In Proc. AISTATS 2022, pp. 5815--5852, PMLR, 2022.

PAC Mode Estimation using PPR Martingale Confidence Sequences

Shubham Anand Jain^{*1} **Rohan Shah**$\dagger^{1}$ **Sanit Gupta**$\dagger^{2}$ **Denil Mehta**$\dagger^{2}$
Inderjeet Nair$\dagger^{2}$ **Jian Vora**^{*2} **Sushil Khyalia**$\dagger$ **Sourav Das**‡
Vinay J. Ribeiro$\dagger$ **Shivaram Kalyanakrishnan**†
†Indian Institute of Technology Bombay, *Stanford University, ‡ University of Illinois Urbana-Champaign
(Authors marked ^{1} contributed equally; authors marked ^{2} contributed equally.)

Abstract

We consider the problem of correctly identifying the *mode* of a discrete distribution $\mathcal{P}$ with sufficiently high probability by observing a sequence of i.i.d. samples drawn from $\mathcal{P}$. This problem reduces to the estimation of a single parameter when $\mathcal{P}$ has a support set of size $K = 2$. After noting that this special case is tackled very well by prior-posterior-ratio (PPR) martingale confidence sequences (Waudby-Smith and Ramdas, 2020), we propose a generalisation to mode estimation, in which $\mathcal{P}$ may take $K \geq 2$ values. To begin, we show that the "one-versus-one" principle to generalise from $K = 2$ to $K \geq 2$ classes is more efficient than the "one-versus-rest" alternative. We then prove that our resulting stopping rule, denoted PPR-1v1, is asymptotically optimal (as the mistake probability is taken to 0). PPR-1v1 is parameter-free and computationally light, and incurs significantly fewer samples than competitors even in the non-asymptotic regime. We demonstrate its gains in two practical applications of sampling: election forecasting and verification of smart contracts in blockchains.

1 INTRODUCTION

We investigate the problem of estimating the mode of a given, arbitrary, discrete probability distribution $\mathcal{P} = (p, v, K)$ by observing a sequence of i.i.d. samples drawn according to $\mathcal{P}$. Here $\mathcal{P}$ takes values from the support set $v = \{v_1, v_2, \dots, v_K\}$ according to the probability vector $p = \{p_1, p_2, \dots, p_K\}$ for some $K \geq 2$. For $1 \leq i \leq K$, the probability of obtaining v_i from $\mathcal{P}$ is p_i. We assume that $\mathcal{P}$ has a unique mode, and without loss of generality, $p_1 > p_2 \geq p_3 \geq p_4 \geq \dots \geq p_K$ (which makes v_1 the mode).

Our aim is to provide a procedure $\mathcal{L}$ to identify the mode of $\mathcal{P}$. At each step $t \geq 1$, $\mathcal{L}$ can either ask for a sample $x^t \sim \mathcal{P}$ or it can terminate and declare its answer. For "mistake probability" $\delta \in (0, 1)$, $\mathcal{L}$ is said to be δ-correct if for every qualifying discrete distribution $\mathcal{P}$, $\mathcal{L}$ terminates with probability 1 and correctly identifies the mode of $\mathcal{P}$ with probability at least $1 - \delta$. If $\mathcal{L}$ terminates after observing the sequence of samples $x^1, x^2, \dots, x^T$ for some $T \geq 1$, we may assume that its answer is the most frequent value of $\mathcal{P}$ in this sequence, since it can be argued that no other choice can decrease the mistake probability across all problem instances. Hence, it is convenient to view $\mathcal{L}$ simply as a *stopping rule*, which only needs to decide when to terminate. We aim to devise a δ-correct stopping rule $\mathcal{L}$ with low *sample complexity*—informally the number of samples T observed before stopping.

In order to make our problem "properly" PAC, we could introduce a tolerance parameter ϵ, with the implication that any returned value with associated probability at least $p_1 - \epsilon$ will be treated as correct. We omit this generalisation, noting that it can be handled quite easily by the methods proposed in the paper. In fact, our version with $\epsilon = 0$ exactly matches the problem defined by Shah et al. (2020), whose state-of-the-art results are our primary baseline. Shah et al. (2020) show the following lower bound.

Theorem 1 (Lower bound (Shah et al., 2020))**.** *Fix $\delta \in (0, 1)$, $K \geq 2$, and a δ-correct stopping rule $\mathcal{L}$. For each categorical distribution $\mathcal{P} = (p, v, K)$, the expected number of samples observed by $\mathcal{L}$ is at least*

$$LB(\mathcal{P}, \delta) \overset{def}{=} \sup_{\mathcal{P}' : mode(\mathcal{P}') \neq mode(\mathcal{P})} \frac{1}{KL(\mathcal{P}||\mathcal{P}')} \ln\left(\frac{1}{2.4\delta}\right),$$

where $KL(P||P')$ denotes the KL divergence between categorical distributions P, P' with same support set.

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

Shah et al. (2020) also give a stopping rule, denoted $\mathcal{A}_1$, whose sample complexity is upper-bounded to within a logarithmic factor of this lower bound.

In this paper, we approach the PAC mode estimation problem from a different perspective. In recent work, Waudby-Smith and Ramdas (2020) propose prior-posterior-ratio (PPR) martingale confidence sequences as a novel framework to obtain "anytime" confidence bounds on unknown parameters of a probability distribution. The resulting stopping rule is simple, with no need for tuning, and yet works surprisingly well in practice. Encouraged by this empirical finding, we investigate the application of the PPR martingale test to PAC mode estimation. Below we summarise the contents and contributions of our paper.

- We begin by reviewing the PPR martingale test (Waudby-Smith and Ramdas, 2020) in Section 2, and apply it directly to our "base case" of $K = 2$. Empirical comparisons establish clear evidence of the relative efficiency of this test.
- In Section 3, we propose three natural methods to generalise the PPR martingale test to mode estimation ($K \geq 2$). Two of these are the "one-versus-one" (1v1) and "one-versus-rest" (1vr) approaches used commonly in multi-class machine learning tasks; the third applies the multi-dimensional (MD) variant of the PPR martingale test (Waudby-Smith and Ramdas, 2020). The "one-versus-one" method, denoted PPR-1v1, is parameter-free, easy to implement, and computationally lighter than competitors. Experiments indicate that PPR-1v1 is also the most sample-efficient among the algorithms.
- In Section 4, we provide two theoretical arguments to explain the efficiency of PPR-1v1.
    1. We prove that for many commonly used Chernoff bounds, the 1vr adaptation to mode estimation cannot terminate before the 1v1 variant; additionally the MD variant of PPR cannot terminate before PPR-1v1. These results hold for *every single run*, and establish 1v1 as a clear choice for mode estimation. Even $\mathcal{A}_1$, originally implemented as a 1vr variant (Shah et al., 2020), is seen to perform much better by switching to 1v1 (although it remains inferior to PPR-1v1).
    2. We prove that PPR-1v1 is *asymptotically optimal*, in the sense that for every categorical distribution $\mathcal{P}$, the ratio of the expected sample complexity of PPR-1v1 and LB$(\mathcal{P}, \delta)$ goes to 1 as the mistake probability δ is taken to 0. To the best of our knowledge, this guarantee is the first of its kind for mode estimation, although similar results have been provided in the multi-armed bandits literature (Garivier and Kaufmann, 2016). Interestingly, 1vr variants (such as $\mathcal{A}_1$) appear not to be asymptotically optimal.
- Over the years, the mode estimation problem has received attention in many different contexts (Parzen, 1962; Manku and Motwani, 2002). In Section 5, we illustrate the relevance of PPR-1v1 in two contrasting real-life applications. First, we show that when used as a subroutine, PPR-1v1 can reduce the sample complexity of winner-forecasting in indirect elections (Karandikar, 2018). Thereafter, we present its application to probabilistic verification in permissionless blockchains (Das et al., 2019).

In short, our paper proposes PPR-1v1 as a novel stopping rule for PAC mode estimation, and provides both theoretical and empirical reasons to justify the choice.

2 THE PPR MARTINGALE TEST

In this section, we consider the "base case" of mode estimation, in which $\mathcal{P}$ takes exactly $K = 2$ values. Notice that $\mathcal{P}(p, v, 2)$ is a Bernoulli distribution that generates v_1 with probability p_1 and v_2 with probability $p_2 = 1 - p_1$. Treating $p_1 \in [0, 1]$ as the sole parameter of the distribution, our task is to devise a δ-correct stopping rule to test if $p_1 > \frac{1}{2}$. Since p_1 may be arbitrarily close to $\frac{1}{2}$, it is not possible to decide beforehand how many samples suffice for the test to succeed. An unfortunate consequence of having a *random* stopping time is that it cannot be used directly within concentration inequalities such as Chernoff bounds. Rather, stopping rules invariably go through a union bound over all possible stopping times, dividing the mistake probability δ among them (Kalyanakrishnan et al., 2012; Kaufmann and Kalyanakrishnan, 2013). Although there has been progress towards optimising this apportioning of δ (Jamieson et al., 2014; Garivier, 2013), resulting methods still have tunable parameters in their "decay rates", which govern the stopping time.

The recent development of "time-uniform" or "anytime" Chernoff bounds (Howard et al., 2020) relieve the experimenter of tedious parameter-tuning. Arising from this line of research is the framework of prior-posterior-ratio (PPR) martingale confidence sequences (Waudby-Smith and Ramdas, 2020), which yields a simple, intuitive stopping rule. Although the rule may be applied more widely, we restrict our upcoming discussion to the Bernoulli case at hand: that is, to test whether $p_1 > \frac{1}{2}$.

