Sequential Mode Estimation with Oracle Queries

Dhruti Shah,[1] **Tuhinangshu Choudhury,**[1] **Nikhil Karamchandani,**[1*] **Aditya Gopalan**[2]
[1]Indian Institute of Technology, Bombay
[2]Indian Institute of Science, Bangalore
{dhruti96shah, choudhurytuhinangshu}@gmail.com,
nikhilk@ee.iitb.ac.in, aditya@iisc.ac.in

Abstract

We consider the problem of adaptively PAC-learning a probability distribution $\mathcal{P}$'s mode by querying an oracle for information about a sequence of i.i.d. samples $X_1, X_2, \ldots$ generated from $\mathcal{P}$. We consider two different query models: (a) each query is an index i for which the oracle reveals the value of the sample X_i, (b) each query is comprised of two indices i and j for which the oracle reveals if the samples X_i and X_j are the same or not. For these query models, we give sequential mode-estimation algorithms which, at each time t, either make a query to the corresponding oracle based on past observations, or decide to stop and output an estimate for the distribution's mode, required to be correct with a specified confidence. We analyze the query complexity of these algorithms for any underlying distribution $\mathcal{P}$, and derive corresponding lower bounds on the optimal query complexity under the two querying models.

1 Introduction

Estimating the most likely outcome of a probability distribution is a useful primitive in many computing applications such as counting, natural language processing, clustering, etc. We study the probably approximately correct (PAC) sequential version of this problem in which the learner faces a stream of elements sampled independently and identically distributed (i.i.d.) from an unknown probability distribution, wishing to learn an element with the highest probability mass on it (a mode) with confidence. At any time, the learner can issue queries to obtain information about the identities of samples in the stream, and aims to use as few queries as possible to learn the distribution's mode with high confidence. Specifically, we consider two natural models for sample identity queries – (a) each query, for a single sample of the stream so far, unambiguously reveals the identity (*label*) of the sample, (b) each query, for a pair of samples in the stream, reveals whether they are the same element or not.

A concrete application of mode estimation (and one of the main reasons that led to this formulation) is the problem of adaptive, *partial* clustering, where the objective is to find the largest cluster (i.e., equivalence class) of elements as opposed to learning the entire cluster grouping (Mazumdar and Saha 2017a; 2017b; 2017c; 2016). We are given a set of elements with an unknown clustering or partition, and would like to find the elements comprising the largest cluster or partition. Suppose a stream of elements is sampled uniformly and independently from the set, and at each time one can ask a *comparison oracle* questions of the form: "Do two sampled elements u and v belong to the same cluster or not?" Under this uniformly sampled distribution for the element stream, the probability of an element belonging to a certain cluster is simply proportional to the cluster's size, so learning the heaviest cluster is akin to identifying the mode of the distribution of a sampled element's cluster label.

We make the following contributions towards understanding the sequential query complexity for estimating the mode of a distribution using a stream of samples. (a) For both the individual-label and pairwise similarity query models, we give sequential PAC query algorithms which provably output a mode of the sample-generating distribution with large probability, together with guarantees on the number of queries they issue. These query complexity upper bounds explicitly depend on parameters of the unknown discrete probability distribution, in that they scale inversely with the gap between the probability masses at the mode and at the other elements in the distribution's support. The proposed algorithms exploit the probabilistic i.i.d. structure of the data stream to resolve uncertainty about the mode in a query-efficient fashion, and are based on the upper and lower confidence bounds (UCB, LCB) principle from online learning to guide adaptive exploration across time; in fact, we employ more refined empirical Bernstein bounds (Maurer and Pontil 2009) to take better advantage of the exact structure of the unknown sample distribution. (b) We derive fundamental limits on the query complexity of any sequential mode-finding algorithm for both query models, whose constituents resemble those of the query complexity upper bounds for our specific query algorithms above. This indicates that the algorithms proposed make good use of their queries and the associated information in converging upon a mode estimate. (c) We report numerical simulation results that support our theoretical query complexity performance bounds.

*N. Karamchandani's research was supported in part by Indo-French grant No. IFC/DST-Inria-2016-01/448 "Machine Learning for Network Analytics"

1.1 Related Work

The mode estimation problem has been studied classically in the batch or non-sequential setting since many decades back, dating to the work of Parzen (Parzen 1962) and Chernoff (Chernoff 1964), among others. This line of work, however, focuses on the objective of consistent mode estimation (and the asymptotic distribution of the estimate) for continuous distributions, instead of finite-time PAC guarantees for large-support discrete distributions as considered here. Our problem is essentially a version of sequential composite hypothesis testing with adaptive "actions" or queries, and with an explicit high-confidence requirement on the testing algorithm upon stopping.

There has been a significant amount of work in the streaming algorithms community, within computer science, on the "heavy hitter" problem – detecting the most frequent symbol in an arbitrary (non-stochastic) sequence – and generalizations thereof pertaining to estimation of the empirical moments, see e.g., (Misra and Gries 1982), (Karp, Shenker, and Papadimitriou 2003), (Manku and Motwani 2002). However the focus here is on understanding resource limits, such as memory and computational effort, on computing on arbitrary (non stochastic / unstructured) streams that arise in highly dynamic network applications. We are instead interested in quantifying the *statistical* efficiency of mode estimation algorithms in terms of the structure of the generating probability distribution.

Adaptive decision making and resolution of the explore-exploit trade off is the subject of work on the well-known multi-armed bandit model, e.g., (Bubeck, Cesa-Bianchi, and others 2012). At an abstract level, our problem of PAC-mode estimation is like a multi-armed bandit "best arm" identification problem (Kaufmann, Cappé, and Garivier 2016) but with a different information structure – queries are not directly related to any utility structure for rewards as in bandits.

Perhaps the closest work in spirit to ours is the recent work by Mazumdar and co-authors (Mazumdar and Saha 2017a; 2017b; 2017c; 2016), where the aim is to learn the entire structure of an unknown clustering by making information queries. In this regard, studying the mode estimation problem helps to shed light on the simpler, and often more natural, objective of merely identifying the largest cluster in many machine learning applications, which has not been addressed by previous work.

