$

while if $r = 1$, $T_r \leq n_1 m$.

As previously, we will bound (the expectation of) $1_{\{Y_{i,r} \notin S_r\}}$ by a deterministic quantity under the events $(A_j)_{j \in [4]}$.

LEMMA 18. *When the events $(A_j)_{j\in[4]}$ holds, then $y_i \neq y_1$ is added to $S_r$ no later than when $r = r(y)$, where the elimination time $r(y)$ is defined as*

$$r(y) = \min \left\{ r \,\middle|\, n_r > 108 \max \left( (p(y_1) - p(y))^{-2}, (p(y_1)/2)^{-2} \right) p(y_1) \ln(1/\delta_r) \right\}$$

*In particular*

$$n_{r(y)} \leq 216 \max \left( (p(y_1) - p(y))^{-2}, (p(y_1)/2)^{-2} \right) p(y_1) \ln(1/\delta_r).$$

PROOF. We write $\tilde{\sigma}_r = \sqrt{3p(y_1)\ln(1/\delta_r)/n_r}$; we have seen in the proof of Theorem 5 that $A_2$ implies $4\tilde{\sigma}_r \geq \sigma_r$. Let us now consider $y \in \mathcal{Y}$, and let $S_r(y)$ be the set of the partition $\mathcal{P}_r$ to which $y$ belongs, assuming that it has not yet been eliminated at the start of round $r$. Note that by definition of $\mathcal{P}_r$, either $\hat{p}_r(S_r(y)) \leq \hat{p}_r(\hat{y}_r)/2$ or $S_r(y) = \{y\}$.

We first examine the case $p(y) \leq p(y_1)/2$. The event $A_3$ implies that $\hat{p}_r(y) \leq p(y_1)/2 + \tilde{\sigma}_r$ and that $\hat{p}_r(\hat{y}_r)/2 \leq p(\hat{y}_r)/2 + \tilde{\sigma}_r/2 \leq p(y_1)/2 + \tilde{\sigma}_r/2$. Thus $\hat{p}_r(S_r(y)) \leq p(y_1)/2 + \tilde{\sigma}_r$. Thanks to $A_3$ again, we know that $\hat{p}_r(\hat{y}_r) \geq p(y_1) - \tilde{\sigma}_r$. Hence

$$\hat{p}_r(\hat{y}_r) - \hat{p}_r(S_r(y)) \geq p(y_1)/2 - 2\tilde{\sigma}_r.$$

If $p(y_1)/2 \geq 6\tilde{\sigma}_r$, then

$$\hat{p}_r(\hat{y}_r) - \hat{p}_r(S_r(y)) \geq 4\tilde{\sigma}_r \geq \sigma_r$$

and $S_r(y)$ gets eliminated at the end of round $r$.

Now let us consider the case $p(y) \geq p(y_1)/2$. Similarly, the event $A_3$ ensures that $\hat{p}_r(y) \leq p(y) + \tilde{\sigma}_r$ and that

$$\hat{p}_r(S_r(y)) \leq \hat{p}_r(\hat{y}_r)/2 \leq p(y_1)/2 + \tilde{\sigma}_r/2 \leq p(y) + \tilde{\sigma}_r/2$$

in the case where $|S_r(y)| \geq 2$, hence that

$$\hat{p}_r(S_r(y)) \leq p(y) + \tilde{\sigma}_r.$$

As above, $\hat{p}_r(\hat{y}_r) \geq p(y_1) - \tilde{\sigma}_r$, hence

$$\hat{p}_r(\hat{y}_r) - \hat{p}_r(S_r(y)) \geq p(y_1) - p(y) - 2\tilde{\sigma}_r.$$

If $p(y_1) - p(y) \geq 6\tilde{\sigma}_r$, then

$$\hat{p}_r(\hat{y}_r) - \hat{p}_r(S_r(y)) \geq 4\tilde{\sigma}_r \geq \sigma_r,$$

and $S_r(y)$ must be eliminated at the end of round $r$.

We have shown that if $p(y) \leq p(y_1)/2$, $y$ is eliminated at the latest at the end of the first round $r$ for which $p(y_1)/2 \geq 6\tilde{\sigma}_r$, and if $p(y) \geq p(y_1)/2$, it is eliminated at the latest at the

end of the first round for which $p(y_1) - p(y) \geq 6\tilde{\sigma}_r$. Note that as in the proof of Theorem 5, the condition $\rho > 6\tilde{\sigma}_r$ (for $\rho \in \{p(y_1) - p(y), p(y_1)/2\}$) is equivalent to

$$n_r > 108\rho^{-2} p(y_1) \ln(1/\delta_r),$$

which defines the elimination time $r(y)$. □

The previous lemma allows us to bound $\mathbf{1}_{Y_{i,r} \notin S_r}$ by $\mathbf{1}_{r \leq r(y)}$. We are now left with the computation of the upper bound in Equation (25). We provide the derivations in Appendix A.5, which yield the following lemma.

LEMMA 19. *With the events $(A_i)_{i\in[4]}$ as defined in Lemma 15,*

$$\begin{aligned}
\frac{\mathbb{E}\left[T \,\middle|\, (A_i)_{i\in[4]}\right]}{\ln(1/\delta)} &\leq 432 \frac{p(y_1)}{(p(y_1) - p(y_2))^2} p(y_1) \\
&+ 864 \sum_{y \in \mathcal{Y} \text{ s.t. } p(y) < p(y_1)/2} p(y) \frac{4}{p(y_1)^2} p(y_1) \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil \\
&+ 864 \sum_{y \in \mathcal{Y} \text{ s.t. } p(y) \geq p(y_1)/2} p(y) \frac{1}{(p(y_1) - p(y))^2} p(y_1) \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil + o(1).
\end{aligned}$$

Note that since

$$\max\left((p(y_1) - p(y))^{-2}, (p(y_1)/2)^{-2}\right) \leq 4(p(y_1) - p(y))^{-2}$$

for any $y \in \mathcal{Y}$, we can weaken and simplify this upper bound as

$$\frac{\mathbb{E}\left[T \,\middle|\, (A_i)_{i\in[4]}\right]}{\ln(1/\delta)} \leq 432 \frac{p(y_1)^2}{(p(y_1) - p(y_2))^2} + 3456 \sum_{y \in \mathcal{Y}} \frac{p(y_i) p(y_1)}{\nabla_i^2} \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil + o(1)).$$

This completes the proof.

*Aside on "Forever Running" Algorithms.* It is easy to reuse parts of the proofs of Theorems 5 and 6 to turn algorithms that output the empirical mode, namely the Exhaustive, Adaptive and Truncated Search Algorithms, into $(\epsilon, \delta)$-PAC algorithms for $\epsilon \geq 0$ and $\delta > 0$. On can also modify the Elimination and Set Elimination Algorithms to turn them into algorithms that run forever for ever increasing accuracy by adapting the doubling trick credited to Auer et al. (1995) to our partial feedback setting. The asymptotic expected number of queries as a function of the probability of error $\delta$ would scale in the same way as for the initial algorithms, though with worse constants.

**Conclusion.** This article introduces Problem 1, a new framework in which to formalize the problem of active learning with weak supervision. It presents three important ideas on how to solve it, namely the use of adaptive entropy coding, coarse sufficient statistics and confidence intervals, and illustrates these ideas through increasingly complex algorithms. Finally, it combines those ideas into Algorithm 9, which provably only needs an expected number

$$\mathbb{E}[T] \leq \left( C_1 \Delta_2^{-2} + C_2 \sum_{i \leq m} \Delta_i^{-2} \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil + o(1) \right) \log(1/\delta),$$

of queries to identify the mode with probability at least $1 - \delta$ (assuming that $p(y_1)$ is bounded away from 1).

# APPENDIX

**A.1. Information Projection Computation.** Recall the definitions of the set

$$\mathcal{Q} = \{q \in P(\mathcal{Y}) \,|\, \arg\max q(y) \neq \arg\max p(y)\},$$

$$\mathcal{Q}_{n,-} = \left\{q \in P(\mathcal{Y}) \cap n^{-1} \cdot \mathbb{N}^{\mathcal{Y}} \,\middle|\, y_1 \notin \arg\max q(y)\right\}, \tag{8}$$

and

$$\mathcal{Q}_{n,+} = \left\{q \in P(\mathcal{Y}) \cap n^{-1} \cdot \mathbb{N}^{\mathcal{Y}} \,\middle|\, \arg\max q(y) \neq \arg\max p(y)\right\}. \tag{9}$$

In this section, we prove Lemma 1 from Section 2, which we restate here for the reader's convenience, as well as some related results.

LEMMA 1. *For any distribution $p \in P(\mathcal{Y})$, there exists a constant $c_p$ such that for any $n \in \mathbb{N}$ with $\mathcal{Q}_{n,-}$ and $\mathcal{Q}_{n,+}$ defined by Equations (8) and (9),*

$$\frac{\Delta_2^2}{\ln(2)} + \frac{c_p}{n\ln(2)} \geq \min_{q\in\mathcal{Q}_{n,-}} D(q\|p) \geq \min_{q\in\mathcal{Q}_{n,+}} D(q\|p) \geq \min_{q\in\mathcal{Q}} D(q\|p) = \frac{\Delta_2^2}{\ln(2)}.$$

Lemma 1 is a substatement of Lemmas 21 and 23 below. We first prove the following intermediate result.

LEMMA 20. *Let $\mathcal{Y} = \{y_1, \ldots, y_m\}$. Let $q, p \in \mathbb{P}(\mathcal{Y})$ and $y', y'' \in \mathcal{Y}$ be such that*

$$\frac{q(y')}{p(y')} > \frac{q(y'')}{p(y'')}.$$

*Let us define for $\epsilon \in (0, \min\{q(y'), 1-q(y'')\}]$, the distribution $q_\epsilon \in P(\mathcal{Y})$*

$$q_\epsilon(y) = \begin{cases} q(y) - \epsilon \text{ if } y = y' \\ q(y) + \epsilon \text{ if } y = y'' \\ q(y) \quad \text{otherwise,} \end{cases}$$

*then*

$$\epsilon \in \left(0, \frac{q(y')p(y'') - q(y'')p(y')}{p(y') + q(y')}\right] \qquad \Rightarrow \qquad D(q_\epsilon\|p) < D(q\|p).$$

PROOF. Indeed,

$$D(q_\epsilon\|p) = (q(y') - \epsilon)\log\left(\frac{q(y') - \epsilon}{p(y')}\right) + (q(y'') + \epsilon)\log\left(\frac{q(y'') + \epsilon}{p(y'')}\right) + C(q,p),$$

where $C(q,p)$ are some terms that do not depend on $\epsilon$, and

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon} D(q_\epsilon\|p) = \log\left(\frac{p(y')(q(y'') + \epsilon)}{(q(y') - \epsilon)p(y'')}\right)$$

is strictly negative for $\epsilon > 0$ such that $\frac{p(y')(q(y'')+\epsilon)}{(q(y')-\epsilon)p(y'')} < 1$, which leads to the condition on $\epsilon$. □

We can now characterize the information projection $\min_{q\in Q} D(q\|p)$.