To apply the PPR martingale framework, we maintain a belief distribution π for p_1 over its range $[0, 1]$, and update π according to Bayes' rule as samples are observed. Our aim is still to provide a frequentist guarantee that holds for all possible values of p_1 (δ-correctness). To this end we must ensure that the prior distribution π^0 gives non-zero density to all possible values of p_1. We do so by adopting the uniform prior $\pi^0(q) = 1$ for $q \in [0, 1]$. For $t \geq 1$, we update our belief distribution after observing sample x^t:

$$\pi^t(q) = \frac{\pi^{t-1}(q) \cdot (q)^{\mathbf{1}[x^t=v_1]} \cdot (1-q)^{\mathbf{1}[x^t=v_2]}}{\int_{\rho=0}^{1} \pi^{t-1}(\rho) \cdot (\rho)^{\mathbf{1}[x^t=v_1]} \cdot (1-\rho)^{\mathbf{1}[x^t=v_2]} d\rho}.$$

The prior-posterior-ratio (PPR) at $q \in [0, 1]$ is given by $R^t(q) = \frac{\pi^0(q)}{\pi^t(q)}$. Waudby-Smith and Ramdas (2020) show that the sequence of sets $(C^t)_{t=0}^{\infty}$, where $C^t \overset{\text{def}}{=} \{q : R^t(q) < \frac{1}{\delta}\}$, is a $(1-\delta)$-confidence sequence for p_1 (Waudby-Smith and Ramdas, 2020). In other words, we have the "anytime" guarantee that

$$\mathbb{P}\{\, \exists\, t \geq 0 : p_1 \notin C^t\} \leq \delta. \tag{1}$$

The correctness of (1) is shown by establishing that the PPR evaluated at the *true* parameter value, p_1, is a martingale, and then applying Ville's inequality for nonnegative supermartingales (Waudby-Smith and Ramdas, 2020, see Appendix B.1). For our special case of estimating the parameter of a Bernoulli distribution, the belief distribution π^t and hence the PPR R^t assume a convenient form if initialised with the uniform prior. Suppose the sequence of samples up to time t is $x^1, x^2, \ldots, x^t$, which contains s_1^t occurrences of v_1 and $s_2^t = t - s_1$ occurrences of v_2. Then for $t \geq 0$ and $q \in [0, 1]$, we obtain $\pi^t(q) = \text{Beta}(q; s_1^t + 1, s_2^t + 1)$ (the pdf of a Beta distribution with parameters $s_1^t + 1$ and $s_2^t + 1$, evaluated at q). We can terminate as soon as the $(1-\delta)$-confidence sequence on p_1 does not contain $\frac{1}{2}$. For easy readability, let us define indices first(t) and second(t), where $(\text{first}(t), \text{second}(t)) \in \{(1, 2), (2, 1)\}$ satisfies $s_{\text{first}(t)}^t \geq s_{\text{second}(t)}^t$. We obtain the following simple stopping rule, applied at each time step $t \geq 1$.

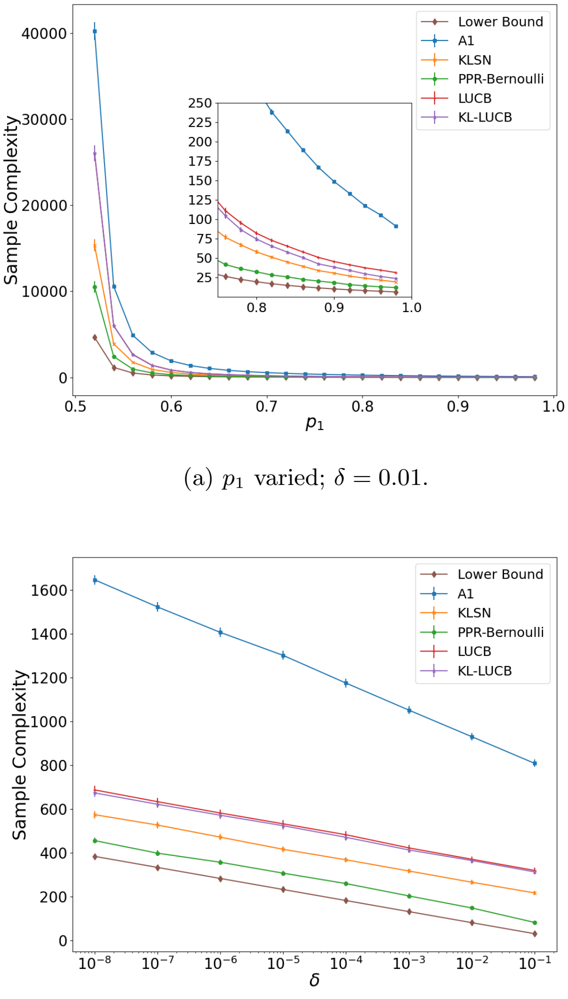
PPR-Bernoulli: Stop, declare $v_{\text{first}(t)}$ as mode iff $\text{Beta}\left(\frac{1}{2}; s_{\text{first}(t)}^t + 1, s_{\text{second}(t)}^t + 1\right) \leq \delta$.

Note that the LHS of the PPR-Bernoulli stopping rule can be evaluated *exactly* as a rational, using integer arithmetic, requiring only a lightweight incremental update after each sample. As we see shortly, many other stopping rules require much heavier computation, such as to perform numerical optimisation.

2.1 Empirical Comparisons, $K = 2$

For the problem of determining the sign of $p_1 - \frac{1}{2}$ from samples, the predominant approach in the literature is to construct lower and upper confidence bounds on p_1 that hold with probability $1 - \delta_t$ for each $t \geq 1$, satisfying $\sum_{t=1}^{\infty} \delta_t \leq \delta$. The δ-correctness of the procedure is ensured by terminating only when the lower confidence bound exceeds $\frac{1}{2}$, or the upper confidence bound falls below $\frac{1}{2}$. We compare PPR-Bernoulli with several variants from the literature. In Figure 1, we plot the sample complexity of different algorithms as p_1 and δ are varied.

A common choice is to set $\delta_t = k\frac{\delta}{t^\alpha}$, with constants k and α tuned for efficiency, while ensuring δ-correctness. As representatives of this approach, we pick the LUCB and KL-LUCB algorithms (Kauf-



(a) p_1 varied; $\delta = 0.01$.

(b) δ varied; $p_1 = 0.65$.

Figure 1: Comparison of stopping rules for the Bernoulli case ($K = 2$). Both plots show sample complexity: in (a) as p_1 is varied, and in (b) as δ is varied. The results are averages from 100 runs. Error bars show one standard error (in both plots very small).

mann and Kalyanakrishnan, 2013). The former inverts Hoeffding's inequality to obtain lower and upper confidence bounds, while the latter uses a tighter Chernoff bound. Although these algorithms themselves are meant for bandit applications, their efficiency crucially depends on the tightness of the confidence bounds applied to each arm. The tuned confidence bounds (Kaufmann and Kalyanakrishnan, 2013) hence become suitable baselines for our comparison.[1]

With the intent of avoiding a naïve union bound over time, Garivier (2013) applies a so-called *peeling* argument to divide time into increasingly-sized slices. He obtains confidence regions by associating the random stopping time with a self-normalised process. The resulting stopping rule, which we denote KL-SN, still has a tunable parameter "c", which we set as recommended by Garivier (2013). Although the $\mathcal{A}_1$ algorithm (Shah et al., 2020) is designed specifically for mode estimation, we include it in this comparison to observe its performance when $K = 2$. In this special case, the algorithm reduces to an application of an empirical Bernstein bound (Maurer and Pontil, 2009).

The two plots in Figure 1 are remarkably consistent as p_1 and δ are varied. KL-LUCB shows a marginal improvement over (Hoeffding) LUCB, while KL-SN clearly outperforms both. However, PPR-Bernoulli is significantly more efficient than even KL-SN. Surprisingly, in spite of using variance information, $\mathcal{A}_1$ registers the worst performance among all the methods compared. We attribute this result to slack in the constants used in its stopping rule.

The empirical evidence of its sample efficiency, along with its simplicity and non-reliance on parameter-tuning, make PPR-Bernoulli an attractive proposition for stopping problems. In Section 3, we consider three separate ways to generalise it to mode estimation. In Section 4 we follow with theoretical analysis to explain the empirical findings in sections 2 and 3.

3 GENERALISATION TO $K \geq 2$

In the broader machine learning literature, the most common approaches for generalising 2-class problems to more classes are "one-versus-one" (denoted 1v1) and "one-versus-rest" (denoted 1vr). We investigate both approaches. We also consider the direct application of the multi-dimensional (MD) variant of the PPR martingale test (Waudby-Smith and Ramdas, 2020).