2 Problem formulation

In this section we develop the required notation and describe the query models.

Consider an underlying unknown discrete probability distribution $\mathcal{P}$ with the support set $\{1, 2, ...k\}$. For each $i \in \{1, 2, \ldots, k\}$ and a random variable $X \sim \mathcal{P}$, let $Pr(X = i) = p_i(\mathcal{P}) \equiv p_i$.

We would like to estimate the *mode* of the unknown distribution $\mathcal{P}$, defined as any member of the set[1] $\arg\max_{1 \le i \le k} p_i$. Towards this, we assume query access to an *oracle* containing a sequence of independently and identically distributed (i.i.d.) samples from $\mathcal{P}$, denoted $X_1, X_2, \ldots$ We study the mode estimation problem under the following query models to access the values of these i.i.d. samples:

1. **Query Model 1 (QM1) :** For each query, we specify an index $i \ge 1$ following which the oracle reveals the value of the sample X_i to us. Since the samples are i.i.d., without loss of generality, we will assume that the t^{th} successive query reveals X_t, $t = 1, 2, \ldots$
2. **Query Model 2 (QM2) :** In this model, the oracle answers *pairwise similarity* queries. For each query, we specify *two* indices $i, j \in \{1, 2, \ldots\}$, following which the oracle reveals if the two samples X_i and X_j are equal or not. Formally, the response of the oracle to a query (i, j) is
$$\mathcal{O}(i,j) = \begin{cases} +1 & \text{if } X_i = X_j, \\ -1 & \text{otherwise.} \end{cases}$$
Note that to know the value of a sample X_i in this query model, multiple pair-wise queries to the oracle might be required.

For each of the query models above, our goal is to design a statistically efficient sequential mode-estimation algorithm which, at each time t, either makes a query to the oracle based on past observations or decides to stop and output an estimate for the distribution's mode. Mathematically, a sequential algorithm with a stopping rule decides an action A_t at each time $t \ge 1$ depending only on past observations. For QM1, A_t can be one of the following:

- *(continue,t)*: Query the index t,
- *(stop,$\hat{m}$)*, $\hat{m} \in \{1, \ldots, k\}$: Stop querying and return $\hat{m}$ as the mode estimate.

For QM2, A_t can be one of the following:

- *(continue,t)*: Continue with the next round, with possibly multiple sequential pairwise queries of the form (t, j) for some $j < t$. That is, we compare the sample X_t with some or all of the previous samples.
- *(stop,$\hat{m}$)*, $\hat{m} \in \{1, \ldots, k\}$: Stop querying and return $\hat{m}$ as the mode estimate.

The stopping time of the algorithm is defined as
$$\tau := \inf\{t \ge 1 : A_t = (stop, .)\}.$$

The cost of the algorithm is measured by its *query complexity* – the number of queries made by it before stopping. For $\delta > 0$, a sequential mode-estimation algorithm is defined to be a **δ-true mode estimator** if it correctly identifies the mode for every distribution $\mathcal{P}$ on the support set $\{1, 2, \ldots, k\}$ with probability at least $1 - \delta$, i.e., $\mathbb{P}_{\mathcal{P}}[\hat{m} \in \arg\max_{1 \le i \le k} p_i(\mathcal{P})] \ge 1 - \delta$. The goal is to obtain δ-true mode estimators for each query model (QM1 and QM2) that require as few queries as possible. For a δ-true mode estimator $\mathcal{A}$ and a distribution $\mathcal{P}$, let $Q_\delta^{\mathcal{P}}(\mathcal{A})$ denote the number of queries made by a δ-true mode estimator when the underlying unknown distribution is $\mathcal{P}$. We are interested in studying the optimal query complexity of δ-true

[1] $\arg\max_{i \in S} p_i$ is used to denote the set of all maximisers of the function $i \to p_i$ on S.

mode estimators. Note that $Q_\delta^{\mathcal{P}}(\mathcal{A})$ is itself a random quantity, and our results either hold in expectation or with high probability.

For the purpose of this paper, we assume that $p_1 > p_2 \geq .... \geq p_k$ i.e. the mode of the underlying distribution is 1, and hence a δ-true mode estimator returns 1 with probability at least $(1-\delta)$. In Sections 3 and 4, we discuss δ-true mode estimators and analyze their query complexity for the QM1 and QM2 query models respectively. We provide some experimental results in Section 5 and further explore a few variations of the problem in Section 6. Several proofs have been relegated to the full version of our paper (Shah et al. 2019).

3 Mode estimation with QM1

We will begin by presenting an algorithm for mode estimation under QM1 and analyzing its query complexity.

3.1 Algorithm

Recall that under the QM1 query model, querying the index t to the oracle reveals the value of the corresponding sample, X_t, generated i.i.d. according to the underlying unknown distribution $\mathcal{P}$. During the course of the algorithm, we form bins for each element i in the support $\{1, 2, \ldots, k\}$ of the underlying distribution. Bin j is created when the first sample with value j is revealed and any further samples with that value are 'placed' in the same bin. For each query t and revealed sample value X_t, define Z_t^i for $i \in \{1, 2, ...k\}$ as follows.

$$Z_t^i = \begin{cases} 1 & \text{if } X_t = i \\ 0 & \text{otherwise.} \end{cases}$$

Note that for each given i, t, Z_t^i is a Bernoulli random variable with $E[Z_t^i] = p_i$. Also for any given i, $\{Z_t^i\}$ are i.i.d. over time.