LEMMA 21. *Let $\mathcal{Y} = \{y_1, \dots, y_m\}$ and $p \in \mathbb{P}(\mathcal{Y})$ be such that $p(y_1) > p(y_2) > 0$ and $p(y_{i-1}) \geq p(y_i)$ for all $i = 2, \dots, m$. Define*

$$\mathcal{Q} := \{q \in P(\mathcal{Y}) \,|\, \exists\, y \in \mathcal{Y} \setminus \{y_1\} \text{ s.t. } q(y) \geq q(y_1)\}\,.$$

*There exists $q^* \in \mathcal{Q}$ such that $D(q^*\|p) = \min_{q \in Q} D(q\|p)$, and*[9]

$$q^*(y_1) = q^*(y_2) = \frac{1 - \lambda(1 - p(y_1) - p(y_2))}{2}$$

*and*

$$q^*(y_i) = \lambda p(y_i) \quad \forall i \in \{3, \dots, m\}$$

*for*

$$\lambda = \frac{1}{1 - (\sqrt{p(y_1)} - \sqrt{p(y_2)})^2}.$$

*It satisfies*

$$D(q^*\|p) = -\log(1 - (\sqrt{p(y_1)} - \sqrt{p(y_2)})^2).$$

PROOF. The statement is trivial when $m = 2$ or $\sum_{i=3}^m p(y_i) = 0$ (with $\lambda = 1$), which we assume henceforth not to be the case.

Observe first that if $y \in \mathcal{Y}$ is such that $p(y) = 0$, then necessarily $q^*(y) = 0$. Indeed, if $p(y) = 0$ and $q^*(y) > 0$, then $D(q^*\|p) = \infty$, while $\min_{q \in Q} D(q\|p) < 0$ (consider for example $q \in \mathbb{P}(\mathcal{Y})$ defined as $q(y_1) = q(y_2) = (p(y_1) + p(y_2))/2$ and $q(y) = p(y)$ for all $y \in \mathcal{Y} \setminus \{y_1, y_*\}$).

Let $y_* \in \mathcal{Y} \setminus \{y_1\}$ be a mode of $q^*$, which exists by hypothesis. We must have

$$\forall\, y', y'' \in \mathcal{Y} \setminus \{y_*, y_1\} \quad \text{s.t. } p(y') \neq 0, p(y'') \neq 0, \qquad \frac{q(y')}{p(y')} \leq \frac{q(y'')}{p(y'')},$$

as otherwise one could apply Lemma 20 to $y'$ and $y''$ to find $q_\epsilon \in \mathcal{Q}$ such that $q_\epsilon(y_1) = q(y_1) \leq q(y_*) = q_\epsilon$, hence $q_\epsilon \in \mathcal{Q}$, and $D(q_\epsilon\|p) < D(q\|p)$, which would contradict the definition of $q^*$.

In particular, this implies by symmetry that

$$\exists\, \lambda \geq 0 \quad \text{s.t.} \quad \forall\, y \in \mathcal{Y} \setminus \{y_1, y_*\} \quad q(y) = \lambda p(y).$$

We observe that

$$q^*(y_1) = q^*(y_*).$$

Indeed, we always have $q^*(y_*)/p(y_*) > q^*(y_1)/p(y_1)$, since $q^*(y_*) \geq q^*(y_1)$ and $p(y_1) > p(y_*)$. If $q^*(y_1) < q^*(y_*)$, we could apply Lemma 20 to $y_1, y_*$ with a small enough $\epsilon$ to find $q_\epsilon \in Q$ with $D(q_\epsilon\|p) < D(q\|p)$.

Now let us show that

$$\exists\, y \in \{y \in \mathcal{Y} \setminus \{y_1\} | q^*(y) = q^*(y_*)\} \quad \text{s.t.} \quad p(y) = p(y_2).$$

Indeed, assume that it is not the case: then $y_*$ does not satisfy those conditions, hence it must be such that $p(y_*) < p(y_2)$, and $y_2$ does not satisfy those conditions, hence it must be such

[9] This characterization of $q^*$ is exact up to some renaming of classes $y$ with equal probability masses.

that $q^*(y_*) > q^*(y_2)$. Consider the distribution $q$ defined as $q(y_2) = q^*(y_*)$, $q(y_*) = q^*(y_2)$ and $q(y) = q^*(y)$ for all $y \in \mathcal{Y} \setminus \{y_2, y_*\}$. Then $q \in Q$, and

$$\begin{aligned} D(q^*\|p) - D(q\|p) &= q^*(y_*)\log(q^*(y_*)/p(y_*)) + q^*(y_2)\log(q^*(y_2)/p(y_2)) \\ &\quad - q^*(y_2)\log(q^*(y_2)/p(y_*)) - q^*(y_*)\log(q^*(y_*)/p(y_2)) \\ &= q^*(y_*)\log(p(y_2)/p(y_*)) + q^*(y_2)\log(p(y_*)/p(y_2)) \\ &= (q^*(y_*) - q^*(y_2))\log(p(y_2)/p(y_*)) > 0, \end{aligned}$$

leading to a contradiction. Hence there exists $\tilde{y} \in \{y \in \mathcal{Y} \setminus \{y_1\} | q^*(y) = q^*(y_*)\}$ such that $p(\tilde{y}) = p(y_2)$. Consider now the distribution $q$ defined as $q(\tilde{y}) = q^*(y_2)$, $q(y_2) = q^*(\tilde{y})$ and $q(y) = q^*(y)$ for all $y \in \mathcal{Y}\setminus\{y^*, \tilde{y}\}$; then $q \in \mathcal{Q}$ and $D(q\|p) = D(q^*\|p)$, and without loss of generality we can rename $q$ into $q^*$ for the remainder of the proof.

Let us write $x = q^*(y_1) = q^*(y_2)$. As

$$1 = \sum_{i=1}^{m} q^*(y_i) = 2x + \sum_{i=3}^{m} \lambda p(y_i) = 2x + \lambda(1 - p(y_1) - p(y_2)),$$

we find that $x = (1 - \lambda(1 - p(y_1) - p(y_2)))/2$.

We should have

$$\frac{q^*(y_1)}{p(y_1)} \leq \lambda = \frac{q^*(y)}{p(y)} \leq \frac{q^*(y_2)}{p(y_2)}$$

for $y \in \mathcal{Y} \setminus \{y_1, y_2\}$, as we could otherwise apply Lemma 20 to reach a contradiction with the definition of $q^*$. This leads to the constraint

$$\frac{x}{p(y_1)} = \frac{1 - \lambda(1 - p(y_1) - p(y_2))}{2p(y_1)} \leq \lambda, \text{ i.e. } \lambda \geq \frac{1}{(1 + p(y_1) - p(y_2))},$$

as well as

$$\lambda \leq \frac{x}{p(y_2)} = \frac{1 - \lambda(1 - p(y_1) - p(y_2))}{2p(y_2)}, \text{ i.e. } \lambda \leq \frac{1}{(1 - p(y_1) + p(y_2))}.$$

Hence we have shown that $q^*$ must be some distribution $q_\lambda$ defined as $q_\lambda(y_i) = \lambda p(y_i)$ for all $i \geq 3$ and

$$q_\lambda(y_1) = q_\lambda(y_2) = \frac{1 - \lambda(1 - p(y_1) - p(y_2))}{2}.$$

for some

$$\lambda \in \left[\frac{1}{1 + p(y_1) - p(y_2)}, \frac{1}{1 - p(y_1) + p(y_2)}\right].$$

If we show that $D(q_\lambda\|p)$ is minimized for $\lambda = \frac{1}{1-(\sqrt{p(y_1)}-\sqrt{p(y_2)})^2}$ and that $D(q^*\|p) = -\log(1 - (\sqrt{p(y_1)} - \sqrt{p(y_2)})^2)$, the proof is complete; this is the statement of Lemma 22 below. □

LEMMA 22. *Consider the family of distributions $q_\lambda$ defined for $\lambda \in [0, (1 - p(y_1) - p(y_2))^{-1}]$ as $q_\lambda(y_i) = \lambda p(y_i)$ for all $i \geq 3$ and*

$$q_\lambda(y_1) = q_\lambda(y_2) = \frac{1 - \lambda(1 - p(y_1) - p(y_2))}{2}.$$

*Then*

$$\arg\min_\lambda D(q_\lambda\|p) = \frac{1}{1 - (\sqrt{p(y_1)} - \sqrt{p(y_2)})^2}$$

*and*

$$\min_{\lambda} D(q_\lambda \| p) = \log(1 - (\sqrt{p(y_1)} - \sqrt{p(y_2)})^2).$$

PROOF. We compute

$$\begin{aligned} D(q_\lambda \| p) &= \frac{1 - \lambda(1 - p(y_1) - p(y_2))}{2} \log\left(\frac{(1 - \lambda(1 - p(y_1) - p(y_2)))^2}{4p(y_1)p(y_2)}\right) \\ &\quad + \sum_{i=3}^{m} \lambda p(y_i) \log(\lambda) \\ &= \frac{1 - a\lambda}{2} \log\left(\frac{(1 - a\lambda)^2}{b}\right) + a\lambda \log(\lambda), \end{aligned}$$

where $a = 1 - p(y_1) - p(y_2)$ and $b = 4p(y_1)p(y_2)$, which implies that the derivative reads

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\lambda} D(q_\lambda \| p) &= -\frac{a}{2} \log\left(\frac{(1 - a\lambda)^2}{b}\right) + (1 - a\lambda) \cdot \frac{-a}{1 - a\lambda} + a \log(\lambda) + a \\ &= \frac{a}{2} \log\left(\frac{b\lambda^2}{(1 - a\lambda)^2}\right) \\ &= \frac{1 - p(y_1) - p(y_2)}{2} \log\left(\frac{4p(y_1)p(y_2)\lambda^2}{(1 - \lambda(1 - p(y_1) - p(y_2)))^2}\right). \end{aligned}$$

We can study the sign of this derivative,

$$\begin{aligned} \operatorname{sign}\left(\frac{\mathrm{d}}{\mathrm{d}\lambda} D(q_\lambda \| p)\right) &= \operatorname{sign}\left(\frac{b\lambda^2}{(1 - a\lambda)^2} - 1\right) = \operatorname{sign}\left(b\lambda^2 - (1 - a\lambda)^2\right) \\ &= \operatorname{sign}\left((b - a^2)\lambda^2 + 2a\lambda - 1\right). \end{aligned}$$

The roots of this polynomial are

$$\begin{aligned} \lambda_\pm &= \frac{-a \pm \sqrt{a^2 + b - a^2}}{b - a^2} = \frac{-a \pm \sqrt{b}}{b - a^2} = \frac{1}{a \pm \sqrt{b}} \\ &= \frac{1}{1 - p(y_1) - p(y_2) \pm 2\sqrt{p(y_1)p(y_2)}} = \frac{1}{1 - (\sqrt{p(y_1)} \mp \sqrt{p(y_2)})^2}. \end{aligned}$$

The sign of the leading coefficient of the polynomial is

$$\begin{aligned} \operatorname{sign}(b - a^2) &= \operatorname{sign}(\sqrt{b} - a) = \operatorname{sign}(2\sqrt{p(y_1)p(y_2)} + p(y_1) + p(y_2) - 1) \\ &= \operatorname{sign}((\sqrt{p(y_1)} + \sqrt{p(y_2)})^2 - 1) = \operatorname{sign}(\sqrt{p(y_1)} + \sqrt{p(y_2)} - 1). \end{aligned}$$

As a consequence, there are two possibilities.