3.1 One-versus-one (1v1) Approach

In the first $t \geq 1$ samples, let the number of occurrences of value v_i be s_i^t, $1 \leq i \leq K$. The 1v1 generalisation is based on the idea that if v_i is to be declared the mode, we need to be sufficiently sure that v_i is more probable than v_j for $j \in \{1, 2, \ldots, K\}, i \neq j$. Correspondingly, we simultaneously run PPR-Bernoulli tests on each (i, j) pair with mistake probability $\frac{\delta}{K-1}$. Each (i, j) test relies solely on the number of occurrences of v_i and v_j, disregarding other values. Hence it amounts to observing samples from a Bernoulli variable with parameter $\frac{p_i}{p_i+p_j}$, and verifying which side of $\frac{1}{2}$ its mean lies. The overall procedure stops when some $i \in \{1, 2, \ldots, K\}$ has won each of its tests. By a union bound, with probability at least $1-\delta$, the (true) mode v_1 will not ever lose a test. Thus, upon termination, v_1 is returned with probability at least $1 - \delta$.

Whereas the description above suggests we need to monitor $\binom{K}{2}$ tests at each step, closer inspection reveals that a much lighter implementation is possible. As before, let first(t) denote the index of the most-frequently occurring value (with arbitrary tie-breaking) after t samples: that is, $s_{\text{first}(t)}^t \geq s_i^t$ for $i \in \{1, 2, \ldots, K\}$. Now, if at all a winner is identified after t samples, clearly it must be $v_{\text{first}(t)}$, which has as many occurrences as any other value. Hence, we only need to track tests involving $v_{\text{first}(t)}$. Now, it is also immediate that $v_{\text{first}(t)}$ wins all its tests if and only if it defeats the second most frequently occurring value, which we denote second(t): that is, second$(t) \in \{1, 2, \ldots, K\}$, second$(t) \neq$ first(t) satisfies $s_{\text{second}(t)}^t \geq s_i^t$ for $i \in \{1, 2, \ldots, K\} \setminus \{\text{first}(t)\}$. Hence, we may implement our stopping rule, denoted PPR-1v1, using a *single* PPR-Bernoulli test at each $t \geq 1$.

PPR-1v1: Stop and declare $v_{\text{first}(t)}$ as mode iff
$$\text{Beta}\left(\tfrac{1}{2}; s_{\text{first}(t)}^t + 1, s_{\text{second}(t)}^t + 1\right) \leq \tfrac{\delta}{K-1}.$$

Tracking first(t) and second(t) is a simple computation; as observed earlier, it is also efficient to compute the Beta density at $\frac{1}{2}$. Indeed our experiments show that PPR-1v1 is much faster computationally than other mode estimation algorithms (see Appendix B).

3.2 One-versus-rest (1vr) Approach

Notice that under PPR-1v1, sample x^t at each step $t \geq 1$ contributes only to the $K - 1$ PPR-Bernoulli tests of the particular i from $\{1, 2, \ldots, K\}$ that satisfies $v_i = x^t$. The $\binom{K-1}{2}$ tests corresponding to values other than x^t receive no information. The 1vr approach becomes an alternative to address this *apparent* wastage of information. Under the 1vr scheme, we associate a Bernoulli variable B_i with each value

[1] Details of all our implementations are given in Appendix A; links to code are provided in Appendix B.

v_i, $1 \leq i \leq K$, which has probability p_i of generating v_i, and probability $1 - p_i$ of generating its *negation* "$\neg v_i$". Consequently, each sample of $\mathcal{P}$ adds to one of the outcomes of B_i for *each* $i \in \{1, 2, \ldots, K\}$. We draw an anytime confidence sequence for B_i with mistake probability $\frac{\delta}{K}$, and terminate after $t \geq 1$ samples if the confidence set of $B_{\text{first}(t)}$ does not overlap with any of the others. Invoking the PPR martingale confidence sequence, we note that with probability at least $1 - \frac{\delta}{K}$, p_i will lie in all the intervals $(\text{LCB}_i^t, \text{UCB}_i^t)$, $t \geq 1$, where $\text{LCB}_i^t = \min\{q \in [0,1] : \pi^t(q) = \frac{\delta}{K}\}$ and $\text{UCB}_i^t = \max\{q \in [0,1] : \pi^t(q) = \frac{\delta}{K}\}$ can be computed numerically. The δ-correctness of the 1vr rule, given below, follows from a union bound on the mistake probabilities of each B_i, $1 \leq i \leq K$.

PPR-1vr: Stop and declare $v_{\text{first}(t)}$ as mode iff for $1 \leq i \leq K$, $i \neq \text{first}(t)$, $\text{LCB}_{\text{first}(t)}^t \geq \text{UCB}_i^t$.

3.3 Multi-dimensional (MD) PPR Test

PPR Martingale confidence sequences can be directly constructed for the multi-dimensional parameter vector of $\mathcal{P}$ (Waudby-Smith and Ramdas, 2020, see Appendix C). In this approach, denoted PPR-MD, at each $t \geq 1$ we maintain a confidence set C^t with $\bar{p} \in [0,1]^k$, such that $C^t \overset{\text{def}}{=} \{\bar{p} : R^t(\bar{p}) < \frac{1}{\delta}\}$. We stop at time t when all $\bar{p} \in C^t$ have the same unique mode. With the Dirichlet distribution being the conjugate prior of the categorical distribution, $R^t(\bar{p})$ has a convenient form if initialised with a uniform prior:

$$R^t(\bar{p}) = \frac{1}{(K-1)!} \times \frac{1}{\prod_{i=1}^{K} \bar{p}_i^{\alpha_i^t - 1}} \times \frac{\prod_{i=1}^{K} \Gamma(\alpha_i^t)}{\Gamma(\sum_{i=1}^{K} \alpha_i^t)},$$

where for $1 \leq i \leq K$, $\alpha_i^t = s_i^t + 1$. Observe that this formulation reduces to PPR-Bernoulli for $K = 2$. However, for $K > 2$, checking for a unique mode in C^t does not simplify to a convenient formula; it requires a numerical computation that increases steeply with K. We do not perform extensive experiments with PPR-MD—a choice justified by Lemma 2 (Section 4).

3.4 Empirical Comparisons

We compare PPR-1v1 and PPR-1vr with other mode estimation algorithms on a variety of discrete distributions. Table 1 summarises the results.

The $\mathcal{A}_1$ algorithm (Shah et al., 2020) is essentially a 1vr approach that uses Empirical Bernstein confidence bounds (Maurer and Pontil, 2009). Noting that it can just as well be implemented in a 1v1 form, we include such a variant, denoted $\mathcal{A}_1$-1v1, in our experimental comparisons with PPR. For good measure, we also include 1v1 and 1vr variants based on the KL-SN confidence bound (Garivier, 2013), which finished second to PPR-Bernoulli for $K = 2$ (see Section 2).

In Table 1, we observe the same trend on *each* problem instance: (1) The 1v1 variant of each stopping rule outperforms the corresponding 1vr variant, and (2) PPR is most sample-efficient, followed by KL-SN and $\mathcal{A}_1$. Although the prohibitive running time of PPR-MD prevents a thorough assessment, a few informal runs indicate that its sample complexity is well in excess of even PPR-1vr.

Whereas the results in Table 1 are for a fixed value of $\delta = 0.01$, we conduct a second set of experiments to compare the performance of the different algorithms as δ is varied. In particular, we investigate the "asymptotic" regime, in which δ is taken to 0. In Figure 2, we plot ratio of the empirical sample complexity and the lower bound from Theorem 1, varying δ while keeping the distribution fixed to $\mathcal{P}_3$ from Table 1. Observe that once again, the relative order among the algorithms remains the same. The 1v1 variant of each algorithm performs better than its 1vr counterpart. Moreover, the curves for PPR-1v1 and KLSN-1v1 suggest that these rules might be asymptotically optimal.

The empirical evidence supporting the PPR martingale test and the 1v1 approach to mode estimation is compelling. In the forthcoming section, we provide theoretical reasons to explain our observations.[2]

4 THEORETICAL JUSTIFICATION

In our experiments, we observe that not only do the 1v1 variants of each method perform better than 1vr in aggregate, they terminate *before* the 1vr variants on *every single run.* We prove this result true for some

[2] After this paper was submitted for review, the authors were pointed to recent related work by Haddenhorst et al. (2021), who focus on the identification of a generalised Condorcet winner in multi-dueling bandits. Mode estimation is a special case of the problem they consider. Their algorithm, based on the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality, has an upper bound which improves upon that given by Shah et al. (2020). In particular, the bound is independent of K, and depends on $\frac{1}{(p_1-p_2)^2} \ln \ln \frac{1}{(p_1-p_2)}$, rather than $\frac{1}{(p_1-p_2)^2} \ln \frac{1}{(p_1-p_2)}$. However, our experiments show that the two variants of their algorithm—DKW-1, which is provided the knowledge of $p_1 - p_2$, and DKW-2, which has no such prior knowledge—both perform worse than PPR-1v1. For example, when run on problem instance with $K = 2, p_1 = 0.66, \delta = 0.01$, PPR-Bernoulli requires roughly 124 samples, whereas DKW-1 and DKW-2 take roughly 469 and 871 samples, respectively. The inferior empirical performance of the latter algorithms in spite of their superior upper bound can be explained by the accompanying constant factors. The notion of asymptotic optimality that we present in sections 3 and 4 requires even the constant factor to be tight, albeit as $\delta \to 0$.

Table 1: Sample complexity comparison for mode estimation, run with mistake probability $\delta = 0.01$. The number after the × symbol indicates the multiplicity of that particular probability value in the distribution; thus $\mathcal{P}_1$ has $p = (0.5, 0.25, 0.25)$. The values reported are averages from 100 or more runs, and show one standard error.