Our mode estimation scheme is presented in Algorithm 1. At each stage of the algorithm, we maintain an empirical estimate of the probability of bin i, p_i, for each $i \in \{1, 2, \ldots, k\}$. Let $\hat{p}_i^t$ denote the estimate at time t, given by

$$\hat{p}_i^t = \frac{\sum_{j=1}^t Z_j^i}{t}, \tag{1}$$

where recall that Z_j^i for $j = 1, 2....t$ are the t i.i.d. samples. Also, at each time instant, we maintain confidence bounds for the estimate of p_i. The confidence interval for the i^{th} bin probability at the t^{th} iteration is denoted by β_i^t, and it captures the deviation of the empirical value $\hat{p}_i^t$ from its true value p_i. In particular, the confidence interval value β_i^t is chosen so that the true value p_i lies in the interval $[\hat{p}_i^t - \beta_i^t, \hat{p}_i^t + \beta_i^t]$ with a significantly large probability. The lower and upper boundaries of this interval are referred to as the lower confidence bound (LCB) and the upper confidence bound (UCB) respectively. The particular choice for the value of β_i^t that we use for our algorithm is presented in Section 3.2.

Finally, our stopping rule is as follows : A_t = *(stop, i)* when there exists a bin $i \in \{1, 2, \ldots, k\}$ whose LCB is greater than the UCB of all the other bins, upon which the index i is output as the mode estimate.

Given the way our confidence intervals are defined, this ensures that the output of the estimator is the mode of the underlying distribution with large probability.

Algorithm 1 Mode estimation algorithm under QM1

$t = 1$
A_0 = *(continue,1)* : Obtain X_1.
loop
  if a bin with value X_t already exists **then**
    Add query index t to the corresponding bin.
  else
    Create a new bin with value X_t.
  end if
  Update the empirical estimate $\hat{p}_i^t$ (1) and confidence interval β_i^t (2) for all bins $i \in \{1, 2, \ldots, k\}$.

$$A_t = \begin{cases} \textit{(stop,i)}: & \text{if } \exists i, \text{ s.t. } \forall j \neq i \\ \text{Exit, Output } i & \hat{p}_i^t - \beta_i^t > \hat{p}_j^t + \beta_j(t), \\ \textit{(continue,t+1)}: & \text{otherwise} \\ \text{Obtain } X_{t+1} \end{cases}$$

  $t = t + 1$
end loop

3.2 Analysis

Theorem 1. *For the following choice of β_i^t, Algorithm 1 is a δ-true mode estimator.*

$$\beta_i^t = \sqrt{\frac{2V_t(Z^i)\log(4kt^2/\delta)}{t}} + \frac{7\log(4kt^2/\delta)}{3(t-1)}, \tag{2}$$

where $V_t(Z^i) = \dfrac{1}{t(t-1)} \displaystyle\sum_{1 \leq p < q \leq t} (Z_p^i - Z_q^i)^2$ *is the empirical variance.*

Proof. This proof is based on confidence bound arguments. To construct the confidence intervals for the probability values $\{p_i\}$'s, we use the empirical version of the Bernstein bound given in (Maurer and Pontil 2009). The result used is stated as Theorem 9 in (Shah et al. 2019, Appendix A). Using the result in our context, we get the following for any given pair (i, t) with probability at least $(1 - \delta_1)$:

$$|p_i - \hat{p}_i^t| \leq \sqrt{\frac{2V_t(Z^i)\log(2/\delta_1)}{t}} + \frac{7\log(2/\delta_1)}{3(t-1)}, \tag{3}$$

where $V_t(Z^i)$ is the sample variance, i.e., $V_t(Z^i) = \dfrac{1}{t(t-1)} \displaystyle\sum_{1 \leq p < q \leq t} (Z_p^i - Z_q^i)^2$.

To establish confidence bounds on the sample variance, we use the result given in (Maurer and Pontil 2009), which is stated as Theorem 10 in (Shah et al. 2019, Appendix A). Using the result in our context, and noting that the expected value of $V_t(Z^i)$ would be $p_i(1-p_i)$, we get the following for any given pair (i, t) with probability at least $(1 - \delta_2)$:

$$|\sqrt{p_i(1-p_i)} - \sqrt{V_t(Z^i)}| \leq \sqrt{\frac{2\log(1/\delta_2)}{t-1}}. \tag{4}$$

Let $\mathcal{E}_1$ denote the error event that for some pair (i,t) the confidence bound (3) around $\hat{p}_i^t$ is violated. Also, let $\mathcal{E}_2$ denote the error event that for some pair (i,t) the confidence bound (4) around the sample variance $V_t(Z^i)$ is violated. Choosing $\delta_1 = \delta_2 = \frac{\delta}{2kt^2}$, and taking the union bound over all i,t, from (3) and (4), we get $\mathbb{P}[\mathcal{E}_1] \leq \delta/2$ and $\mathbb{P}[\mathcal{E}_2] \leq \delta/2$. Hence we get that

$$\mathbb{P}[\mathcal{E}_1^c \cap \mathcal{E}_2^c] \geq 1 - \delta. \tag{5}$$

This means that with probability at least $1-\delta$ the confidence bounds corresponding to both equations (3) and (4) hold true for all pairs (i,t).

We now show that if the event $\mathcal{E}_1^c \cap \mathcal{E}_2^c$ is true, then Algorithm 1 returns 1 as the mode. To see this, assume the contrary that the algorithm returns $i \neq 1$. Under $\mathcal{E}_1^c \cap \mathcal{E}_2^c$ the confidence intervals hold true for all pairs (i,t) and hence the stopping condition defined in Line 10, Algorithm 1 will imply

$$p_i \geq \hat{p}_i^t - \beta_i^t > \hat{p}_1^t + \beta_1^t \geq p_1$$

which is false. Thus Algorithm 1 returns 1 as the mode if the event $\mathcal{E}_1^c \cap \mathcal{E}_2^c$ is true. Hence, the probability of returning 1 as the mode is at least $\mathbb{P}[\mathcal{E}_1^c \cap \mathcal{E}_2^c]$, which by (5) implies that Algorithm 1 is a δ-true mode estimator. □

3.3 Query Complexity Upper bound

Theorem 2. *For a δ-true mode estimator $\mathcal{A}_1$, corresponding to Algorithm 1, we have the following with probability at least $(1-\delta)$.*

$$Q_\delta^{\mathcal{P}}(\mathcal{A}_1) \leq \frac{592}{3}\frac{p_1}{(p_1-p_2)^2}\log\left(\frac{592}{3}\sqrt{\frac{k}{\delta}}\frac{p_1}{(p_1-p_2)^2}\right).$$