If $\sqrt{p(y_1)} + \sqrt{p(y_2)} \geq 1$, then $b - a^2$, as well as $\sqrt{b} - a$, is positive, and $\mathrm{d}D/\mathrm{d}\lambda$ is negative from $\lambda_- = (a - \sqrt{b})^{-1} \leq 0$ to $\lambda_+ = (a + \sqrt{b})^{-1}$ and positive afterwards, in which case $\lambda_+$ is the minimizer of $D(q_\lambda \| p)$.

If $\sqrt{p(y_1)} + \sqrt{p(y_2)} \leq 1$, then $b - a^2$, as well as $\sqrt{b} - a$ is negative, and $\mathrm{d}D/\mathrm{d}\lambda$ is negative from $\lambda = 0$ to $\lambda_+ = (a + \sqrt{b})^{-1}$, then positive until $\lambda_- = (a - \sqrt{b})^{-1}$ and negative afterwards, in which case either $\lambda_+$ or the right extremity of the domain of $\lambda$ is the minimizer of $D(q_\lambda \| p)$ on the domain of $\lambda$. Yet this right extremity is $(1 - p(y_1) - p(y_2))^{-1}$ is smaller than $\lambda_-$, since

$$p(y_1) + p(y_2) \leq p(y_1) + 2\sqrt{p(y_1)p(y_2)} + p(y_2) = (\sqrt{p(y_1)} + \sqrt{p(y_2)})^2,$$

implies

$$(1-p(y_1)-p(y_2))^{-1} \leq (1-(\sqrt{p(y_1)}+\sqrt{p(y_2)})^2)^{-1} = \lambda_-,$$

hence the minimizer has to be $\lambda_+$.

We conclude that $\lambda_+$ is always the minimizer of $D(q_{\lambda_+}\|p)$. Hence, using that $\lambda_+$ solves $b\lambda^2 = (1-a\lambda)^2$,

$$\begin{aligned} D(q_{\lambda_+}\|p) &= \frac{1-a\lambda_+}{2}\log\left(\frac{(1-a\lambda_+)^2}{b}\right) + a\lambda_+\log(\lambda_+) \\ &= \frac{1-a\lambda_+}{2}\log\left(\lambda_+^2\right) + a\lambda_+\log(\lambda_+) = \log(\lambda_+). \end{aligned}$$

This ends the computation of the information projection. □

To close this subsection, let us finally prove the upper bound on $\min_{q\in\mathcal{Q}_{n,-}} D(q\|p)$ from Lemma 1 (in fact, we prove a little more).

LEMMA 23. *Let* $\mathcal{Y} = \{y_1,\dots,y_m\}$, $n \geq 1$,

$$\mathcal{Q}_{n,-} = \left\{q \in P(\mathcal{Y}) \cap n^{-1}\cdot\mathbb{N}^{\mathcal{Y}} \,\middle|\, y_1 \notin \arg\max q(y)\right\}$$

*and*

$$\mathcal{Q} = \left\{q \in P(\mathcal{Y}) \,|\, \arg\max q(y) \neq \arg\max p(y)\right\}.$$

*Then*

$$\begin{aligned} \min_{q\in\mathcal{Q}_{n,-}} D(q\|p) \leq \min_{q\in\mathcal{Q}} D(q\|p) \\ + \frac{m}{n}\left(\max\left\{-\log(\lambda), \log\left(\frac{1-\lambda(1-p(y_1)-p(y_2))}{2p(y_2)}\right)\right\}+1\right) \\ + \frac{1}{n}\sum_{y\in\mathcal{Y}_{>0}}\log\left(1+\frac{1}{nq^*(y)}\right) \end{aligned}$$

*where* $\lambda = (1-(\sqrt{p(y_1)}-\sqrt{p(y_2)})^2)^{-1}$ *and* $\mathcal{Y}_{>0} = \{y\in\mathcal{Y}\,|p(y)\neq 0\}$.

PROOF. Let $q^* \in \arg\min_{q\in\mathcal{Q}} D(q\|p)$ be such that $q^*(y_1) = q^*(y_2) = \frac{1-\lambda(1-p(y_1)-p(y_2))}{2}$ and $q^*(y) = p(y)$ for $\lambda = (1-(\sqrt{p(y_1)}-\sqrt{p(y_2)})^2)^{-1}$ (as in Lemma 21). Then for $n > m$, one can construct $q_n \in \mathcal{Q}_{n,+}$ such that $q_n(y) \leq q^*(y) + 1/n$ for all $y\in\mathcal{Y}$ and such that $q_n(y) = 0$ if $y \notin \mathcal{Y}_{>0}$. The general idea is as follows: let $q_n(y) = 0$ for all $y\notin\mathcal{Y}_{>0}$. Then let $q_n(y_1) = \lfloor q^*(y_1)n\rfloor/n$ and $q_n(y_2) = (\lfloor q^*(y_1)n\rfloor+1)/n$, and $q_n(y) = (\lfloor q^*(y)n\rfloor+\epsilon_{y,n})/n$ for $y\in\mathcal{Y}\backslash\{y_1,y_2\}$ such that $q^*(y)\neq 0$, with $\epsilon_{y,n}\in\{0,1\}$ chosen so that $\sum_{y\in\mathcal{Y}} q_n(y) = 1$ (small adjustments can be necessary depending on whether $q^*(y_1)n\in\mathbb{N}$ etc.).

Then

$$D(q_n\|p) = \sum_{y\in\mathcal{Y}} q_n(y)\log\left(\frac{q_n(y)}{p(y)}\right) = \sum_{y\in\mathcal{Y}} q_n(y)\log\left(\frac{q^*(y)}{p(y)}\right) + \sum_{y\in\mathcal{Y}} q_n(y)\log\left(\frac{q_n(y)}{q^*(y)}\right).$$

The first sum can be bounded as

$$\begin{aligned} \sum_{y\in\mathcal{Y}} q_n(y)\log\left(\frac{q^*(y)}{p(y)}\right) &= \sum_{y\in\mathcal{Y}} q^*(y)\log\left(\frac{q^*(y)}{p(y)}\right) + \sum_{y\in\mathcal{Y}}(q_n(y)-q^*(y))\log\left(\frac{q^*(y)}{p(y)}\right) \\ &\leq D(q^*\|p) + \frac{m}{n}\max\left\{\log(\lambda), \log\left(\frac{q^*(y_2)}{p(y_2)}\right)\right\}, \end{aligned}$$

and the second sum can be bounded as follows

$$\begin{aligned}\sum_{y\in\mathcal{Y}} q_n(y)\log\left(\frac{q_n(y)}{q^*(y)}\right) &\leq \sum_{y\in\mathcal{Y}_{>0}} (q^*(y)+1/n)\log\left(1+\frac{1}{nq^*(y)}\right)\\ &\leq \sum_{y\in\mathcal{Y}_{>0}} q^*(y)\log\left(1+\frac{1}{nq^*(y)}\right)+\sum_{y\in\mathcal{Y}_{>0}}\frac{1}{n}\log\left(1+\frac{1}{nq^*(y)}\right)\\ &\leq \sum_{y\in\mathcal{Y}_{>0}} q^*(y)\frac{1}{nq^*(y)}+\sum_{y\in\mathcal{Y}_{>0}}\frac{1}{n}\log\left(1+\frac{1}{nq^*(y)}\right)\\ &\leq \frac{m}{n}+\frac{1}{n}\sum_{y\in\mathcal{Y}_{>0}}\log\left(1+\frac{1}{nq^*(y)}\right).\end{aligned}$$

Thus

$$\begin{aligned}&\min_{q\in\mathcal{Q}_{n,-}} D(q\|p)\leq D(q_n\|p)\\ \leq& D(q^*\|p)+\frac{m}{n}\left(\max\left\{\log(\lambda),\log\left(\frac{q^*(y_2)}{p(y_2)}\right)\right\}+1\right)+\frac{1}{n}\sum_{y\in\mathcal{Y}_{>0}}\log\left(1+\frac{1}{nq^*(y)}\right).\end{aligned}$$

This concludes the proof of the Lemma. □

An alternative construction yields

$$\min_{q\in\mathcal{Q}_{n,-}} D(q\|p)\leq D(q^*\|p)+C(m,p(y_1),p(y_2))/\sqrt{n}$$

for some constant $C(m,p(y_1),p(y_2))$ that only depends on $m,p(y_1),p(y_2)$ (rather than on all $p(y)$), though at the cost of a less favorable asymptotic behaviour in $n$.