Distribution	K	Type	$\mathcal{A}_1$ (Shah et al., 2020)	KL-SN (Garivier, 2013)	PPR
$\mathcal{P}_1$: .5, .25 × 2	3	1vr 1v1	1344±20 1158±19	418±14 346±13	262±12 **218±11**
$\mathcal{P}_2$: .4, .2 × 3	4	1vr 1v1	1919±29 1516±24	632±18 468±15	397±15 **298±13**
$\mathcal{P}_3$: .2, .1 × 8	9	1vr 1v1	5082±51 3340±43	1900±42 1138±31	1201±29 **789±28**
$\mathcal{P}_4$: .1, .05 × 18	19	1vr 1v1	12015±129 7352±88	4686±81 2554±57	2850±55 **1840±53**
$\mathcal{P}_5$: .35, .33, .12, .1 × 2	5	1vr 1v1	155277±2356 117988±2078	63739±2238 47205±1291	38001±1311 **33660±1125**
$\mathcal{P}_6$: .35, .33, .04 × 8	10	1vr 1v1	158254±2442 121150±2183	66939±2241 49576±1341	41963±1330 **36693±1185**

of the methods. We also formally establish that PPR-MD and $\mathcal{A}_1$ cannot terminate before PPR-1v1.

Lemma 2. *Let $X = x^1, x^2, \ldots$ be an infinite sequence of samples from $\mathcal{P}$. For algorithm $\mathcal{L}$ and $\delta \in (0, 1)$, let $T(\mathcal{L}, X, \delta)$ be the stopping time of $\mathcal{L}$ on X, when run with mistake probability δ. For algorithms $\mathcal{L}_1, \mathcal{L}_2$, let the proposition $\mathcal{G}(\mathcal{L}_1, \mathcal{L}_2, X, \delta)$ denote "If $T(\mathcal{L}_2, X, \delta)$ is finite, then $T(\mathcal{L}_1, X, \delta) \leq T(\mathcal{L}_2, X, \delta)$". For all $\mathcal{P}$, for all X generated from $\mathcal{P}$, for all $\delta \in (0, 1)$, we have*

(i) $\mathcal{G}(LUCB\text{-}1v1, LUCB\text{-}1vr, X, \delta)$,

(ii) $\mathcal{G}(\mathcal{A}_1\text{-}1v1, \mathcal{A}_1\text{-}1vr, X, \delta)$,

(iii) $\mathcal{G}(PPR\text{-}1v1, PPR\text{-}MD, X, \delta)$,

(iv) $\mathcal{G}(PPR\text{-}1v1, \mathcal{A}_1\text{-}1v1, X, \delta)$.

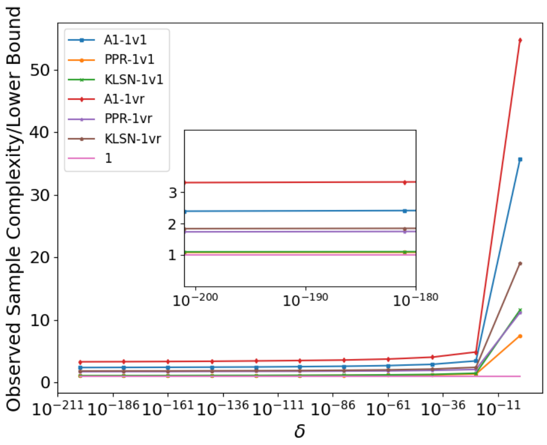


Figure 2: Comparison of different stopping rules on $\mathcal{P}_3$, for small values of δ. The y axis plots the ratio of the empirical stopping time (averaged over 100 or more runs) and LB$(\mathcal{P}_3, \delta)$, defined in Theorem 1.

We give a proof of the lemma in Appendix C. The proofs of (i) and (ii) formalise the intuition that although the 1vr variants update all their K tests with each sample, the test to separate any two variables is less effective than that under the corresponding 1v1 variant. We conjecture that $\mathcal{G}(\text{PPR-1v1}, \text{PPR-1vr}, X, \delta)$ is also true, in fact verifying it to be the case for $|X| \leq 200$. We also believe that a similar result applies to the 1v1 and 1vr variants of KL-LUCB and KL-SN (but the proofs become cumbersome). Result (iii) is straightforward to establish from first principles. To show (iv), consider that after t samples, $\mathcal{A}_1$-1v1 draws upper and lower confidence bounds on the mean as $\hat{p}^t \pm \beta^t$, where $\hat{p}^t$ is the empirical mean and β^t the confidence width. In particular, $\beta^t = \beta_1^t + \beta_2^t$, where $\beta_1^t = \sqrt{\frac{2V^t \ln(4t^2/\alpha)}{t}}$ and $\beta_2^t = \frac{7 \ln(4t^2/\alpha)}{3(t-1)}$, with V^t being the empirical variance and α the input mistake probability. When $\hat{p}^t$ is within a constant distance from 0.5 (hence V^t is close to its maximum of 0.25), we show that the PPR confidence set is contained in $[\hat{p}^t - \beta_1^t, \hat{p}^t + \beta_1^t]$. When $\hat{p}^t$ is either sufficiently small or sufficiently large, t (at termination) can itself be upper-bounded in terms of $\hat{p}^t$, in turn lower-bounding β_2^t and guaranteeing that the PPR confidence set is contained in $[\hat{p}^t - \beta_2^t, \hat{p}^t + \beta_2^t]$.

Lemma 2 offers theoretical justification for many of the trends observed in Table 1. The theoretical question that remains open is an explanation of Figure 2: is PPR-1v1 indeed asymptotically optimal? We obtain an affirmative answer.

Theorem 3 (Optimality of PPR-1v1)**.** *Fix $\delta \in (0, 1)$, $K \geq 2$, and distribution $\mathcal{P} = (p, v, K)$. Let $\tau(\mathcal{P}, \delta)$ be the expected stopping time of PPR-1v1 on $\mathcal{P}$ when* *run with mistake probability δ, and let $LB(\mathcal{P}, \delta)$ be the lower bound defined in Theorem 1. Then*

$$\lim_{\delta \to 0} \frac{\tau(\mathcal{P}, \delta)}{LB(\mathcal{P}, \delta)} = 1.$$

The proof of the theorem is given in Appendix D. In the proof, we establish that with a little over $\mathrm{LB}(\mathcal{P}, \delta)$ samples from $\mathcal{P}$, a sufficient number of samples are obtained for separating value v_1 from each of the others. The proof of separation for each pair uses a similar sequence of steps as of Garivier and Kaufmann (2016).

In short, our theoretical analysis reinforces PPR-1v1 as the method of choice for PAC mode estimation. For good measure, Appendix E presents an explicit upper bound on the sample complexity of PPR-1v1, which holds for all $\delta \in (0, 1)$. This bound improves upon that of $\mathcal{A}_1$ (Shah et al., 2020) by a constant factor.

5 PRACTICAL APPLICATIONS

Our main motivation for devising better mode estimation algorithms is their practical significance, which we illustrate through two contrasting applications.

5.1 Forecasting in Indirect Elections

Opinion polls to forecast the winner of an upcoming election are a natural application of mode estimation. In fact, the algorithms discussed in Section 3 can all be applied with only minor alterations to *plurality* systems, wherein the task is precisely that of determining the choice preferred by the largest fraction of the target population. Waudby-Smith and Ramdas (2020) illustrate the use of the PPR martingale test on this application, while focusing on *without-replacement* sampling. In parliamentary democracies such as India (Karandikar et al., 2002) and the U.K. (Payne, 2003), a two-level voting system is used to elect governments. In this system, individuals in each *constituency* (or *seat*)—typically a geographically contiguous region—elect a party based on plurality; the party winning the most seats forms the government. Forecasting the winning party in such an *indirect* voting system calls for a more sophisticated sampling procedure. Whereas it would *suffice* to separately identify the winner from each seat by sampling, it might be wasteful to do so when the overall winning party has a clear majority in its number of seats.

Formally, consider a setting where we have K parties and N constituencies. Each constituency $c \in \{1, 2, \ldots, N\}$ represents a discrete probability distribution $\mathcal{P}^c = (p^c, v, K)$, where for $i \in \{1, 2, \ldots, K\}$, v_i represents the political party i and p_i^c denotes the fraction of votes won by party i in constituency c (for simplicity we have assumed all parties compete in all constituencies). If $i_c^\star$ is the index of the mode of $\mathcal{P}^c$ (assumed unique), our objective is to determine

$$\operatorname*{argmax}_{i \in \{1,2,\ldots,K\}} \sum_{c=1}^{N} \mathbf{1}[i = i_c^\star]$$

correctly with probability at least $1 - \delta$. The objective of the sampling rule is to minimise the total votes sampled, $\sum_{c=1}^{N} T^c$, where T^c represents the number of votes queried in constituency c.

We consider a procedure that (1) keeps track of the current winners and leaders at the aggregate level, and (2) at each step samples the constituencies that appear most promising to confirm the aggregate trend. In principle, this algorithm, denoted DCB (for "Difference in Confidence Bounds") can be coupled with any algorithm that uses confidence bounds for mode estimation. Yet, we obtain the best results when DCB uses PPR-1v1 as a subroutine, thereby highlighting the relevance of PPR-1v1 not only as a stopping rule, but also as an input to on-line decision making.