Proof. If the event $\mathcal{E}_1^c \cap \mathcal{E}_2^c$ is true, it implies that all confidence intervals hold true. We find a value T^*, which ensures that by $t = T^*$, the confidence intervals of the bins have separated enough such that the algorithm stops. Since at each time one query is made, this value of T^* would also be an upper bound on the query complexity $Q_\delta^{\mathcal{P}}(\mathcal{A})$. The derivation of T^* has been relegated to (Shah et al. 2019, Appendix B), which gives the upper bound as stated above. □

3.4 Query Complexity Lower bound

Theorem 3. *For any δ-true mode estimator $\mathcal{A}$, we have,*

$$\mathbb{E}[Q_\delta^{\mathcal{P}}(\mathcal{A})] \geq \frac{p_1}{(p_1-p_2)^2}\log(1/2.4\delta).$$

Proof. Let $x(\mathcal{P}) = \arg\max_{i\in[k]} p_i$ denote the mode of the distribution $\mathcal{P}$. By assumption, we have $x(\mathcal{P}) = 1$. Consider any $\mathcal{P}'$ such that $x(\mathcal{P}') \neq x(\mathcal{P}) = 1$. Let $\mathcal{A}$ be a δ-true mode estimator and let $\hat{m}$ be its output. Then, by definition,

$$\begin{aligned}\mathcal{P}(\hat{m}=1) &\geq 1-\delta \\ \mathcal{P}'(\hat{m}=1) &\leq \delta.\end{aligned} \tag{6}$$

Let τ be the stopping time associated with estimator $\mathcal{A}$. Using Wald's lemma (Wald 1944) we have,

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{\tau}\log\frac{p(X_t)}{p'(X_t)}\right] = \mathbb{E}_{\mathcal{P}}[\tau].\mathbb{E}_{\mathcal{P}}\left[\log\frac{p(X_1)}{p'(X_1)}\right]$$
$$= \mathbb{E}_{\mathcal{P}}[\tau].D(\mathcal{P}||\mathcal{P}'). \tag{7}$$

where $D(\mathcal{P}||\mathcal{P}')$ refers to the Kullback-Leibler (KL) divergence between the distributions $\mathcal{P}$ and $\mathcal{P}'$.
Also,

$$\begin{aligned}&\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{\tau}\log\frac{p(X_t)}{p'(X_t)}\right] \\ &= \mathbb{E}_{\mathcal{P}}\left[\log\frac{p(X_1,X_2....X_\tau)}{p'(X_1,X_2....X_\tau)}\right] \\ &= D(p(X_1,....X_\tau)||p'(X_1,....X_\tau)) \\ &\geq D(Ber(\mathcal{P}(\hat{m}=1))||Ber(\mathcal{P}'(\hat{m}=1))),\end{aligned}$$

where $Ber(x)$ denotes the Bernoulli distribution with parameter $x \in (0,1)$ and the last step is obtained by the data processing inequality (Cover and Thomas 2012). Furthermore we have,

$$D(Ber(\mathcal{P}(\hat{m}=1))||Ber(\mathcal{P}'(\hat{m}=1))) \geq \log(1/2.4\delta),$$

where the above follows from (6) and (Kaufmann, Cappé, and Garivier 2016, Remark 2). Finally, combining with (7) and noting that the argument above works for any $\mathcal{P}'$ such that $x(\mathcal{P}) \neq x(\mathcal{P}')$, we have

$$\mathbb{E}_{\mathcal{P}}[\tau] \geq \frac{\log(1/2.4\delta)}{\inf_{\mathcal{P}':x(\mathcal{P}')\neq x(\mathcal{P})} D(\mathcal{P}||\mathcal{P}')}. \tag{8}$$

We choose $\mathcal{P}'$ as follows, for some small $\epsilon > 0$:

$$\begin{aligned}p_i' &= p_i, \ \forall i > 2 \\ p_1' &= \frac{p_1+p_2}{2} - \epsilon = q - \epsilon, \text{ where } q = \frac{p_1+p_2}{2} \\ p_2' &= \frac{p_1+p_2}{2} + \epsilon = q + \epsilon\end{aligned}$$

We have

$$\begin{aligned}D(\mathcal{P}||\mathcal{P}') &= p_1\log\left(\frac{p_1}{q-\epsilon}\right) + p_2\log\left(\frac{p_2}{q+\epsilon}\right) \\ &= (p_1+p_2)\left[\frac{p_1}{p_1+p_2}\log\left(\frac{\frac{p_1}{p_1+p_2}}{\frac{q-\epsilon}{p_1+p_2}}\right) + \right. \\ &\qquad \left. \frac{p_2}{p_1+p_2}\log\left(\frac{\frac{p_2}{p_1+p_2}}{\frac{q+\epsilon}{p_1+p_2}}\right)\right] \\ &= (p_1+p_2)D\left(\frac{p_1}{p_1+p_2}||\frac{q-\epsilon}{p_1+p_2}\right) \\ &\overset{(a)}{\leq} (p_1+p_2)\left[\frac{(p_1-q+\epsilon)^2}{(q-\epsilon)\cdot(q+\epsilon)}\right] \\ &= (p_1+p_2)\left[\frac{(p_1-p_2+2\epsilon)^2}{(p_1+p_2+2\epsilon)\cdot(p_1+p_2-2\epsilon)}\right] \\ &\overset{(b)}{\approx} \frac{(p_1-p_2)^2}{(p_1+p_2)} \\ &\leq \frac{(p_1-p_2)^2}{p_1},\end{aligned} \tag{9}$$

where (a) follows from (Popescu et al. 2016, Theorem 1.4) which gives an upper bound on the KL divergence and (b)

follows because ϵ can be arbitrarily small. Note that an upper bound on the stopping time τ would also give an upper bound on $Q_\delta^{\mathcal{P}}(\mathcal{A})$, since we have one query at each time. Hence, using (8) and (9), we get the following lower bound:

$$\mathbb{E}[Q_\delta^{\mathcal{P}}(\mathcal{A})] \geq \frac{p_1}{(p_1 - p_2)^2} \log(1/2.4\delta).$$

□

4 Mode estimation with QM2

In this section, we present an algorithm for mode estimation under QM2 and analyse its query complexity.