*Sanity Check with Cramér-Chernoff method.* We remark that the upper bound of Theorem 1 can be proved more directly. Consider first the simple union bound

$$\begin{aligned}\mathbb{P}(\hat{y}_n\neq y_1)&=\mathbb{P}\big(\min_{i\neq 1}\sum_{j\in[n]}\mathbf{1}_{Y_j=y_1}-\mathbf{1}_{Y_j=y_i}\leq 0\big)=\mathbb{P}\big(\cup_{i\neq 1}\big\{\sum_{j\in[n]}\mathbf{1}_{Y_j=y_1}-\mathbf{1}_{Y_j=y_i}\leq 0\big\}\big)\\ &\leq\sum_{i\neq 1}\mathbb{P}\big(\sum_{j\in[n]}\mathbf{1}_{Y_j=y_1}-\mathbf{1}_{Y_j=y_i}\leq 0\big)\leq(m-1)\mathbb{P}\big(\sum_{j\in[n]}\mathbf{1}_{Y_j=y_1}-\mathbf{1}_{Y_j=y_2}\leq 0\big).\end{aligned}$$

The term $\mathbb{P}\big(\sum_{j\in[n]}\mathbf{1}_{Y_j=y_1}-\mathbf{1}_{Y_j=y_2}\leq 0\big)$ can be bounded with Chernoff's method:

LEMMA 24 (Chernoff's bound). *Let $p\in P(\mathcal{Y})$ be a distribution over $\mathcal{Y}$, and $(Y_j)$ be $n$ independent samples distributed according to $p$ with $p(y_1)>p(y_i)$. Then*

$$\mathbb{P}(\sum_{j\in[n]}\mathbf{1}_{Y_j=y_1}-\mathbf{1}_{Y_j=y_i}\leq 0)\leq\exp\left(-n\Delta_i^2\right),$$

*where $\Delta_i$ is defined in Equation (5).*

PROOF. This can be proven using the fact that, for a random variable $X$,

$$\mathbb{P}(X\geq 0)=\inf_{t>0}\mathbb{P}(\exp(tX)\geq 1)\leq\inf_{t>0}\mathbb{E}[e^{tX}].$$

As a consequence, with $X = \sum X_j$ and $X_j = \mathbf{1}_{Y_j=y_i} - \mathbf{1}_{Y_j=y_1}$,

$$\mathbb{P}(X \geq 0) \leq \inf_{t>0} \mathbb{E}[e^{tX}] = \inf_{t>0} \mathbb{E}[e^{tX_1}]^n = \inf_{t>0} \exp(n \ln(\mathbb{E}[e^{tX_1}])).$$

We are left with a simple computation,

$$\begin{aligned}\inf_{t>0} \mathbb{E}[e^{tX_1}] &= \inf_{t>0} \left(e^t p(y_i) + (1 - p(y_i) - p(y_1)) + e^{-t} p(y_1)\right) \\ &= \left(2\sqrt{p(y_1)p(y_i)} + (1 - p(y_1) - p(y_i))\right) = \left(1 - (\sqrt{p(y_1)} - \sqrt{p(y_i)})^2\right)\end{aligned}$$

where we used that the infimum is found for $e^t = \sqrt{p(y_1)/p(y_i)} > 1$. □

In fact, we know from Cramér (1938) that Chernoff's bound is asymptotically exponentially tight. As we also have

$$\mathbb{P}(\hat{y}_n \neq y_1) \geq \mathbb{P}\big(\sum_{j\in[n]} \mathbf{1}_{Y_j=y_1} - \mathbf{1}_{Y_j=y_2} \leq 0\big),$$

we get that

$$\lim \frac{\ln(\mathbb{P}(\hat{y}_n \neq y_1))}{n} = \lim \frac{\ln(\mathbb{P}\big(\sum_{j\in[n]} \mathbf{1}_{Y_j=y_1} - \mathbf{1}_{Y_j=y_2} \leq 0\big))}{n} = \Delta_2^2.$$

As mentioned in Section 2, Dinwoodie (1992) states that $\ln(\mathbb{P}(\hat{y}_n \neq y_1))/n$ is always below $-\min_{q\in\mathcal{Q}} D(q\|p)$, while Sanov (1957) states that $\ln(\mathbb{P}(\hat{y}_n \neq y_1))/n$ converges to $-\min_{q\in\mathcal{Q}} D(q\|p)$ as $n$ grows large when $\mathcal{Y}$ is discrete. This shows without any further computations that $\min_{q\in\mathcal{Q}} D(q\|p) = \Delta_2^2$, and implies the upper bound of Theorem 1.

### A.2. Additional Proofs for Section 3.

LEMMA 3. *Let $\mathcal{T}$ be a Huffman tree with respect to a value function $v$ on its vertices such that $v(R) = 1$, where $R$ is the root of $\mathcal{T}$. Then for any vertex $V$ of $\mathcal{T}$, we have*

$$D_{\mathcal{T}}(V) \leq 2\left\lceil \log_2(1/v(V)) \right\rceil.$$

*In other words, Huffman codes are 2-balanced.*

PROOF. Let us proceed by induction on $\mathrm{depth}_{\mathcal{T}}(V)$. If $\mathrm{depth}_{\mathcal{T}}(V) = 0$, i.e. $V = R$, or if $\mathrm{depth}_{\mathcal{T}}(V) = 1$, the statement is trivial. Let us assume that $\mathrm{depth}_{\mathcal{T}}(V) \geq 2$. At some point during the construction of $\mathcal{T}$, the element $V$ was merged with another element $V'$ to create a new parent node $P$. At that point, $V$ and $V'$ were the two elements with smallest value $v$ in the heap $\mathcal{S}$. If $v(V') \geq v(V)$, then $v(P) \geq 2v(V)$. By induction,

$$\begin{aligned}\mathrm{depth}_{\mathcal{T}}(V) &= \mathrm{depth}_{\mathcal{T}}(P) + 1 \leq 2\left\lceil \log_2(1/v(P)) \right\rceil + 1 \leq 2\left\lceil \log_2(1/2v(V)) \right\rceil + 1 \\ &\leq 2\left\lceil \log_2(1/v(V)) - 1 \right\rceil + 1 \leq 2\left\lceil \log_2(1/v(V)) \right\rceil.\end{aligned}$$

If $v(V') \leq v(V)$, then any $v(P) \geq v(V)$, and as any other element $V''$ must be such that $v(V'') \geq v(V)$, the parent $\tilde{P}$ (which results from the merging of $P$ and some $V''$) of $P$ satisfies $v(\tilde{P}) \geq 2v(V)$. By induction,

$$\begin{aligned}\mathrm{depth}_{\mathcal{T}}(V) &= \mathrm{depth}_{\mathcal{T}}(\tilde{P}) + 2 \leq 2\left\lceil \log_2(1/v(\tilde{P})) \right\rceil + 2 \leq 2\left\lceil \log_2(1/2v(V)) \right\rceil + 2 \\ &\leq 2\left\lceil \log_2(1/v(V)) - 1 \right\rceil + 2 \leq 2\left\lceil \log_2(1/v(V)) \right\rceil.\end{aligned}$$

This concludes the proof. □

LEMMA 4. *With $N_{i,y}$ denoting the empirical count $\sum_{j\in[i]} \mathbf{1}_{Y_j=y}$,*

$$-\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|\geq\frac{1}{2}\}}\mathbf{1}_{N_{y,i}\geq 1}\log_2\left(\frac{N_{y,i}}{i}\right)\right] \leq 22m\log_2(n),$$

PROOF. Looking at it, we see that

$$\begin{aligned}
&-\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|\geq\frac{1}{2}\}}\mathbf{1}_{N_{y,i}\geq 1}\log_2\left(\frac{N_{y,i}}{i}\right)\right]\\
&\leq \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{P}\left(\left|\frac{N_{y,i}-ip(y)}{ip(y)}\right|\geq\frac{1}{2}\right)\log_2(i)
\end{aligned}$$

Using a simplification of Chernoff's bound for Bernoulli variables (Hoeffding, 1963), we get the following bound

$$\mathbb{P}\left(\left|\frac{N_{y,i}-ip(y)}{ip(y)}\right|\geq\frac{1}{2}\right)\leq 2\exp(-\frac{ip(y)}{10}).$$

Hence

$$\begin{aligned}
\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{P}\left(\left|\frac{N_{y,i}-ip(y)}{ip(y)}\right|\geq\frac{1}{2}\right)\log_2(i) &\leq \sum_{y\in\mathcal{Y}} p(y)\log_2(n)\sum_{i=1}^{n-1}2\sqrt{2}\exp(-\frac{ip(y)}{10})\\
&\leq 2\log_2(n)\sum_{y\in\mathcal{Y}}\frac{p(y)}{1-e^{-\frac{p(y)}{10}}}\leq 22m\log_2(n),
\end{aligned}$$

where the last inequality comes from the fact that $p\mapsto p/(1-e^{-\frac{p}{10}})$ is upper bounded by 1 for $p\in[0,1]$. Note that the constant could be improved, but not dramatically. □

LEMMA 5. *With $N_{i,y}$ denoting the empirical count $\sum_{j\in[i]} \mathbf{1}_{Y_j=y}$,*

$$-\ln(2)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\log_2\left(\frac{N_{y,i}}{ip(y)}\right)\right]\leq 4m(\ln(n)+1).$$

PROOF. We will develop the logarithm up to the second order –higher orders do not easily yield better bounds.

$$\begin{aligned}
&-\ln(2)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\log_2\left(\frac{N_{y,i}}{ip(y)}\right)\right]\\
&=-\ln(2)\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\log_2\left(1+\frac{N_{y,i}-ip(y)}{ip(y)}\right)\right]\\
&=-\mathbb{E}_{(Y_j)}\left[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\left(\frac{N_{y,i}-ip(y)}{ip(y)}-\frac{1}{2(1+\xi)^2}\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)^2\right)\right]
\end{aligned}$$

where $\xi$ is a random variable (inheriting its randomness from the $(Y_j)$) that belongs to the interval $\left[0,\frac{N_{y,i}-ip(y)}{ip(y)}\right]$ if $\frac{N_{y,i}-ip(y)}{ip(y)}\geq 0$ (respectively $\left[\frac{N_{y,i}-ip(y)}{ip(y)},0\right]$ if $\frac{N_{y,i}-ip(y)}{ip(y)}\leq 0$) and depends on the value of $N_{y,i}$. Note that the Taylor expansion of $x\mapsto\ln(1+x)$ is valid

because $\left|\frac{N_{y,i}-ip(y)}{ip(y)}\right| < \frac{1}{2}$. Now

$$
\begin{aligned}
&1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\left(-\frac{N_{y,i}-ip(y)}{ip(y)}+\frac{1}{2(1+\xi)^2}\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)^2\right)\\
&<1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|<\frac{1}{2}\}}\left(-\frac{N_{y,i}-ip(y)}{ip(y)}+2\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)^2\right)\\
&\le 2\left(\frac{N_{y,i}-ip(y)}{ip(y)+1}\right)^2-\frac{N_{y,i}-ip(y)}{ip(y)}+1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|\ge\frac{1}{2}\}}\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)\\
&\le 2\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)^2-\frac{N_{y,i}-ip(y)}{ip(y)}+1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|\ge\frac{1}{2}\}}\,2\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)^2\\
&\le 4\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)^2-\frac{N_{y,i}-ip(y)}{ip(y)}
\end{aligned}
$$

where the first inequality is valid because $1+\xi > \frac{1}{2}$. The last two inequalities are used to avoid computing means of truncated binomials, together with the observations that directly bounding $\sum_{y,i} p(y)\mathbb{E}[1_{\{|\frac{N_{y,i}-ip(y)}{ip(y)}|\ge\frac{1}{2}\}}\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)]$ with a bound on $\mathbb{P}(|\frac{N_{y,i}-ip(y)}{ip(y)}| \ge \frac{1}{2})$ does not yield a good dependence in $p(y)$, hence the rather crude upper bound with the square. Using the standard formulas for the moments of binomial variables, we get that

$$
\begin{aligned}
&\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} p(y)\mathbb{E}_{(Y_j)}\left[4\left(\frac{N_{y,i}-ip(y)}{ip(y)}\right)^2-\frac{N_{y,i}-ip(y)}{ip(y)}\right]\\
&=\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\frac{4ip(y)^2(1-p(y))}{(ip(y))^2}-0\le\sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\frac{4}{i}\le 4m(\ln(n)+1).
\end{aligned}
$$