DCB takes cue from the LUCB algorithm for best-arm identification in bandits (Kalyanakrishnan et al., 2012). At each step t, it identifies two parties, a^t and b^t, that appear the most promising to win the overall election: these parties are picked based on their current number of wins and "leads" in individual constituencies. Subsequently the algorithm chooses a constituency each for a^t and b^t, samples from which could "most" help distinguish the tally of the two. We provide a detailed specification of DCB in Appendix F.

We compare DCB with a round-robin strategy for picking the next constituency to sample. Both approaches can be implemented with different stopping rules, which are also varied. Table 2 shows our results on the 2014 parliamentary elections in India.[3] In this election, the winning party secured 282 seats from among 543, giving it a very large victory over the second-place party, which won 44 seats. Results from a much closer contest, in the state of Bihar, are given in Appendix G. We see a similar trend in both cases.

While it is not the central feature of this paper, it is worth noting that the DCB strategy indeed improves over round-robin polling by roughly a factor of two, regardless of the stopping rule. As intended, it does not waste samples on constituencies that are inconsequential to the overall result (observed in the "seats resolved" columns). Of more direct relevance to the theme of the paper is that even when embedded within

[3] Election results are in the public domain; the authors accessed them at `https://www.indiavotes.com/`.

Table 2: Sample complexity of various stopping rules when coupled with (1) round-robin (RR) polling of constituencies and (2) DCB. All experiments are run with mistake probability $\delta = 0.01$. Values shown are averages from 10 runs, and show one standard error. "Seats resolved" indicates the number of constituencies in which a winner was identified before the overall procedure terminated.

Algorithm	India-2014 (543 seats)	
	Samples	Seats Resolved
RR-$\mathcal{A}_1$-1v1	1578946 ± 10255	239 ± 2
RR-$\mathcal{A}_1$-1vr	1767464 ± 15828	234 ± 1
RR-KLSN-1v1	610181 ± 7072	240 ± 2
RR-KLSN-1vr	726512 ± 6186	236 ± 2
RR-PPR-1v1	**471661 ± 7373**	241 ± 3
RR-PPR-1vr	560815 ± 7778	238 ± 2
DCB-$\mathcal{A}_1$-1v1	856678 ± 3935	182 ± 1
DCB-$\mathcal{A}_1$-1vr	951684 ± 5246	180 ± 1
DCB-KLSN-1v1	325265 ± 2096	186 ± 2
DCB-KLSN-1vr	376108 ± 4067	181 ± 1
DCB-PPR-1v1	**256911 ± 2096**	188 ± 1
DCB-PPR-1vr	296580 ± 2372	184 ± 1

a decision-making outer loop, PPR continues to outperform $\mathcal{A}_1$ and KL-SN, and the 1v1 approach still dominates 1vr. Surmising that the sheer efficiency of PPR-1v1 makes it a good choice for embedding in more complex systems, we proceed to our next application.

5.2 Verifying Blockchain Smart Contracts

Our second application of PAC mode estimation is in a domain of growing contemporary relevance. Permissionless blockchains such as Bitcoin (Nakamoto, 2009) and Ethereum (Buterin, 2014) allow uncertified agents to join a pool of service providers, also called *nodes*. A recent feature that has emerged in such blockchains is the execution of "smart contracts" (Das et al., 2019; Buterin, 2014), which could include, for example, running computationally-heavy jobs such as machine learning algorithms. In an ideal world, a client who requires some computation performed can simply enter a smart contract with some particular node, and pay a fee for the service. Unfortunately, there is no guarantee that nodes in a permissionless blockchain are honest. A "Byzantine" (*malicious*) node could potentially return a quick-to-compute, incorrect output, to the detriment of the client.

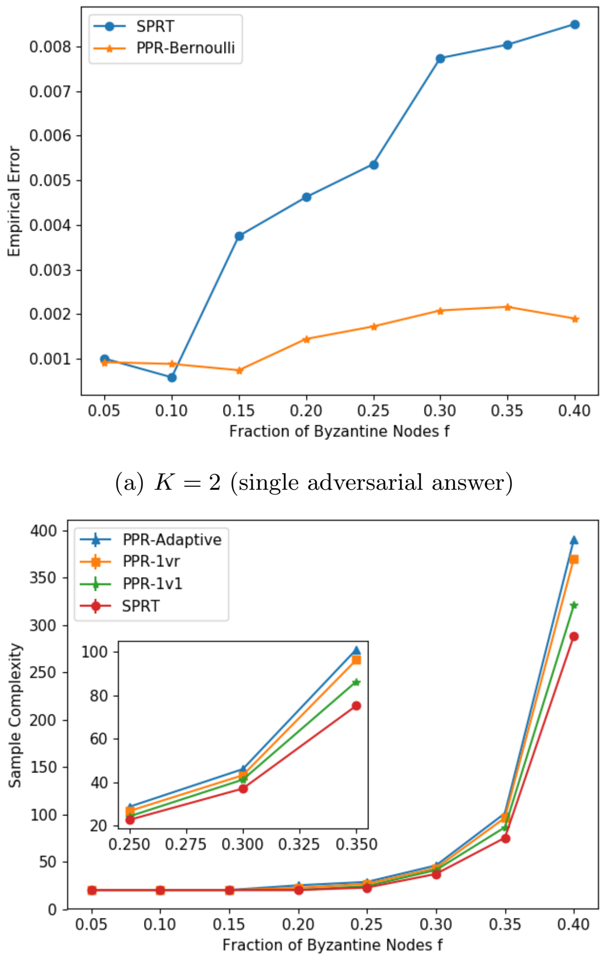
In recent work, Das et al. (2019) propose an approach for the *probabilistic verification* of smart contracts. Abstractly, assume that the computation to be performed for the client is deterministic, and it has a (yet unknown) output o_{correct}. The proposed model accommodates any blockchain in which the fraction of Byzantine nodes f is at most $f_{\max} \in [0, \frac{1}{2})$. With this assumption, it becomes feasible to give a probabilistic guarantee on obtaining the correct output. For any fixed mistake probability $\delta \in (0, 1)$, the client could ship out the computation to $\Theta\left(\frac{1}{(\frac{1}{2}-f_{\max})^2}\log(\frac{1}{\delta})\right)$ nodes, and take their majority response as the answer, thereby ensuring δ-correctness. Unfortunately, transaction costs can be substantial, especially those for computationally-intensive contracts. Hence, it is in the client's interest to minimise the number of nodes queried to achieve the same probabilistic guarantee. For example, a sequential procedure could potentially query fewer nodes if $f \ll f_{\max}$.

Our contribution in the context of this application is to propose PPR-1v1 as an alternative to the Sequential Probability Ratio Test (SPRT) (Wald, 1945), which is used by Das et al. (2019) for their verification procedure. This classical test finds use in many other engineering applications (Gross and Humenik, 1991; Chen et al., 2008), some of which could also benefit from the advantages of PPR-1v1 over SPRT.

To apply SPRT for verifying smart contracts, Das et al. (2019) assume that out of the total of N nodes in the blockchain, batches of size m, chosen uniformly at random, can be queried in sequence (the m queries per batch are performed *in parallel*, hence saving some time). For simplicity assume the answers returned are from the set $\{0, 1, 2, \dots\}$. Let $q = \frac{m}{N}$, and let $c_{i,t}$ be the number of times answer i is reported in the t^{th} step, $i \geq 0$, $t \geq 1$. Defining $l_{i,T} = \sum_{t=1}^{T}(2c_{i,t} - m)m$, a derivation (Das et al., 2019) establishes that δ-correct SPRT stops at time T, giving i as the answer if

$$l_{i,T} > \ln\left(\frac{1-\delta}{\delta}\right)\frac{2q(1-q)N(1-f_{\max})f_{\max}}{1-2f_{\max}}.$$

The primary disadvantage of SPRT herein is the need for the user to provide $f_{\max}$, which is used in the stopping rule. While a lower value of $f_{\max}$ will improve the efficiency of the rule, unfortunately δ-correctness no longer holds if f, the true fraction of Byzantine nodes, exceeds f_{max}. Figure 3a plots the empirical error made by SPRT (averaged over 50,000 runs) on a problem instance in which the Byzantine nodes all give the same (incorrect) answer. We fix $f_{\max} = 0.1$, and plot the error by varying f. The blockchain has $N = 1600$ nodes, of which SPRT samples $m = 20$ at a time. Although the test is run using mistake probability $\delta = 0.005$, observe that the empirical error exceeds δ when $f > f_{\max}$. Since the verification task at hand is precisely that of PAC mode estimation, PPR-1v1 becomes a viable alternative, especially since it does not need the knowledge of $f_{\max}$. In fact, PPR-1v1 can identify the mode even if its associated probability is less than $\frac{1}{2}$ (although in this case, it can no



(a) $K = 2$ (single adversarial answer)

(b) $K = 10$

Figure 3: Comparisons with SPRT for the probabilistic verification of smart contracts, obtained with parameter settings $N = 1600$, $m = 20$, $\delta = 0.005$, $f_{max} = 0.1$. Plot (a) shows the empirical error rates of the algorithms as the true Byzantine fraction f is varied, taking $K = 2$. Plot (b) shows the sample complexity of various algorithms against varying f, on an instance with $K = 10$ answers.

longer be guaranteed that the mode is o_{correct}, since the Byzantine nodes may collude). Observe from Figure 3a that unlike SPRT, the empirical error rate of PPR-1v1 (equivalent to PPR-Bernoulli since we have set $K = 2$) remains within δ even for $f > f_{\text{max}}$.