4.1 Algorithm

Recall that under the QM2 query model, a query to the oracle with indices i and j reveals whether the samples X_i and X_j have the same value or not. Our mode estimation scheme is presented in Algorithm 2. During the course of the algorithm we form bins, and the i^{th} bin formed is referred to as bin i. Here, each bin is a collection of indices having the same value as revealed by the oracle. Let $\sigma(i)$ denote the element from the support that that i^{th} bin represents. The algorithm proceeds in rounds. In round t, let $b(t)$ denote the number of bins present. Based on the observations made so far, we form a subset of bins $\mathcal{C}(t)$. We go over the bins in $\mathcal{C}(t)$ one by one, and in each iteration within the round we query the oracle with index t and a sample index j_l from some bin l, i.e., $j_l \in$ bin l for $l \in \mathcal{C}(t)$. The round ends when the oracle gives a positive answer to one of the queries or we exhaust all bins in $\mathcal{C}(t)$. So, the number of queries in round t is at most $|\mathcal{C}(t)| \leq b(t)$. If we get a positive answer from the oracle, we add t to the corresponding bin. A new bin will be created if the oracle gives a negative answer to each of these queries.

We will now describe how we construct the subset $\mathcal{C}(t)$ of bins we compare against in round t. To do so, for each bin i, corresponding to $\sigma(i)$, and for each round t, we maintain an empirical estimate of each $p_{\sigma(i)}$ during each round, denoted by $\hat{p}^t_{\sigma(i)}$. We also maintain confidence intervals for each $p_{\sigma(i)}$, denoted by $\beta^t_{\sigma(i)}$. The choice for $\beta^t_{\sigma(i)}$ is the same as that in QM1, given in (2). $\mathcal{C}(t)$ is formed in each round t by considering only those bins whose UCB is greater than the LCB of all other bins. Mathematically,

$$\mathcal{C}(t) = \{i \in \{1, 2, \ldots, b(t)\} \text{ such that } \nexists l \text{ for which } \hat{p}^t_l - \beta^t_l > \hat{p}^t_{\sigma(i)} + \beta^t_{\sigma(i)}\} \quad (10)$$

The rest of the algorithm is similar to Algorithm 1, including the stopping rule, i.e. we stop when there is a bin whose LCB is greater than the UCB of all other bins in $\mathcal{C}(t)$. As before, the choice of confidence intervals and the stopping rule ensures that when $A_t =$ *(stop, i)*, the corresponding element $\sigma(i) = 1$ with probability at least $(1 - \delta)$.

4.2 Analysis

Theorem 4. *For the choice of β^t_i as given by* (2)*, Algorithm 2 is a δ-true mode estimator.*

Algorithm 2 Mode estimation algorithm under QM2

$t = 1$
$A_0 =$ *(continue,1)* : Obtain X_1.
loop
  Form $\mathcal{C}(t)$ according to (10).
  flag=0.
  for all bins $l \in \mathcal{C}(t)$ **do**
    Obtain $j_l \in$ bin i .
    if $\mathcal{O}(t, j_l) == +1$ **then**
      Add t to the corresponding bin.
      flag=1; *BREAK*
    end if
  end for
  if flag==0 **then**
    Create a new bin for index t.
  end if
  Update the empirical estimate $\hat{p}^t_i$ (1) and confidence interval β^t_i (2) for all bins $i \in \{1, 2, \ldots, k\}$.

$$A_t = \begin{cases} \textit{(stop,i)}: & \text{if } \exists i, \text{ s.t. } \forall j \in \mathcal{C}(t) \\ \text{Exit, Output } i & \hat{p}^t_i - \beta^t_i > \hat{p}^t_j + \beta_j(t), \\ \textit{(continue,t+1)}: & \text{otherwise} \\ \text{Move to next round} & \end{cases}$$

  $t = t + 1$
end loop

Proof. The error analysis for Algorithm 2 is similar as that for Algorithm 1 as given in Section 3.2. We consider the same two events $\mathcal{E}_1$ and $\mathcal{E}_2$. Recall that $\mathcal{E}_1$ denotes the error event that for some pair (i, t) the confidence bound (3) around $\hat{p}^t_i$ is violated; $\mathcal{E}_2$ denotes the error event that for some pair (i, t) the confidence bound (4) around the sample variance $V_t(Z^i)$ is violated. Again choosing similar values for δ_1 and δ_2, we get $\mathbb{P}[\mathcal{E}_1^c \cap \mathcal{E}_2^c] \geq 1 - \delta$. We now need to show that if the event $\mathcal{E}_1^c \cap \mathcal{E}_2^c$ is true, then Algorithm 2 returns 1 as the mode. This then implies that Algorithm 2 is a δ-true mode estimator.

The analysis remains same as discussed for QM1. The only additional factor we need to consider here, is the event that the bin corresponding to the mode 1 of the underlying distribution is discarded in one of the rounds of the algorithm, i.e., it is not a part of $\mathcal{C}(t)$ for some round t. A bin is discarded when its UCB becomes less than the LCB of any other bin. Under event $\mathcal{E}_1^c$, all confidence intervals are true, and since $p_1 > p_i, \forall i$ the UCB of the corresponding bin can never be less than the LCB of any other bin. This implies that under $\mathcal{E}_1^c$, the bin corresponding to the mode 1 is never discarded. Hence, the probability of returning 1 as the mode is at least $\mathbb{P}[\mathcal{E}_1^c \cap \mathcal{E}_2^c]$, which by (5) implies that Algorithm 2 is a δ-true mode estimator. □

4.3 Query Complexity Upper bound

From the analysis of the sample complexity for the QM1 model derived in Section 3.3, we get one upper bound for the QM2 case. Algorithm 1 continues for at most T^* rounds

with probability at least $1-\delta$, where T^* is given by Theorem 2. A sample accessed in each of these rounds can be compared to at most $\min\{k, T^*\}$ other samples. Thus, a natural upper bound on the query complexity of Algorithm 2 is $(T^* \cdot \min\{k, T^*\})$. The following result provides a tighter upper bound.