This proves the lemma. □

### A.3. Additional Proofs for Section 4.

LEMMA 6. *The event $\mathcal{A}$ defined by (15) satisfies*

$$
\mathbb{P}\left({}^c\mathcal{A}\right) \le 2^{m+1}\exp\left(-\frac{n_{r-1}}{2}\left(\frac{\epsilon_{r-1}-\epsilon_r}{4}\right)^2\right). \tag{16}
$$

PROOF. Consider the crude union bound

$$
\begin{aligned}
\mathbb{P}({}^c\mathcal{A}) \le\ &\mathbb{P}\left(\exists S\in 2^{\mathcal{Y}} \text{ s.t. } |\hat{p}_{r-1}(S)-p(S)| > \frac{\epsilon_{r-1}-\epsilon_r}{4}\right)\\
&+\mathbb{P}\left(\exists S\in 2^{\mathcal{Y}} \text{ s.t. } |\hat{p}_{r}(S)-p(S)| > \frac{\epsilon_{r-1}-\epsilon_r}{4}\right).
\end{aligned}
$$

Summing over all possible sets $S$ and using Hoeffding's inequality, we find the following (rather rough) bounds:

$$
\mathbb{P}\left(\exists S\in 2^{\mathcal{Y}} \text{ s.t. } |\hat{p}_{r-1}(S)-p(S)| > \frac{\epsilon_{r-1}-\epsilon_r}{4}\right) \le 2^m\exp\left(-\frac{n_r}{2}\left(\frac{\epsilon_{r-1}-\epsilon_r}{4}\right)^2\right)
$$

and

$$\mathbb{P}\left(\exists S \in 2^{\mathcal{Y}} \text{ s.t. } |\hat{p}_r(S) - p(S)| > \frac{\epsilon_{r-1} - \epsilon_r}{4}\right) \leq 2^m \exp\left(-\frac{n_{r-1}}{2}\left(\frac{\epsilon_{r-1} - \epsilon_r}{4}\right)^2\right),$$

Hence the lemma. □

LEMMA 7. *When running Algorithm 7 with the schedule of Theorem 4, the expected number of queries needed for round $r$ satisfies*

$$\mathbb{E}[T_r] \leq 2^r \left(\left\lceil \log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + m2^{m+1} \exp\left(-\left(\frac{4}{3}\right)^r \frac{C}{m^2}\right)\right)$$

*for some constant $C > 0$.*

PROOF. Dissociating the number of queries when $\mathcal{A}$ holds and when it does not, together with Equation (16), the expected number of queries needed for round $r \geq 2$ satisfies

$$\begin{aligned}
\mathbb{E}[T_r] &\leq n_r \left(2\left\lceil \log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + m\mathbb{P}({}^c\mathcal{A})\right) \\
&\leq n_r \left(2\left\lceil \log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + m2^{m+1} \exp\left(-\frac{n_{r-1}}{2}\left(\frac{\epsilon_{r-1} - \epsilon_r}{4}\right)^2\right)\right) \\
&\leq 2^r \left(2\left\lceil \log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + m2^{m+1} \exp\left(-2^{r-6}\left(\frac{2}{3}\right)^{r-1}(\epsilon_0 - \epsilon_1)^2\right)\right) \\
&\leq 2^r \left(2\left\lceil \log_2\left(\frac{4}{p(y_1)}\right)\right\rceil + m2^{m+1} \exp\left(-\left(\frac{4}{3}\right)^r \frac{C}{m^2}\right)\right)
\end{aligned}$$

for some constant $C > 0$. In the special case $r = 1$, we need at most

$$\mathbb{E}[T_1] \leq n_1 m = 2m$$

queries to complete the round. □

**A.4. Additional Proofs for Section 5.** Let us first prove Proposition 8, which we restate here.

PROPOSITION 8. *Let $p \in P(\mathcal{Y})$ with $p(y_1) > p(y_i)$ for all $i \in [m]$. We have the following inequality, with $\nabla_i := p(y_1) - p(y_i)$ and $\Delta_i^2 = -\ln(1 - (\sqrt{p(y_1)} - \sqrt{p(y_i)})^2)$,*

$$\frac{p(y_1)}{-\ln(1 - p(y_1))} p(y_1) \nabla_i^{-2} \leq \Delta_i^{-2} \leq 4p(y_1)\nabla_i^{-2}.$$

*In particular, if $p(y_1)$ is bounded away from 1, then $-p(y_1)/\ln(1 - p(y_1)$ is bounded away from 0 and $\Delta_i^{-2} \simeq p(y_1)\nabla_i^{-2}$ up to a multiplicative constants.*

PROOF. $(\sqrt{p(y_1)} + \sqrt{p(y_i)})^2 \leq 2(p(y_i) + p(y_1))$, together with $\ln(1 + x) \leq x$, imply

$$\Delta_i^2 = -\ln\Big(1 - \frac{(p(y_1) - p(y_i))^2}{(\sqrt{p(y_1)} + \sqrt{p(y_i)})^2}\Big) \geq \frac{(p(y_1) - p(y_i))^2}{2(p(y_1) + p(y_i))} \geq \frac{(p(y_1) - p(y_i))^2}{4p(y_1)}.$$

For the other side, note that $\ln$ is always above its cords, hence for $x \in [a, b]$,

$$\ln(1 - x) \geq \frac{(x - a)}{b - a}(\ln(1 - b) - \ln(1 - a)) + \ln(1 - a).$$

Applied to $a=0$ and $b=p(y_1)$, we get that

$$\forall\, x \in [0, p(y_1)], \qquad \ln(1-x) \geq \frac{x \ln(1-p(y_1))}{p(y_1)}.$$

As a consequence,

$$\Delta_i^2 \leq \frac{-(p(y_1)-p(y_i))^2}{(\sqrt{p(y_1)}+\sqrt{p(y_i)})^2} \frac{\ln(1-p(y_1))}{p(y_1)} \leq \frac{-(p(y_1)-p(y_i))^2}{p(y_1)} \frac{\ln(1-p(y_1))}{p(y_1)}.$$

This completes the proof. □

We now recall the following classical concentration result.

LEMMA 25 (Chernoff's bound). *Let $(X_i)_{i=1}^n$ be independent Bernoulli variables with mean $p$. For any $\lambda \geq 0$,*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i\in[n]} X_i > (1+\lambda)p\right) \leq \exp\left(-\frac{n\lambda^2 p}{2+\lambda}\right).$$

*Similarly, for $\lambda \in (0,1)$*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i\in[n]} X_i < (1-\lambda)p\right) \leq \exp\left(-\frac{n\lambda^2 p}{2}\right).$$

PROOF. Those are classical relaxations of some results that can be found in Hoeffding (1963). □

LEMMA 9. *For any $c>1$, let $\hat{p}_r$ be the empirical probability associated with $r$ random samples $(Y_i)_{i\leq r}$ independently distributed according to $p \sim P(\mathcal{Y})$. It holds that*

$$\forall\, r \geq \frac{c+1}{(c-1)^2}\frac{1}{p(y)}\ln(1/\delta), \qquad \mathbb{P}(\hat{p}_r(y) > cp(y)) \leq \delta$$

*and*

$$\forall\, r \geq \frac{c^2}{(c-1)^2}\frac{1}{p(y)}\ln(1/\delta), \qquad \mathbb{P}(\hat{p}_r(y) < c^{-1}p(y)) \leq \delta.$$

PROOF. This follows from Lemma 25 since

$$\begin{aligned}\mathbb{P}(\hat{p}_r(y) > cp(y)) &= \mathbb{P}(\hat{p}_r(y) > (1+(c-1))p(y)) \leq \exp\left(\frac{-r(c-1)^2p(y)}{2+(c-1)}\right)\\ &= \exp\left(\frac{-r(c-1)^2p(y)}{c+1}\right) \leq \exp(\ln(1/\delta) = \delta,\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}(\hat{p}_r(y) > c^{-1}p(y)) &= \mathbb{P}(\hat{p}_r(y) > (1-(1-c^{-1}))p(y))\\ &\leq \exp\left(\frac{-r(1-c^{-1})^2p(y)}{2}\right) \leq \exp(\ln(1/\delta)) = \delta.\end{aligned}$$

This explains the result of the Lemma. □

LEMMA 10. *For any $r \in \mathbb{N}_{\geq 1}$, $\delta > 0$ and $c > 1$, let $\hat{y}_r = \arg\max_{y \in \mathcal{Y}} \hat{p}_r(y)$ and consider the event*

$$\mathcal{A}_r = \{ r\hat{p}_r(\hat{y}_r) \leq c \ln(1/\delta) \} .$$

*Then*

$$\forall\, r \leq \frac{2c^2 - c}{2c + 1 + \sqrt{1+8c}} \frac{1}{p(y_1)} \ln(1/\delta), \qquad \mathbb{P}({}^c\mathcal{A}_r) \leq m\delta,$$

*and*

$$\forall\, r \geq \frac{c^2}{c + 1 - \sqrt{1+2c}} \frac{1}{p(y_1)} \ln(1/\delta), \qquad \mathbb{P}(\mathcal{A}_r) \leq \delta.$$