In Figure 3b, we compare the sample complexities of PPR-1v1 and SPRT. Whereas other problem parameters (including f_{max}) stay the same as before, we consider an instance in which $K = 10$. The single correct answer is given by a $(1 - f)$-fraction of the nodes, while 9 different incorrect answers are given by the Byzantine nodes, each equally common. The version of SPRT used is a 1vr adaptation of the basic procedure (Das et al., 2019) to $K = 10$. First, we observe that SPRT terminates before PPR-1v1 and PPR-1vr at all values of f. The PPR algorithms pay this price for having to assure δ-correctness at all values of $f < \frac{1}{2}$, unlike SPRT, which does so only for $f < f_{max}$. In the same plot, we show the performance of another PPR variant, denoted "PPR-Adaptive". In reality, we cannot be sure about the number of answers K that will be returned by the blockchain's nodes—and hence cannot use it in our stopping rule. Under PPR-Adaptive, which is a 1v1 strategy, the overall mistake probability δ is divided into the *infinite* sequence $k\frac{\delta}{1^2}, k\frac{\delta}{2^2}, k\frac{\delta}{3^2}, \dots$ (with $k = \frac{6}{\pi^2}$). Whenever a new answer is revealed, it is inserted into the list of possible answers, and its pairwise tests given mistake probabilities from the unused portion of the sequence. In principle, PPR-Adaptive can accommodate any number of answers, incurring only a small increase in sample complexity, as visible from Figure 3b.

6 CONCLUSION

In this paper, we apply the framework of PPR Martingale confidence sequences to the problem of PAC mode estimation. Our investigation follows two different dimensions that play a significant role in determining the efficiency of stopping rules. First is the tightness of the confidence bounds used internally in the stopping rule. By separately focusing on the Bernoulli case, we show that the PPR Martingale stopping rule is sample-efficient. The second aspect of mode estimation is the template applied to generalise from $K = 2$ to $K \geq 2$, which can typically be applied with any valid confidence bounds. Of the three major choices—"one-versus-one" (1v1), "one-versus-rest" (1vr), and a multi-dimensional test (MD)—we find 1v1 to be the most efficient. Our empirical findings are affirmed by theoretical analysis that shows (1) regardless of the problem instance and the mistake probability, the 1v1 approach is guaranteed to terminate no later than the 1vr approach for many popular confidence bounds, and (2) the PPR-1v1 stopping rule is indeed asymptotically optimal. The PPR-1v1 algorithm is also parameter-free and computationally much faster than the other algorithms, making it a natural choice to apply to practical mode estimation problems. We illustrate its efficacy on two distinct real-world tasks.

Our paper opens several directions to explore in future work, including the application of the PPR martingale test on pure exploration problems in stochastic bandits and Markov Decision Problems. It could also be of much practical benefit to incorporate the PPR martingale test (in place of existing ones) in large-scale applications of sampling and decision making.

Acknowledgements

The authors thank anonymous reviewers for providing valuable suggestions. Shivaram Kalyanakrishnan was partially supported by SERB grant ECR/2017/002479.

References

Vitalik Buterin. A next-generation smart contract and decentralized application platform, 2014. URL `https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf`.

R. Chen, J.-M. Park, and K. Bian. Robust distributed spectrum sensing in cognitive radio networks. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 1876–1884. IEEE Press, 2008. doi: 10.1109/INFOCOM.2008.251.

Sourav Das, Vinay Joseph Ribeiro, and Abhijeet Anand. YODA: enabling computationally intensive contracts on blockchains with Byzantine and Selfish nodes. In *26th Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2019.

David Galvin. Three tutorial lectures on entropy and counting. *arXiv preprint arXiv:1406.7872*, 2014.

Aurélien Garivier. Informational confidence bounds for self-normalized averages and applications. In *2013 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE press, 2013. doi: 10.1109/ITW.2013.6691311.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027. PMLR, 2016.

Kenny C. Gross and Keith E. Humenik. Sequential probability ratio test for nuclear plant component surveillance. *Nuclear Technology*, 93(2):131–137, 1991. doi: 10.13182/NT91-A34499.

Björn Haddenhorst, Viktor Bengs, and Eyke Hüllermeier. Identification of the generalized Condorcet winner in multi-dueling bandits. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=omDF-uQ_OZ`.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform Chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17:257–317, 2020.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil' ucb : An optimal exploration algorithm for multi-armed bandits. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 423–439. PMLR, 2014.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*, page 227–234. Omnipress, 2012.

Rajeeva Karandikar. Power and limitations of opinion polls in the context of Indian parliamentary democracy. In *Special Proceeding of 20th Annual Conference of SSCA*, pages 09 – 16. Society of Statistics, Computer and Applications, 2018.

Rajeeva L. Karandikar, Clive Payne, and Yogendra Yadav. Predicting the 1998 Indian parliamentary election. *Electoral Studies*, 21(1):69–89, 2002. doi: https://doi.org/10.1016/S0261-3794(00)00042-1.

Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 228–251. PMLR, 2013.

Gurmeet Singh Manku and Rajeev Motwani. Approximate frequency counts over data streams. In *Proceedings of the 28th International Conference on Very Large Data Bases*, page 346–357. VLDB Endowment, 2002.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. In *Proceedings of the 22nd Conference on Learning Theory (COLT 2009)*, 2009. URL `http://www.cs.mcgill.ca/~colt2009/papers/012.pdf#page=1`.

Wolfgang Mulzer. Five proofs of Chernoff's bound with applications. *CoRR*, abs/1801.03365, 2018.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009. URL `http://www.bitcoin.org/bitcoin.pdf`.

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

Clive Payne. Election forecasting in the UK: The BBC's experience. *Euramerica*, 33(1):193–234, 2003.

Dhruti Shah, Tuhinangshu Choudhury, Nikhil Karamchandani, and Aditya Gopalan. Sequential mode estimation with oracle queries. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34, pages 5644–5651. AAAI Press, 2020.

A. Wald. Sequential Tests of Statistical Hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

Ian Waudby-Smith and Aaditya Ramdas. Confidence sequences for sampling without replacement. In *Advances in Neural Information Processing Systems*, volume 33, pages 20204–20214. Curran Associates, Inc., 2020.

Supplementary Material: PAC Mode Estimation using PPR Martingale Confidence Sequences

A IMPLEMENTATION DETAILS

We specify the various baseline algorithms used in our comparisons in sections 2 and 3. For actual code see Appendix B.

Whether the implementation is 1v1 or 1vr, each algorithm has an atomic operation to maintain upper and lower confidence bounds on the parameter of a Bernoulli distribution.

- In the 1v1 approach, there is a separate Bernoulli associated with each pair of values v_i and v_j for $1 \leq i < j \leq k$ (hence a Bernoulli with mean $\frac{p_i}{p_i+p_j}$). Each confidence bound is drawn with mistake probability $\frac{\delta}{K-1}$, which ensures δ-correctness upon termination.

- In the 1vr approach, there is a separate Bernoulli associated with each value v_i for $i \leq i \leq k$, taking $\neg v_i$ as its other outcome (hence a Bernoulli with mean p_i). To ensure δ-correctness upon termination, the mistake probability used is $\frac{\delta}{K}$.

The actual confidence bounds used in different algorithms are listed below. In 1v1, the total number of samples (t) shown in the confidence bounds is to be taken as the sum of the number of occurrences of the corresponding (v_i, v_j) pair in question, while under 1vr, it is the total number of samples of $\mathcal{P}$ observed yet. Below we denote the empirical mean after t samples $\hat{p}^t$, and the mistake probability associated with the confidence bound δ.

A.1 $\mathcal{A}_1$ (Shah et al., 2020)

We implement the algorithm as given in the original paper (Shah et al., 2020), which defines a confidence width

$$\beta(t,\delta) = \sqrt{\frac{2V^t \ln(4t^2/\delta)}{t}} + \frac{7\ln(4t^2/\delta)}{3(t-1)}$$

after t samples, where V^t is the empirical variance of the samples. Lower and upper confidence bounds are given by $\hat{p}^t \pm \beta(t,\delta)$.

A.2 LUCB (Kalyanakrishnan et al., 2012) and KL-LUCB (Kaufmann and Kalyanakrishnan, 2013)

For LUCB (Kalyanakrishnan et al., 2012) and KL-LUCB (Kaufmann and Kalyanakrishnan, 2013), the "exploration rate" used is $\beta(t,\delta) = \ln(\frac{405.5t^{1.1}}{\delta})$, which ensures δ correctness when a union bound over t is performed (Kaufmann and Kalyanakrishnan, 2013).