Theorem 5. *For a δ-true mode estimator $\mathcal{A}_2$, corresponding to Algorithm 2, we have the following with probability at least $(1-\delta)$.*

$$Q_\delta^{\mathcal{P}}(\mathcal{A}_2) \leq \frac{592}{3}\frac{p_1}{(p_1-p_2)^2}\log\left(\frac{592}{3}\sqrt{\frac{k}{\delta}}\frac{p_1}{(p_1-p_2)^2}\right) + \sum_{i=2}^{T}\frac{592}{3}\frac{p_1}{(p_1-p_i)^2}\log\left(\frac{592}{3}\sqrt{\frac{k}{\delta}}\frac{p_1}{(p_1-p_i)^2}\right)$$

$$\textit{for } T = \min\left\{k, \tfrac{592}{3}\tfrac{p_1}{(p_1-p_2)^2}\log\left(\tfrac{592}{3}\sqrt{\tfrac{k}{\delta}}\tfrac{p_1}{(p_1-p_2)^2}\right)\right\}.$$

Proof. The detailed calculations are provided in (Shah et al. 2019, Appendix B), here we give a sketch. Under the event $\mathcal{E}_1^c \cap \mathcal{E}_2^c$, where all confidence bounds hold true, for any bin represented during the run of the algorithm and corresponding to some element $i \neq 1$ from the support, we have that it will definitely be excluded from $\mathcal{C}(t)$ when its confidence bound $\beta_i^t < \frac{p_1-p_i}{4}$ and $\beta_1^t < \frac{p_1-p_i}{4}$. We find a t which satisfies the stopping condition for each of the bins represented, and summing over them gives the total query complexity. Following the calculations in (Shah et al. 2019, Appendix B), we get the following value of t_i^* for the bin corresponding to element $i \neq 1$, such that by $t = t_i^*$ the bin will definitely be excluded from $\mathcal{C}(t)$.

$$t_i^* = \frac{592}{3}\frac{p_1}{(p_1-p_i)^2}\log\left(\frac{592}{3}\sqrt{\frac{2k}{\delta}}\frac{p_1}{(p_1-p_i)^2}\right)$$

Also, for the first bin we get t_1^* as follows.

$$t_1^* = \frac{592}{3}\frac{p_1}{(p_1-p_2)^2}\log\left(\frac{592}{3}\sqrt{\frac{2k}{\delta}}\frac{p_1}{(p_1-p_2)^2}\right)$$

Also, the total number of bins created will be at most $T = \min\left\{k, \tfrac{592}{3}\tfrac{p_1}{(p_1-p_2)^2}\log\left(\tfrac{592}{3}\sqrt{\tfrac{2k}{\delta}}\tfrac{p_1}{(p_1-p_2)^2}\right)\right\}$.

A sample from the bin corresponding to element i will be involved in at most t_i^* queries. Hence the total number of queries, $Q_\delta^{\mathcal{P}}(\mathcal{A})$, is bounded as follows

$$Q_\delta^{\mathcal{P}}(\mathcal{A}) \leq \frac{592}{3}\frac{p_1}{(p_1-p_2)^2}\log\left(\frac{592}{3}\sqrt{\frac{k}{\delta}}\frac{p_1}{(p_1-p_2)^2}\right) + \sum_{i=2}^{T}\frac{592}{3}\frac{p_1}{(p_1-p_i)^2}\log\left(\frac{592}{3}\sqrt{\frac{k}{\delta}}\frac{p_1}{(p_1-p_i)^2}\right)$$

□

4.4 Query Complexity Lower bound

The following theorem gives a lower bound on the expected query complexity for the QM2 model.

Theorem 6. *For any δ-true mode estimator $\mathcal{A}$, we have,*

$$\mathbb{E}[Q_\delta^{\mathcal{P}}(\mathcal{A})] \geq \frac{p_1}{2(p_1-p_2)^2}\log(1/2.4\delta).$$

Proof. Consider any δ-true mode estimator $\mathcal{A}$ under query model QM2 and let τ denote its average (pairwise) query complexity when the underlying distribution is $\mathcal{P}$. Next, consider the QM1 query model and let estimator $\mathcal{A}'$ simply simulate the estimator $\mathcal{A}$ under QM2, by querying the values of any sample indices involved in the pairwise queries. It is easy to see that since $\mathcal{A}$ is a δ-true mode estimator under QM2, the same will be true for $\mathcal{A}'$ as well under QM1. Furthermore, the expected query complexity of $\mathcal{A}'$ under QM1 will be at most 2τ since each pairwise query involves two sample indices. Thus, if the query complexity τ of $\mathcal{A}$ under QM2 is less than $\frac{p_1}{2(p_1-p_2)^2}\log(1/2.4\delta)$, then we have a δ-true mode estimator under query model QM1 with query complexity less than $\frac{p_1}{(p_1-p_2)^2}\log(1/2.4\delta)$, which contradicts the lower bound in Theorem 3. The result then follows. □

The lower bound on the query complexity as given by the above theorem matches the first term of the upper bound given in Theorem 5. So, the lower bound will be close to the upper bound when the first term dominates, in particular when $\frac{p_1}{(p_1-p_2)^2} \gg \sum_{i=3}^{k}\frac{p_1}{(p_1-p_i)^2}$.

While the above lower bound matches the upper bound in a certain restricted regime, we would like a sharper lower bound which works more generally. Towards this goal, we consider a slightly altered (and relaxed) problem setting which relates to the problem of best arm identification in a multi-armed bandit (MAB) setting studied in (Kaufmann, Cappé, and Garivier 2016), (Soare, Lazaric, and Munos 2014). The altered setting and the corresponding result are discussed in (Shah et al. 2019, Appendix C).