PROOF. We can upper bound the probability of $\mathcal{A}_r$ not happening with

$$\mathbb{P}({}^c\mathcal{A}_r) = \mathbb{P}(r\hat{p}_r(\hat{y}_r) > c\ln(1/\delta)) = \mathbb{P}(r \max_{y \in \mathcal{Y}} \hat{p}_r(y) > c\ln(1/\delta))$$
$$\leq \sum_{y \in \mathcal{Y}} \mathbb{P}(r\hat{p}_r(y) > c\ln(1/\delta)) \leq m\mathbb{P}(r\hat{p}_r(y_1) > c\ln(1/\delta)).$$

Let us set

$$\alpha_r = \frac{c\ln(1/\delta)}{rp(y_1)} - 1, \qquad \text{i.e.,} \qquad r = \frac{c\ln(1/\delta)}{p(y_1)(1+\alpha_r)}.$$

Note that $\alpha_r \geq 0$ when $r \leq \frac{2c^2-2c}{2c+1+\sqrt{1+8c}} \frac{1}{p(y_1)} \ln(1/\delta)$. Using Lemma 25 leads to

$$\mathbb{P}({}^c\mathcal{A}_r) \leq m\mathbb{P}\Big(\hat{p}_r(y_1) - p(y_1) > c\ln(1/\delta)/r - p(y_1)\Big) = m\mathbb{P}\Big(\hat{p}_r(y_1) - p(y_1) > \alpha_r p(y_1)\Big)$$
$$\leq m \exp\left(\frac{-r\alpha_r^2 p(y_1)}{2+\alpha_r}\right) = m \exp\left(\frac{-c\ln(1/\delta)\alpha_r^2}{(1+\alpha_r)(2+\alpha_r)}\right).$$

We check that

$$r \leq \frac{2c^2-2c}{2c+1+\sqrt{1+8c}} \frac{1}{p(y_1)} \ln(1/\delta) \quad \Leftrightarrow \quad \frac{c}{1+\alpha_r} \leq \frac{2c^2-2c}{2c+1+\sqrt{1+8c}}$$
$$\Leftrightarrow \quad \alpha_r \geq \frac{3+\sqrt{1+8c}}{2(c-1)} \quad \Rightarrow \quad (c-1)\alpha_r^2 - 3\alpha_r - 2 \geq 0 \quad \Leftrightarrow \quad \frac{c\alpha_r^2}{(1+\alpha_r)(2+\alpha_r)} \geq 1,$$

which allows us to conclude that $\mathbb{P}({}^c\mathcal{A}_r) \leq m\delta$. This explains the condition on $r$ stated in the Lemma.

We can similarly lower bound the probability of $\mathcal{A}_r$ happening with

$$\mathbb{P}(\mathcal{A}_r) = \mathbb{P}(r\hat{p}_r(\hat{y}_r) < c\ln(1/\delta)) = \mathbb{P}(r \max_{y \in \mathcal{Y}} \hat{p}_r(y) < c\ln(1/\delta)) \leq \mathbb{P}(r\hat{p}_r(y_1) < c\ln(1/\delta)).$$

This time we set

$$\alpha_r = 1 - \frac{c\ln(1/\delta)}{rp(y_1)}, \qquad \text{i.e.,} \qquad r = \frac{c\ln(1/\delta)}{p(y_1)(1-\alpha_r)}.$$

One can check that if $r \geq \frac{c^2}{c+1-\sqrt{1+2c}} \frac{1}{p(y_1)} \ln(1/\delta)$, then $\alpha_r \in (0,1)$. Using Lemma 25,

$$\mathbb{P}(\mathcal{A}_r) \leq \mathbb{P}\Big(\hat{p}_r(y_1) - p(y_1) < c\ln(1/\delta)/r - p(y_1)\Big) = \mathbb{P}\Big(\hat{p}_r(y_1) - p(y_1) < -\alpha_r p(y_1)\Big)$$
$$\leq \exp\left(\frac{-r\alpha_r^2 p(y_1)}{2}\right) \leq \exp\left(\frac{-c\ln(1/\delta)\alpha_r^2}{2(1-\alpha_r)}\right).$$

We check that

$$r \geq \frac{c^2}{c+1-\sqrt{1+2c}} \frac{1}{p(y_1)} \ln(1/\delta) \qquad \Leftrightarrow \qquad \frac{c}{1-\alpha_r} \geq \frac{c^2}{c+1-\sqrt{1+2c}}$$

$$\Leftrightarrow \qquad \alpha_r \geq \frac{\sqrt{1+2c}-1}{c} \qquad \Rightarrow \qquad c\alpha_r^2 + 2\alpha_r - 2 \geq 0 \qquad \Leftrightarrow \qquad \frac{c\alpha_r^2}{2(1-\alpha_r)} \geq 1.$$

We conclude that $\mathbb{P}(\mathcal{A}_r) \leq \delta$ for the $r$ defined in the Lemma. □

LEMMA 11. *Let us write* $\tilde{\sigma}_r = \sqrt{3p(y_1)\ln(1/\delta_r)/r}$, *and define the events*

- $A_1 = \{\hat{p}_r(y_1) \geq p(y_1)/2$ *for all* $r \geq 4\ln(1/\delta_r)/p(y_1)\}$,
- $A_2 = \{\hat{p}_r(\hat{y}_r) \leq 2p(y_1)$ *for all* $r \geq 4\ln(1/\delta_r)/p(y_1)\}$,
- $A_3 = \{|\hat{p}_r(y_i) - p(y_i)| \leq \tilde{\sigma}_r\}$ *for all* $r \geq 4\ln(1/\delta_r)/p(y_1)$ *and* $i \in [m]$,
- $A_4 = \{\sigma_r \geq \hat{p}_r(\hat{y}_r)$ *for all* $r \leq 4\ln(1/\delta_r)/p(y_1)\}$.

*Then* $\mathbb{P}(A_i) \geq 1 - \delta/6$ *for* $i \in \{1,2,4\}$ *and* $\mathbb{P}(A_3) \geq 1 - \delta/3$.

PROOF. We can apply the second statement of Lemma 9 with $c = 2$ to $y_1$ to see that for any $r \geq \frac{4}{p(y_1)}\ln(1/\delta_r) = \frac{2^2}{(2-1)^2}\frac{1}{p(y_1)}\ln(1/\delta_r)$, we have

$$\mathbb{P}\left(\hat{p}_r(y_1) < p(y_1)/2\right) \leq \delta_r.$$

Hence

$$\mathbb{P}({}^cA_1) \leq \sum_{r \geq \frac{4}{p(y_1)}\ln(1/\delta_r)} \delta_r \leq \sum_{r \geq 1} \delta_r \leq \delta/6.$$

Similarly, we can apply the first statement of Lemma 9 with $c = 2$ to $y_1$ to see that for any $r \geq \frac{4}{p(y_1)}\ln(1/\delta_r) \geq \frac{2+1}{(2-1)^2}\frac{1}{p(y_1)}\ln(1/\delta_r)$ and any $y \in \mathcal{Y}$, we have

$$\mathbb{P}\left(\hat{p}_r(y) > 2p(y_1)\right) \leq \mathbb{P}\left(\hat{p}_r(y_1) > 2p(y_1)\right) \leq \delta_r.$$

Hence, using a union bound over all $y \in \mathcal{Y}$, we find that

$$\mathbb{P}\left(\hat{p}_r(\hat{y}) < 2p(y_1)\right) \leq m\delta_r \qquad \forall r \geq \frac{4}{p(y_1)}\ln(1/\delta_r), \tag{26}$$

thus

$$\mathbb{P}({}^cA_2) \leq \sum_{r \geq \frac{4}{p(y_1)}\ln(1/\delta_r)} m\delta_r \leq \sum_{r \geq 1} m\delta_r \leq \delta/6.$$

We can now consider $A_3$. Let $i \in \{1,\ldots,m\}$. Applying[10] Lemma 25, we get that if $r \geq \frac{4}{p(y_1)}\ln(1/\delta_r)$ (hence $\sqrt{\frac{4\ln(1/\delta_r)}{p(y_1)r}} \in (0,1)$), then

$$\begin{aligned}
\mathbb{P}\left(\hat{p}_r(y_i) < p(y_i) - \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{r}}\right) &= \mathbb{P}\left(\hat{p}_r(y_i) < p(y_i)\left(1 - \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{p(y_i)^2 r}}\right)\right) \\
&\leq \exp\left(-\frac{rp(y_i)3p(y_1)\ln(1/\delta_r)}{2p(y_i)^2 r}\right) \\
&= \exp\left(-\ln(1/\delta_r)\frac{3}{2}\right) \leq \delta_r.
\end{aligned}$$

[10] Note that we can also use Hoeffding inequality, which leads to

$$\mathbb{P}\left(\hat{p}_r(y_i) \geq p(y_i) + \tilde{\sigma}_r\right) \leq \exp(-r\tilde{\sigma}_r^2/2)$$

which is tighter when $p(y_i) > 1 - \tilde{\sigma}_r/2$.

Using this time the second inequality of Lemma 25, and writing $\tilde{\sigma}_r = \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{r}}$, we see that for any $r \geq 1$ we have

$$\mathbb{P}\left(\hat{p}_r(y_i) > p(y_i) + \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{r}}\right) = \mathbb{P}\left(\hat{p}_r(y_i) > p(y_i)\left(1 + \frac{\tilde{\sigma}_r}{p(y_i)}\right)\right)$$

$$\leq \exp\left(-\frac{rp(y_i)\frac{\tilde{\sigma}_r^2}{p(y_i)^2}}{2 + \frac{\tilde{\sigma}_r}{p(y_i)}}\right) = \exp\left(-\frac{r\tilde{\sigma}_r^2}{2p(y_i) + \tilde{\sigma}_r}\right)$$

$$\leq \exp\left(-\frac{r\tilde{\sigma}_r^2}{2p(y_1) + \tilde{\sigma}_r}\right).$$

Furthermore, if $r \geq \frac{4}{p(y_1)}\ln(1/\delta_r)$, then $\tilde{\sigma}_r = \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{r}} \leq \sqrt{\frac{3}{4}}p(y_1)$ and consequently

$$\exp\left(-\frac{r\tilde{\sigma}_r^2}{2p(y_1) + \tilde{\sigma}_r}\right) \leq \exp\left(-\frac{r\tilde{\sigma}_r^2}{p(y_1)(2 + \sqrt{\frac{3}{4}})}\right) \leq \exp\left(-\ln(1/\delta_r)\frac{3}{2 + \sqrt{\frac{3}{4}}}\right) \leq \delta_r.$$

Combining those two bounds, we find that

$$\mathbb{P}({}^cA_3) \leq \sum_{r \geq \frac{4}{p(y_1)}\ln(1/\delta_r)} \sum_{i=1}^{m} \mathbb{P}\left(|\hat{p}_r(y_i) - p(y_i)| > \sqrt{\frac{3p(y_1)\ln(1/\delta_r)}{r}}\right) \leq \sum_{r \geq 1} 2m\delta_r \leq \delta/3.$$