LUCB sets its lower and upper confidence bounds as $\hat{p}^t \pm \sqrt{\frac{\beta(t,\delta)}{2t}}$. KL-LUCB obtains them by performing a numerical computation to obtain the lower and upper confidence bounds as given below.

$$\begin{aligned}
\text{KL-LUCB lower confidence bound } &= \min\{q \in [0, \hat{p}^t] : t \times D_{KL}(\hat{p}^t || q) \leq \beta(t,\delta)\}; \\
\text{KL-LUCB upper confidence bound } &= \max\{q \in [\hat{p}^t, 1] : t \times D_{KL}(\hat{p}^t || q) \leq \beta(t,\delta)\}.
\end{aligned}$$

A.3 KL-SN (Garivier, 2013)

KL-SN (Garivier, 2013) uses a more sophisticated exploration rate so as to avoid a naïve union bound over time. We first find $\gamma > 1$ that satisfies the following equation, and thereafter set the exploration rate as given below.

$$2e^2\gamma e^{-\gamma} = \delta \text{ and } \beta(t,\delta) = \frac{\gamma(1+\ln(\gamma))}{(\gamma-1)\ln(\gamma)}\ln(\ln(t)) + \gamma.$$

Using the above exploration rate, the confidence bounds are constructed in the same way as KL-LUCB.

$$\begin{aligned}\text{KL-SN lower confidence bound } &= \min\{q \in [0, \hat{p}^t] : t \times D_{KL}(\hat{p}^t || q) \le \beta(t,\delta)\};\\ \text{KL-SN upper confidence bound } &= \max\{q \in [\hat{p}^t, 1] : t \times D_{KL}(\hat{p}^t || q) \le \beta(t,\delta)\}.\end{aligned}$$

B CODE DETAILS

The code used to run our experiments (from sections 2, 3, 5.1, and 5.2) is at `https://github.com/rohanshah13/pac_mode_estimation`. Below we provide details on the running time of our algorithms on a couple of problem instances, which are indicative of their relative order in general.

The results in Table 1 were obtained by performing each of the experiments for 100 iterations, with mistake probability $\delta = 0.01$. For the last 3 rows, for all algorithms except KL-SN 1v1, 200 iterations were used so as to reduce the error bars. A running-time comparison of all the algorithms across the 100 runs is given in Table 3. The runs were performed on an Intel Core i7-8750H CPU @ 2.20GHz processor.

Table 3: Average running time and one standard error of mode estimation algorithms across 100 iterations, distribution $\mathcal{P}_6 = (.35, .33, .04 \times 8)$ from Table 1.

Algorithm	Type	Run time (seconds)
$\mathcal{A}_1$	1vr	9.01 ± 0.2
	1v1	6.65 ± 0.17
KL-SN	1vr	169.47 ± 5.76
	1v1	2.74 ± 0.10
PPR	1vr	51.5 ± 2.37
	1v1	**1.28 ± 0.06**

The results in Table 2 and Table 4 were obtained by performing each of the experiments for 10 random seeds and setting the mistake probability to 0.01. The sampling of the votes from each of the constituencies were done with a batch size of 200. We used Google's colaboratory services [4] for performing these experiments.

[4] `https://colab.research.google.com/`

C RELATING THE TERMINATION OF 1V1, 1VR, MD ALGORITHMS

In this section, we provide a proof of Lemma 2 and also comment on the plausible applicability of the result to concentration bounds not covered by the lemma. Since the actual working is relatively verbose, we divide the appendix into separate subsections. In Section C.1, we provide a relatively straightforward proof that the 1v1 versions of LUCB and $\mathcal{A}_1$ always terminate before their corresponding 1vr versions. In Section C.2, we work towards proving that PPR-1v1 always terminates before PPR-1vr, and we reduce this claim to an inequality on beta functions, which we conjecture to be true, and have verified empirically for a range of values. In Section C.3, we consider the multi-dimensional version of the PPR martingale stopping rule (denoted PPR-MD), which serves as an alternative for designing a stopping rule for mode estimation. This rule would use a Dirichlet prior (which is the conjugate of the multinomial distribution), and is analogous to the mutlti-variate PPR considered in Appendix C of Waudby-Smith and Ramdas (2020). We show that PPR-MD always stops after PPR-1v1 on every run. In Section C.4, we provide a simple proof that shows $\mathcal{A}$1-1v1 (Shah et al., 2020) always terminates after PPR-1v1 on every run.

In summary, sections C.1, C.3 and C.4 complete the proof of Lemma 2, while Section C.2 concludes with a conjecture, which if true, would mean the termination of PPR-1vr implies the termination of PPR-1v1.

C.1 LUCB and $\mathcal{A}_1$ Algorithms

LUCB: Suppose we run both the variants of LUCB algorithm with $\delta' = \frac{\delta}{K}$, where δ is the mistake probability (in reality, we run the 1v1 with $\frac{\delta}{K-1}$; so it is even better). We will show that LUCB-1vr termination implies LUCB-1v1 termination. The LUCB algorithm can differentiate some $i, j \in \{1, 2, .., K\}$ such that $i \neq j$ when

$$\hat{p}_i^t - \sqrt{\frac{\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}{2t}} \geq \hat{p}_j^t + \sqrt{\frac{\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}{2t}},$$

where $\hat{p}_i^t$ denotes the empirical mean corresponds to observation i. For 1v1, the corresponding condition for differentiating i, j will be

$$\frac{\hat{p}_i^t}{\hat{p}_i^t + \hat{p}_j^t} - \sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}} \geq \frac{\hat{p}_j^t}{\hat{p}_i^t + \hat{p}_j^t} + \sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}},$$

where s_{ij}^t denotes the total number of samples coming from p_i and p_j. Thus, to prove 1vr termination implies 1v1 termination, It is sufficient to prove that: If it is true that

$$\hat{p}_i^t - \sqrt{\frac{\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}{2t}} \geq \hat{p}_j^t + \sqrt{\frac{\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}{2t}}$$

then, it is also true that

$$\frac{\hat{p}_i^t}{\hat{p}_i + \hat{p}_j^t} - \sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}} \geq \frac{\hat{p}_j^t}{\hat{p}_i^t + \hat{p}_j} + \sqrt{\frac{\ln(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'})}{2s_{ij}^t}}.$$

We have

$$\hat{p}_i^t - \hat{p}_j^t \geq 2\sqrt{\frac{\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}{2t}},$$

$$\implies \frac{\hat{p}_i^t - \hat{p}_j^t}{2(\hat{p}_i^t + \hat{p}_j^t)\sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}}} \geq \frac{\sqrt{\frac{\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}{2t}}}{(\hat{p}_i^t + \hat{p}_j^t)\sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}}}.$$

Using the fact that $\hat{p}_i^t + \hat{p}_j^t = \frac{s_{ij}^t}{t}$ and the above inequality, we get

$$\frac{\hat{p}_i^t - \hat{p}_j^t}{2(\hat{p}_i^t + \hat{p}_j^t)\sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}}} \geq \frac{\sqrt{t\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}}{\sqrt{s_{ij}^t \ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}}.$$

Since $\sqrt{t\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}$ is an increasing function in t, and we know that $s_{ij}^t \leq t$, we have

$$\frac{\hat{p}_i^t - \hat{p}_j^t}{2(\hat{p}_i^t + \hat{p}_j^t)\sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}}} \geq \frac{\sqrt{t\ln\left(\frac{405.5t^{1.1}}{\delta'}\right)}}{\sqrt{s_{ij}^t \ln(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'})}} \geq 1,$$

$$\implies \frac{\hat{p}_i^t}{\hat{p}_i^t + \hat{p}_j^t} - \sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}} \geq \frac{\hat{p}_j^t}{\hat{p}_i^t + \hat{p}_j^t} + \sqrt{\frac{\ln\left(\frac{405.5(s_{ij}^t)^{1.1}}{\delta'}\right)}{2s_{ij}^t}}.$$

Hence, we showed that the 1v1 variant of LUCB algorithm always terminates before its 1vr variant.

$\mathcal{A}_1$ **Algorithm:** Suppose we run both the variants of $\mathcal{A}_1$ algorithm with $\delta' = \frac{\delta}{K}$, where δ is the mistake probability (As above, we actually run 1v1 with $\frac{\delta}{K-1}$, so 1v1 will terminate even faster). The $\mathcal{A}_1$ algorithm differentiates between $i, j \in \{1, 2, .., K\}$ such that $i \neq j$ and i is the winner when

$$\hat{p}_i^t - \sqrt{\frac{2V_t(Z^i)\ln\left(\frac{4t^2}{\delta'}\right)}{t}} - \frac{7\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)} \geq \hat{p}_j^t + \sqrt{\frac{2V_t(Z^j)\ln\left(\frac{4t^2}{\delta'}\right)}{t}} + \frac{7\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)},$$

where

$$V_t(Z^i) = \frac{s_i^t(t - s_i^t)}{t(t-1)},$$

in which s_i^t denotes the total number of samples coming from v_i. Similarly, the condition for there to be a winner between i, j in $\mathcal{A}_1$ 1v1 will be:

$$\frac{\hat{p}_i^t}{\hat{p}_i^t + \hat{p}_j^t} - \sqrt{\frac{2V_{s_{ij}^t}(Z^i)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{s_{ij}^t}} - \frac{7\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{3(s_{ij}^t - 1)} \geq \frac{\hat{p}_j^t}{\hat{p}_i^t + \hat{p}_j^t} + \sqrt{\frac{2V_{s_{ij}^t}(Z^j)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{s_{ij}^t}} + \frac{7\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{3(s_{ij}^t - 1)},$$

where $s_{ij}^t = s_i^t + s_j^t$ denotes the total number of samples coming from v_i and v_j, and s_i^t denotes the number of samples coming from v_i.
To prove 1vr termination implies 1v1 termination, It is sufficient to prove that: If it is true that

$$\hat{p}_i^t - \sqrt{\frac{2V_t(Z^i)\ln\left(\frac{4t^2}{\delta'}\right)}{t}} - \frac{7\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)} \geq \hat{p}_j^t + \sqrt{\frac{2V_t(Z^j)\ln\left(\frac{4t^2}{\delta'}\right)}{t}} + \frac{7\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)}$$

then, it is also true that

$$\frac{\hat{p}_i^t}{\hat{p}_i^t + \hat{p}_j^t} - \sqrt{\frac{2V_{s_{ij}^t}(Z^i)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{s_{ij}^t}} - \frac{7\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{3(s_{ij}^t - 1)} \geq \frac{\hat{p}_j^t}{\hat{p}_i^t + \hat{p}_j^t} + \sqrt{\frac{2V_{s_{ij}^t}(Z^j)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{s_{ij}^t}} + \frac{7\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{3(s_{ij}^t - 1)}$$