5 Experimental Results

For both the QM1 and QM2 models, we simulate Algorithm 1 and Algorithm 2 for various synthetic distributions. We take $k = 5120$ and keep the difference $p_1 - p_2 = 0.208$ constant for each distribution. For the other p_i's we follow two different models:

1. Uniform distribution : The other p_i's for $i = 3....k$ are chosen such that each $p_i = \frac{1-p_1-p_2}{k-2}$.
2. Geometric distribution : The other p_i's are chosen such that $p_2, p_3....p_k$ form a decreasing geometric distribution which sums upto $1-p_1$.

For each distribution we run the experiment 50 times and take an average to plot the query complexity. In Fig. 1 we plot the number of queries taken by Algorithm 1 for QM1, for both the geometric and uniform distributions. As suggested by our theoretical results, the query complexity increases (almost) linearly with p_1 for a fixed $(p_1 - p_2)$. In

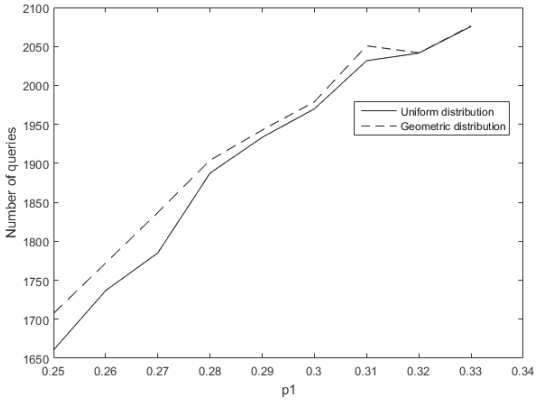

Figure 1: Number of queries for Algorithm 1 under the uniform and geometric distributions.

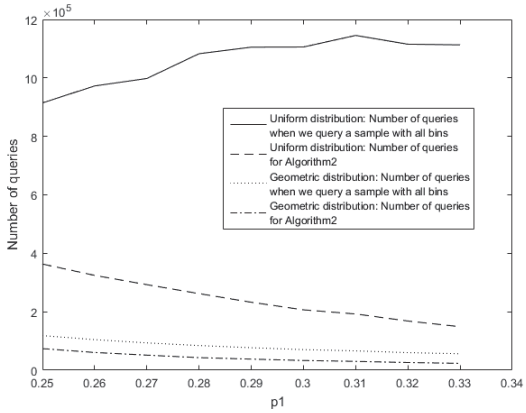

Figure 2: Number of queries when each sample is queried with all the bins, and number of queries for Algorithm 2 for the uniform and geometric distributions.

Fig. 2 we plot the number of queries taken by Algorithm 2 for QM2 and compare it to the number of queries taken by a naive algorithm which queries a sample with all the bins formed, for both the uniform and geometric distributions.
To further see how Algorithm 2 performs better than the naive algorithm which queries a sample against all bins formed, we plot the number of queries in each round for a particular geometric distribution, in Fig. 3. We observe that over the rounds, for Algorithm 2 bins start getting discarded and hence the number of queries per round decreases, while for the naive algorithm, as more and more bins are formed, the number of queries per round keeps increasing.

Real world dataset: As mentioned in the introduction, one of the applications of mode estimation is partial clustering. Via experiments on a real-world purchase data set (Leskovec and Krevl 2014), we were able to benchmark the performance of our proposed Algorithm 2 for pairwise queries, a naive variant of it with no UCB-based bin elimination and the algorithm of (Mazumdar and Saha 2017a) which performs a full clustering of the nodes. Using (Leskovec and Krevl 2014), we create a dataset where each entry represents an item available on Amazon.com along with a label corresponding to a category that the item belongs to. We take entries with a common label to represent a cluster and we consider the task of using pairwise oracle queries to identify a cluster of items within the top-10 clusters, from among the top-100 clusters in the entire dataset. Note that while our algorithm is tailored towards finding a heavy cluster, the algorithm in (Mazumdar and Saha 2017a) proceeds by first learning the entire clustering and then identifying a large cluster. Some statistics of the dataset over which we ran our algorithm are: number of clusters - 100; number of nodes - 291925; size of largest cluster - 53551; and size of 11th largest cluster - 19390.

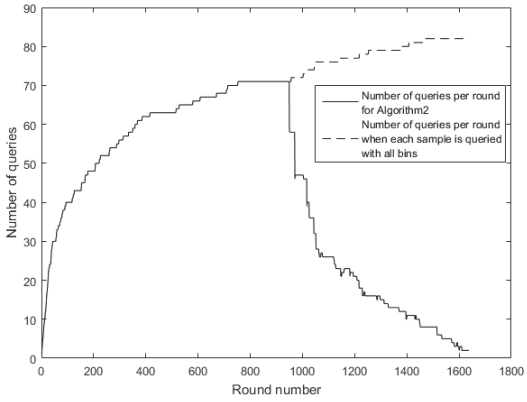

Figure 3: Number of queries per round for Algorithm 2 and for the algorithm which queries each sample with all of the bins formed.

With a target confidence of 99%, our proposed algorithm terminates in $\sim 631k$ pairwise queries while the naive algorithm takes $\sim 1474k$ queries ($2.3x$ more). The algorithm of (Mazumdar and Saha 2017a) clusters all nodes first, and is thus expected to take around $n * k = 29192.5k$ queries ($46x$ more) to terminate; this was larger than our computational budget and hence could not be run successfully. With a target confidence of 95%, our algorithm takes $\sim 558k$ queries instead of the naive version which takes $\sim 1160k$ queries ($2x$ more) to terminate.

6 Discussion

There are a few variations of the standard mode estimation problem which have important practical applications and we discuss them in the following subsections.