Let us now consider $A_4$. We can apply Lemma 10 (with $c = 24$) to see that if

$$r \leq \frac{4}{p(y_1)}\ln(1/\delta_r) \leq \frac{2 \cdot 24^2 - 24}{2 \cdot 24 + 1 + \sqrt{1 + 8 \cdot 24}}\frac{1}{p(y_1)}\ln(1/\delta_r),$$

then

$$\mathbb{P}(\sigma_r < \hat{p}_r(\hat{y})) = \mathbb{P}(r\hat{p}_r(\hat{y}) > 24\ln(1/\delta_r)) \leq \mathbb{P}(r\hat{p}_r(\hat{y}_{all}) > 24\ln(1/\delta_r)) \leq m\delta_r.$$

Hence, using an union bound over all $r \leq \frac{4}{p(y_1)}\ln(1/\delta_r)$, we get that

$$\mathbb{P}({}^cA_4) \leq \sum_{r \leq \frac{4}{p(y_1)}\ln(1/\delta_r)} m\delta_r \leq \sum_{r \geq 1} m\delta_r \leq \delta/6.$$

This ends the proof □

LEMMA 14. *In the setting of Theorem 5, with the event $(A_i)_{i\in[n]}$ defined in Lemma 11,*

$$\frac{\mathbb{E}\left[T|(A_j)_{j\in[4]}\right]}{\ln(1/\delta} \leq 324\frac{p(y_1)}{\nabla_2^2} + 216\sum_{i=1}^{m} p(y_i)\left|\log_2(p(y_i))\right|\frac{p(y_1)}{\nabla_i^2} + o(1)$$

PROOF. For $r \geq 1$ such that Algorithm 8 has not yet terminated at the start of round $r$, let $S_r$ be the set of all classes that have been eliminated in the previous rounds (it is a random set), and let $Q_r$ be the number of queries necessary to identify $Y_r$ according to the partition $\{S_r\} \cup \{\{y\} \,|\, y \in \mathcal{Y}\backslash S_r\}$ using first the query $\mathbf{1}_{Y_r \in S_r}$, then a Huffman tree adapted to the (renormalized) empirical distribution $\frac{\hat{p}_r}{1-\hat{p}_r(S_r)}$ on $\mathcal{Y}\backslash S_r$ if $Y_r \notin S_r$. Hence $Q_r = 1$ if $Y_r \in S_r$, and as in Subsection 3.4,

$$Q_r \leq 1 + 2 + 2 \cdot \mathbf{1}_{\{\hat{p}_r(y) \neq 0\}}\log_2\left(\frac{1 - \hat{p}_r(S_r)}{\hat{p}_r(y)}\right) + \mathbf{1}_{\{\hat{p}_r(y)=0\}}m$$

if $Y_r = y \in \mathcal{Y} \backslash S_r$.

For $y \in \mathcal{Y} \backslash \{y_1\}$, let $r(y)$ be as above the smallest $r_0 \in \mathbb{N}$ such that

$$r > 108 \frac{1}{(p(y_1) - p(y))^2} p(y_1) \ln(1/\delta_r)$$

for all $r \geq r_0$, with the additional convention that $r(y_1) := r(y_2)$. As seen in Lemma 13, if $A_1, A_2, A_3$ and $A_4$ hold, then the class $y$ necessarily belongs to $S_r$ at the start of round $r$ as soon as $r > r(y)$, hence $1_{\{y \notin S_r\}} \leq 1_{\{r \leq r(y)\}}$. Thus the conditional expectation of $Q_r$ with respect to the events $\{A_i\}_{i=1}^4$ satisfies

$$\begin{aligned}
\mathbb{E}\left[Q_r | \{A_i\}_{i=1}^4\right] &\leq 3 + \mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)=0\}} m \,\middle|\, \{A_i\}_{i=1}^4\right] \\
&\quad + 2\mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)\neq 0\}} 1_{\{y \notin S_r\}} \log_2\left(\frac{1 - \hat{p}_r(S_r)}{\hat{p}_r(y)}\right) \,\middle|\, \{A_i\}_{i=1}^4\right] \\
&\leq 3 + \mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)=0\}} m \,\middle|\, \{A_i\}_{i=1}^4\right] \\
&\quad + 2\mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)\neq 0\}} 1_{\{r \leq r(y)\}} \log_2\left(\frac{1}{\hat{p}_r(y)}\right) \,\middle|\, \{A_i\}_{i=1}^4\right].
\end{aligned}$$

Now let $\Omega$ be the probability space of all possible outcomes, and let $S(\omega)$ denote the (positive) random variable $\mathbf{1}_{\{\hat{p}_r(y)\neq 0\}} 1_{\{r \leq r(y)\}} \log_2\left(\frac{1}{\hat{p}_r(y)}\right)$. Then

$$\begin{aligned}
\mathbb{E}\left[S(\omega) \,\middle|\, \{A_i\}_{i=1}^4\right] &= \frac{1}{\mathbb{P}(\cap_{i=1}^4 A_i)} \int_{\cap_{i=1}^4 A_i} S(\omega) d_\omega \\
&\leq \frac{1}{\mathbb{P}(\cap_{i=1}^4 A_i)} \int_{\Omega} S(\omega) d_\omega = \frac{1}{\mathbb{P}(\cap_{i=1}^4 A_i)} \mathbb{E}\left[S(\omega)\right].
\end{aligned}$$

As $\mathbb{P}(\cap_{i=1}^4 A_i) \geq 1 - \delta$, this means that

$$\begin{aligned}
&\mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)\neq 0\}} 1_{\{r \leq r(y)\}} \log_2\left(\frac{1}{\hat{p}_r(y)}\right) \,\middle|\, \{A_i\}_{i=1}^4\right] \\
&\leq \frac{1}{(1-\delta)} \mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)\neq 0\}} 1_{\{r \leq r(y)\}} \log_2\left(\frac{1}{\hat{p}_r(y)}\right)\right] \\
&= \frac{1}{(1-\delta)} \sum_{y \in \mathcal{Y}} p(y) 1_{\{r \leq r(y)\}} \mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)\neq 0\}} \log_2\left(\frac{1}{\hat{p}_r(y)}\right)\right]
\end{aligned}$$

We have already shown in Subsection 3.4 that for any $y$ such that $p(y) \neq 0$,

$$\mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)\neq 0\}} \log_2\left(\frac{1}{\hat{p}_r(y)}\right)\right] \leq \log_2(p(y)) + \frac{4}{p(y) r \ln(2)} + 2\exp\left(-\frac{rp(y)}{10}\right) \log_2(r).$$

As we have assumed that $A_1, A_2, A_3$ and $A_4$ hold, we know that Algorithm 8 terminates at the end of some round $R \leq r(y_2)$. Let $T = \sum_{r=1}^R Q_r$ be the number of queries used to identify those $R$ samples. Combining the results above, we find that

$$\begin{aligned}
\mathbb{E}\left[T_R | \{A_i\}_{i=1}^4\right] &\leq 3R + \sum_{r=1}^R \mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)=0\}} m \,\middle|\, \{A_i\}_{i=1}^4\right] \\
&\quad + 2\sum_{y \in \mathcal{Y}} \sum_{r=1}^{r(y)} \frac{p(y)}{(1-\delta)} \left(\left|\log_2(p(y))\right| + \frac{4}{p(y) r \ln(2)} + 2\exp\left(-\frac{rp(y)}{10}\right) \log_2(r)\right),
\end{aligned}$$

where the terms in the sum are understood to be 0 for those classes $y$ such that $p(y) = 0$. As $\mathbf{1}_{\{\hat{p}_r(y)=0\}}$ can only be non-zero for a single index $r$ for each class $y$,

$$\sum_{r=1}^{R} \mathbb{E}\left[\mathbf{1}_{\{\hat{p}_r(y)=0\}} m \,\middle|\, \{A_i\}_{i=1}^4\right] \leq m^2.$$

Furthermore,

$$\sum_{y\in\mathcal{Y}} \sum_{r=1}^{r(y)} \frac{4}{r\ln(2)} \leq \frac{4m(\ln(R)+1)}{\ln(2)}$$

and

$$\sum_{y\in\mathcal{Y}} \sum_{r=1}^{r(y)} p(y) 2\exp\left(-\frac{rp(y)}{10}\right)\log_2(r) \leq 2\sum_{y\in\mathcal{Y}} \frac{p(y)}{1-e^{\frac{p(y)}{10}}}\log_2(R) \leq 22m\log_2(R)$$

due to $p \mapsto p/(1-e^{-\frac{p}{10}})$ being upper bounded by 11 for $p \in [0,1]$. Finally, using the fact that $r(y_i) = 108\tilde{\Delta}_i^{-2} p(y_1)\ln(1/\delta) + o(\ln(1/\delta))$ (as seen in Equation 20), we see that

$$\begin{aligned}\sum_{y\in\mathcal{Y}} \sum_{r=1}^{r(y)} p(y)\left|\log_2(p(y))\right| &= \sum_{i=1}^{m} p(y_i)\left|\log_2(p(y_i))\right| r(y_i) \\ &= 108\sum_{i=1}^{m} p(y_i)\left|\log_2(p(y_i))\right| \tilde{\Delta}_i^{-2} p(y_1)\ln(1/\delta) + o(\ln(1/\delta)).\end{aligned}$$

Thus

$$\begin{aligned}\mathbb{E}\left[T_R|\{A_i\}_{i=1}^4\right] &\leq 3r(y_2) + \frac{216}{1-\delta}\sum_{i=1}^{m} p(y_i)\left|\log_2(p(y_i))\right| \tilde{\Delta}_i^{-2} p(y_1)\ln(1/\delta) + o(\ln(1/\delta)) \\ &\quad + m^2 + \frac{2}{1-\delta}\left(\frac{4m(\ln(r_2)+1)}{\ln(2)} + 22m\log_2(r_2)\right) \\ &= 324\tilde{\Delta}_2^{-2} p(y_1)\ln(1/\delta) + 216\sum_{i=1}^{m} p(y_i)\left|\log_2(p(y_i))\right| \tilde{\Delta}_i^{-2} p(y_1)\ln(1/\delta) \\ &\quad + o(\ln(1/\delta))\end{aligned}$$