6.1 Top-m estimation

An extension of the mode estimation problem discussed could be estimating the top-m values. i.e. for $p_1 > p_2 > .... > p_k$, an algorithm should return the set $\{1, 2, ...m\}$. A possible application is in clustering, where we are interested in finding the largest m clusters. The algorithms 1 and 2 for

QM1 and QM2 would change only in the stopping rule. The new stopping rule would be such that A_t = *(stop,.)* when there exist m bins such that their LCB is greater than the UCB of the other remaining bins. In this setting, we define a δ-true top-m estimator as an algorithm which returns the set $\{1, 2, ...m\}$ with probability at least $(1-\delta)$. In the following results we give bounds on the query complexity, $Q_\delta^{\mathcal{P}}(\mathcal{A})$ for a δ-true top-m estimator $\mathcal{A}$, for the QM1 query model.

Theorem 7. *For a δ-true top-m estimator $\mathcal{A}_m$, corresponding to Algorithm 1 for the top-m case, we have the following with probability at least $(1-\delta)$.*

$$Q_\delta^{\mathcal{P}}(\mathcal{A}_m) \leq$$

$$\max_{\substack{i\in\{1,2....m\}\\ j\in\{m+1,....k\}}} \frac{592}{3}\frac{p_i}{(p_i-p_j)^2}\log\left(\frac{592}{3}\sqrt{\frac{k}{\delta}}\frac{p_i}{(p_i-p_j)^2}\right)$$

Proof. The proof follows along the same lines as Theorem 2. Here, for a bin $i \in \{1, 2, ...m\}$ and a bin $j \in \{m+1, ...k\}$, their confidence bounds would be separated when $\beta_i^t < \frac{p_i - p_j}{4}$ and $\beta_j^t < \frac{p_i - p_j}{4}$. The calculations then follow similarly as before to give the above upper bound. □

Theorem 8. *For any δ-true top-m estimator $\mathcal{A}$, we have,*

$$\mathbb{E}[Q_\delta^{\mathcal{P}}(\mathcal{A})] \geq \max_{\substack{i\in\{1,2....m\}\\ j\in\{m+1,....k\}}} \frac{p_i}{(p_i-p_j)^2}\log(1/2.4\delta)$$

Proof. The proof follows along the same lines as Theorem 3. Here, for $\epsilon > 0$, the alternate distribution $\mathcal{P}'$ that we choose would have $p'_i = \frac{p_i+p_j}{2} - \epsilon$ and $p'_j = \frac{p_i+p_j}{2} + \epsilon$ for some $i \in \{1, 2....m\}$ and some $j \in \{m+1, ...k\}$. We take the maximum value over all such i, j to give the lower bound. □

6.2 Noisy oracle

Here, we consider a setting where the oracle answers queries noisily and we analyze the impact of errors on the query complexity for mode estimation.

For the noisy QM1 model, we assume that when we query the oracle with some index j, the value revealed is the true sample value X_j with probability $(1-p_e)$ and any of the k values in the support with probability $\frac{p_e}{k}$ each. The problem of mode estimation in this noisy setting is equivalent to that in a noiseless setting where the underlying distribution is given by

$$p'_i = (1-p_e)p_i + \frac{p_e}{k}.$$

Since the mode of this altered distribution is the same as the true distribution, we can use Algorithm 1 for mode estimation under the noisy QM1 model, and the corresponding query complexity bounds in Section 3 hold true.

For the noisy QM2 model, we assume that for any pair of indices i and j sent to the oracle, it returns the correct answer ($+1$ if $X_i = X_j$, -1 otherwise) with probability $(1-p_e)$. This setting is technically more involved and is in fact similar to clustering using pairwise noisy queries (Mazumdar and Saha 2017a). The mode estimation problem in this setting corresponds to identifying the largest cluster, which has been studied in (Choudhury, Shah, and Karamchandani 2019). Deriving tight bounds for this case is still open.

References

Bubeck, S.; Cesa-Bianchi, N.; et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5(1):1–122.

Chernoff, H. 1964. Estimation of the mode. *Annals of the Institute of Statistical Mathematics* 16(1):31–41.

Choudhury, T.; Shah, D.; and Karamchandani, N. 2019. Top-m clustering with a noisy oracle. In *2019 National Conference on Communications (NCC) (NCC 2019)*.

Cover, T. M., and Thomas, J. A. 2012. *Elements of information theory*. John Wiley & Sons.

Karp, R. M.; Shenker, S.; and Papadimitriou, C. H. 2003. A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems (TODS)* 28(1):51–55.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research* 17(1):1–42.

Leskovec, J., and Krevl, A. 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data/com-Amazon.html.

Manku, G. S., and Motwani, R. 2002. Approximate frequency counts over data streams. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, 346–357. Elsevier.

Maurer, A., and Pontil, M. 2009. Empirical Bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*.

Mazumdar, A., and Saha, B. 2016. Clustering via crowdsourcing. *arXiv preprint arXiv:1604.01839*.

Mazumdar, A., and Saha, B. 2017a. Clustering with noisy queries. In *Advances in Neural Information Processing Systems*, 5788–5799.

Mazumdar, A., and Saha, B. 2017b. Query complexity of clustering with side information. In *Advances in Neural Information Processing Systems*, 4682–4693.

Mazumdar, A., and Saha, B. 2017c. A theoretical analysis of first heuristics of crowdsourced entity resolution. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Misra, J., and Gries, D. 1982. Finding repeated elements. *Science of computer programming* 2(2):143–152.

Parzen, E. 1962. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics* 33(3):1065–1076.

Popescu, P. G.; Dragomir, S.; Sluşanschi, E. I.; and Stănăşilă, O. N. 2016. Bounds for kullback-leibler divergence. *Electronic Journal of Differential Equations* 2016.

Shah, D.; Choudhury, T.; Karamchandani, N.; and Gopalan, A. 2019. Sequential mode estimation with oracle queries.

Soare, M.; Lazaric, A.; and Munos, R. 2014. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, 828–836.

Wald, A. 1944. On cumulative sums of random variables. *The Annals of Mathematical Statistics* 15(3):283–296.