This completes the proof. □

### A.5. Additional Proofs for Section 6.

LEMMA 19. *With the events $(A_i)_{i\in[4]}$ as defined in Lemma 15,*

$$\begin{aligned}\frac{\mathbb{E}\left[T \,\middle|\, (A_i)_{i\in[4]}\right]}{\ln(1/\delta)} &\leq 432\frac{p(y_1)}{(p(y_1)-p(y_2))^2}p(y_1) \\ &+ 864\sum_{y\in\mathcal{Y}\ s.t.\ p(y)<p(y_1)/2} p(y)\frac{4}{p(y_1)^2}p(y_1)\left\lceil\log_2\left(\frac{10}{p(y_1)}\right)\right\rceil \\ &+ 864\sum_{y\in\mathcal{Y}\ s.t.\ p(y)\geq p(y_1)/2} p(y)\frac{1}{(p(y_1)-p(y))^2}p(y_1)\left\lceil\log_2\left(\frac{10}{p(y_1)}\right)\right\rceil + o(1).\end{aligned}$$

PROOF. Under $(A_j)_{j\in[4]}$, we have the bound $1_{\{Y_{i,r}\notin S_r\}} \leq 1_{\{r\leq r(Y_{i,r})\}}$. Let us add the convention that $r(y_1) := r(y_2)$. We see that the algorithm necessarily terminates at the end of some round $R \leq r(y_1)$. Going back to Equation (25), we deduce that the conditional expectation of $T_r$ (for $2 \leq r \leq R$) with respect to the events $\{A_i\}_{i=1}^4$ satisfies

$$\mathbb{E}\left[T_r \,\middle|\, (A_i)_{i\in[4]}\right] \leq \mathbb{E}\Big[\sum_{i=1}^{n_r} 1 + 1_{\{r\leq r(Y_{i,r})\}}\left(2\left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil + 1_{{}^cB_r}m\right)\Big|(A_i)_{i\in[4]}\Big]$$

Using the same simple integral argument as in the proof of Theorem 5, we further see that as $\mathbb{P}(\cap_{i=1}^4 A_i) \geq 1-\delta$, then

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n_r} 1 + 1_{\{r\leq r(Y_{i,r})\}}\left(2\left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil + 1_{{}^cB_r}m\right)\,\middle|\,\{A_i\}_{i=1}^4\right] \\
&\leq \frac{1}{1-\delta}\mathbb{E}\left[\sum_{i=1}^{n_r} 1 + 1_{\{r\leq r(Y_{i,r})\}}\left(2\left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil + 1_{{}^cB_r}m\right)\right].
\end{aligned}$$

We can further write

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n_r} 1 + 1_{\{r\leq r(Y_{i,r})\}}\left(2\left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil + 1_{{}^cB_r}m\right)\right] \\
&\leq n_r + n_r\mathbb{P}({}^cB_r)m + 2n_r\sum_{y\in\mathcal{Y}} p(y)1_{\{r\leq r(y)\}}\left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil
\end{aligned}$$

We can now bound the expectation of the total number $T$ of queries required before the algorithm terminates at the end of round $R$:

$$\begin{aligned}
(1-\delta)\mathbb{E}[T] \leq n_1 m + \sum_{r=2}^{R} n_r + \sum_{r=2}^{R} n_r\mathbb{P}({}^cB_r)m \\
+ \sum_{r=2}^{R} 2n_r\sum_{y\in\mathcal{Y}} p(y)1_{\{r\leq r(y)\}}\left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil.
\end{aligned}$$

Observe that $\sum_{r=2}^{R} n_r \leq 2n_R \leq 2n_{r(y_1)}$ and that

$$n_1 m + \sum_{r=2}^{R} n_r\mathbb{P}({}^cB_r)m \leq m(n_1 + \sum_{r\in\mathbb{N}} n_r\mathbb{P}({}^cB_r)) \leq \tilde{C}$$

for some constant $\tilde{C} > 0$ independent from $\delta$ (using Equation (24)). Furthermore,

$$\begin{aligned}
\sum_{r=2}^{R} 2n_r\sum_{y\in\mathcal{Y}} p(y)1_{\{r\leq r(y)\}}\left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil &\leq \left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil \sum_{y\in\mathcal{Y}} p(y)\sum_{r=2}^{r(y)} 2n_r \\
&\leq \left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil \sum_{y\in\mathcal{Y}} p(y)4n_{r(y)}.
\end{aligned}$$

Combining those results and using Lemma (18), we finally see that

$$(1-\delta)\mathbb{E}[T] \leq \tilde{C} + 2n_{r(y_1)} + \left\lceil \log_2\left(\frac{10}{p(y_1)}\right)\right\rceil \sum_{y\in\mathcal{Y}} p(y)4n_{r(y_1)}$$

$$\leq \tilde{C} + 432 \frac{1}{(p(y_1) - p(y))^2} p(y_1) \ln(1/\delta_r)$$
$$+ 864 \sum_{y \in \mathcal{Y} \text{ s.t. } p(y) < p(y_1)/2} p(y) \frac{4}{p(y_1)^2} p(y_1) \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil \ln(1/\delta_r)$$
$$+ 864 \sum_{y \in \mathcal{Y} \text{ s.t. } p(y) \geq p(y_1)/2} p(y) \frac{1}{(p(y_1) - p(y))^2} p(y_1) \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil \ln(1/\delta_r),$$

hence that

$$\mathbb{E}[T] \leq 432 \frac{1}{(p(y_1) - p(y_2))^2} p(y_1) \ln(1/\delta)$$
$$+ 864 \sum_{y \in \mathcal{Y} \text{ s.t. } p(y) < p(y_1)/2} p(y) \frac{4}{p(y_1)^2} p(y_1) \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil \ln(1/\delta)$$
$$+ 864 \sum_{y \in \mathcal{Y} \text{ s.t. } p(y) \geq p(y_1)/2} p(y) \frac{1}{(p(y_1) - p(y))^2} p(y_1) \left\lceil \log_2 \left( \frac{10}{p(y_1)} \right) \right\rceil \ln(1/\delta) + o(\ln(1/\delta)).$$

This ends the proof of the lemma. □

**Acknowledgments.** The authors would like to thank Gilles Blanchard, Remi Jezequiel, Marc Jourdan, Nadav Merlis, and Karen Ullrich for fruitful discussions.

## REFERENCES


AUDIBERT, J.-Y., MUNOS, R. and SZEPESVÁRI, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*.

AUER, P., CESA-BIANCHI, N., FREUND, Y. and SCHAPIRE, R. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Annual Symposium on Foundations of Computer Science*.

BRAVERMAN, M., MAO, J. and PERES, Y. (2019). Sorted Top-k in Rounds. In *COLT*.

BUBECK, S., MUNOS, R. and STOLTZ, G. (2010). Pure Exploration for Multi-Armed Bandit Problems. In *COLT*.

CABANNES, V., BACH, F., PERCHET, V. and RUDI, A. (2022). Active Labeling: Streaming Stochastic Gradients. In *NeurIPS*.

CAPPÉ, O., GARIVIER, A., MAILLARD, O.-A., MUNOS, R. and STOLTZ, G. (2013). Kullback–Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*.

CESA-BIANCHI, N., CESARI, T. and PERCHET, V. (2019). Dynamic Pricing with Finitely Many Unknown Valuations. In *ALT*.

CESA-BIANCHI, N., GENTILE, C. and ZANIBONI, L. (2006). Incremental Algorithms for Hierarchical Classification. *Journal of Machine Learning Research*.

CHEN, W., WANG, Y., YUAN, Y. and WANG, Q. (2016). Combinatorial Multi-Armed Bandit and Its Extension to Probabilistically Triggered Arms. *Journal of Machine Learning Research*.

COVER, T. and THOMAS, J. (1991). *Elements of Information Theory*. Wiley.

CRAMÉR, H. (1938). Sur un nouveau théorème-limite de la théorie des probabilités. *Actual Sci Ind. Colloque consacré à la théorie des probabilités*.

CSISZÁR, I. and KÖRNER, J. (2011). *Information Theory: Coding Theorems for Discrete Memoryless Systems*, 2 ed. Cambridge University Press.

CUCKER, F. and SMALE, S. (2002). On the Mathematical Foundations of Learning. *Bulletin of the American Mathematical Society*.

DEMPSTER, A., LAIRD, N. and RUBIN, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39** 1-38.

DINWOODIE, I. (1992). Mesures dominantes et théorème de Sanov. *Annales de l'Institut Henri Poincare*.

EVEN-DAR, E., MANNOR, S. and MANSOUR, Y. (2006). Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*.

FIEZ, T., JAIN, L., JAMIESON, K. and RATLIFF, L. (2019). Sequential Experimental Design for Transductive Linear Bandits. In *NeurIPS*.

GANGAPUTRA, S. and GEMAN, D. (2006). A Design Principle for Coarse-to-Fine Classification. In *Conference on Computer Vision and Pattern Recognition*.

GRÜNWALD, P. (2007). *The minimum description length principle*. MIT press.

HOEFFDING, W. (1963). Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association* **58** 13-30.

HUFFMAN, D. (1952). A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the Institute of Radio Engineers* **40** 1098–1101.

JAMIESON, K. and NOWAK, R. (2011). Active Ranking using Pairwise Comparisons. In *NeurIPS*.

KORF, R. (1998). A complete anytime algorithm for number partitioning. *Artificial Intelligence* **106** 181–203.

MATHEWS, G. (1896). On the Partition of Numbers. *Proceedings of the London Mathematical Society* **28** 486-490.

OPENAI (2023). GPT-4 Technical Report Technical Report, OpenAI.

ROBBINS, H. (1955). A Remark on Stirling's Formula. *The American Mathematical Monthly*.

SANOV, I. N. (1957). On the probability of large deviations of random variables. *Matematicheskii Sbornik* **42** 11-44.

SHANNON, C. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal* **27** 379–423.

TOUVRON, H., LAVRIL, T., IZACARD, G., MARTINET, X., LACHAUX, M.-A., LACROIX, T., ROZIÈRE, B., GOYAL, N., HAMBRO, E., AZHAR, F., RODRIGUEZ, A., JOULIN, A., GRAVE, E. and LAMPLE, G. (2023). LLaMA: Open and Efficient Foundation Language Models Technical Report, Meta.

VALIANT, L. (1984). A theory of the learnable. *Communications of the ACM*.

VITTER, J. (1987). Design and Analysis of Dynamic Huffman Codes. *Journal of the ACM* **34** 825–845.

ZHU, B., JORDAN, M. and JIAO, J. (2023). Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons. In *ICML*.

ZIEGLER, D., STIENNON, N., WU, J., BROWN, T., RADFORD, A., AMODEI, D., CHRISTIANO, P. and IRVING, G. (2020). Fine-Tuning Language Models from Human Preferences Technical Report, OpenAI.