Using the same steps as in LUCB, we get

$$\frac{\hat{p}_i^t - \hat{p}_j^t}{2(\hat{p}_i^t + \hat{p}_j^t)\left(\sqrt{\frac{2V_{s_{ij}^t}(Z^i)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{s_{ij}^t}} + \sqrt{\frac{2V_{s_{ij}^t}(Z^j)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{s_{ij}^t}} + \frac{14\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{3(s_{ij}^t-1)}\right)}$$

$$\geq \frac{\left(\sqrt{2tV_t(Z^i)\ln\left(\frac{4t^2}{\delta'}\right)} + \sqrt{2tV_t(Z^j)\ln\left(\frac{4t^2}{\delta'}\right)} + \frac{14t\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)}\right)}{\left(\sqrt{2s_{ij}^t V_{s_{ij}^t}(Z^i)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)} + \sqrt{2s_{ij}^t V_{s_{ij}^t}(Z^j)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)} + \frac{14s_{ij}^t\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{3(s_{ij}^t-1)}\right)}.$$

We need to prove that the above expression is ≥ 1. We note that $\sqrt{2tV_t(Z^i)\ln\left(\frac{4t^2}{\delta'}\right)}$, $\sqrt{2tV_t(Z^j)\ln\left(\frac{4t^2}{\delta'}\right)}$ are increasing functions in t. Let $f(t) = \frac{14t\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)}$. We write down the differential:

$$f'(t) \propto 2(t-1) - \ln\left(\frac{4t^2}{\delta}\right).$$

Thus, for t such that $2(t-1) \geq \ln\left(\frac{4t^2}{\delta}\right)$, $f(t)$ is an increasing function. We now look at the minimum value of s_i^t possible, and we show that s_i^t is always such that $f(t)$ is an increasing function for $t \geq s_i^t$. We proceed to look at a scenario in which lowest possible value for s_i^t is observed. Consider the case in which we observe $t\hat{p}_i^t$ samples with value v_i (thus, the empirical mode is $\hat{p}_i^t$ for v_i). Let's assume that $\mathcal{A}_1$ 1vr algorithm declares i as the mode after observing $t\hat{p}_i^t$ samples of v_i. We assume that j has 0 samples, and then find the value of t that arises. Termination of $\mathcal{A}_1$ algorithm after t observations suggests (we ignore the empirical mean term, as it makes our lower bound on t only bigger):

$$\hat{p}_i^t - \frac{7\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)} \geq 0 + \frac{7\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)}$$

$$\implies \hat{p}_i^t \geq \frac{14\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)}.$$

Using the fact that $t\hat{p}_i^t = s_i^t$, we arrive at:

$$\frac{3\hat{p}_i^t(t-1)}{14} \geq \ln\left(\frac{4t^2}{\delta'}\right)$$

$$\implies \frac{3(s_i^t - \hat{p}_i^t)}{14} \geq \ln\left(\frac{4t^2}{\delta'}\right) \geq \ln\left(\frac{4(s_i^t)^2}{\delta'}\right).$$

To show $2(t_i - 1) \geq \ln\left(\frac{4(s_i^t)^2}{\delta}\right)$, it is sufficient to show that $2(s_i^t - 1) \geq \frac{3}{14}(s_i^t - \hat{p}_i^t)$. For proving this, we make a mild assumption that $s_i^t \geq 2$ as *practically* $\mathcal{A}_1$ algorithm would need more than 2 samples from v_i to declare v_i as the mode. Under this assumption, we get:

$$2(s_i^t - 1) - \frac{3}{14}(s_i^t - \hat{p}_i^t) = \frac{25}{14}s_i^t - 2 + \frac{3}{14}\hat{p}_i^t \geq \frac{11}{7} + \frac{3}{14}\hat{p}_i^t > 0.$$

Hence, we note that

$$\frac{\left(\sqrt{2tV_t(Z^i)\ln\left(\frac{4t^2}{\delta'}\right)} + \sqrt{2tV_t(Z^j)\ln\left(\frac{4t^2}{\delta'}\right)} + \frac{14t\ln\left(\frac{4t^2}{\delta'}\right)}{3(t-1)}\right)}{\left(\sqrt{2s_{ij}^t V_{s_{ij}^t}(Z^i)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)} + \sqrt{2s_{ij}^t V_{s_{ij}^t}(Z^j)\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)} + \frac{14s_{ij}^t\ln\left(\frac{4(s_{ij}^t)^2}{\delta'}\right)}{3(s_{ij}^t-1)}\right)} \geq 1$$

since all the 3 terms are increasing, and thus $\mathcal{A}_1$ 1vr always stops after $\mathcal{A}_1$ 1v1.

C.2 PPR

Suppose party i, j have confidence sequences which do not intersect at some time t. Let the number of samples from party i be s_i^t, and the number of samples from party j be s_j^t, and without loss of generality assume $s_i^t > s_j^t$. Let f be the number of samples from parties which are not i or j. Then, the condition for party i, j having confidence sequences which do not intersect can be written as

$$\exists\, \theta \in \left[\frac{s_j^t}{t}, \frac{s_i^t}{t}\right] \text{ such that}$$
$$\frac{1}{\text{Beta}\left(\theta; s_i^t+1, t-s_i^t+1\right)} \geq \frac{K}{\delta},$$
$$\frac{1}{\text{Beta}\left(\theta; s_j^t+1, t-s_j^t+1\right)} \geq \frac{K}{\delta}.$$

Parties i, j have a winner between them in PPR-1v1 when

$$\frac{1}{\text{Beta}\left(\frac{1}{2}; s_i^t+1, s_j^t+1\right)} \geq \frac{K-1}{\delta}.$$

We will attempt to show that party i, j having disjoint confidence sequences in PPR-1vr implies party i, j having a winner between them in PPR-1v1. This will imply that PPR-1v1 always terminates before PPR-1vr. Noting that

$$\text{Beta}\left(\theta; s_i^t, s_j^t\right) = \frac{\theta^{s_i^t-1}(1-\theta)^{s_j^t-1}}{B(s_i^t, s_j^t)}$$
$$\text{where } B(s_i^t, s_j^t) = \frac{\Gamma(s_i^t)\Gamma(s_j^t)}{\Gamma(s_i^t+s_j^t)},$$

we are given that

$$\frac{\delta}{K} \geq \frac{\theta^{s_i^t}(1-\theta)^{t-s_i^t}}{B(s_i^t+1, t-s_i^t+1)},$$
$$\frac{\delta}{K} \geq \frac{\theta^{s_j^t}(1-\theta)^{t-s_j^t}}{B(s_j^t+1, t-s_j^t+1)}.$$

And we wish to show that

$$\frac{\delta}{K-1} \geq \frac{1}{2^{s_i^t+s_j^t}B(s_i^t+1, s_j^t+1)}.$$

We denote

$$\frac{\theta^{s_i^t}(1-\theta)^{t-s_i^t}}{B(s_i^t+1, t-s_i^t+1)} = L_1(\theta),$$
$$\frac{\theta^{s_j^t}(1-\theta)^{t-s_j^t}}{B(s_j^t+1, t-s_j^t+1)} = L_2(\theta).$$

Thus, it is sufficient to show that, $\forall\, \theta \in \left[\frac{s_j^t}{t}, \frac{s_i^t}{t}\right]$,

$$\max\left(L_1(\theta), L_2(\theta)\right) \geq \frac{K-1}{K2^{s_i^t+s_j^t}B(s_i^t+1, s_j^t+1)}.$$

Consider

$$L_1'(\theta) \propto s_i^t(1-\theta) - (t-s_i^t)\theta,$$
$$L_1'(\theta) \propto s_i^t - t\theta$$
$$\implies L_1'(\theta) \geq 0\, \forall\, \theta \leq \frac{s_i^t}{t}.$$

Similarly, consider

$$\begin{gathered} L_2'(\theta) \propto s_j^t(1-\theta) - (t - s_j^t)\theta \\ L_2'(\theta) \propto s_j^t - t\theta, \\ \implies L_2'(\theta) \leq 0 \,\forall\, \theta \geq \frac{s_j^t}{t}. \end{gathered}$$

Thus, we note that in our range of $\theta, L_1(\theta)$ is increasing and $L_2(\theta)$ is decreasing. Consider θ^* such that $L_1(\theta^*) = L_2(\theta^*)$. We will prove that considering only the value $\theta = \theta^*$ is sufficient to prove that

$$\max\left(L_1(\theta), L_2(\theta)\right) \geq \frac{K-1}{K2^{s_i^t+s_j^t}B(s_i^t+1, s_j^t+1)}$$

for the entire range of $\theta \in \left[\frac{s_j^t}{t}, \frac{s_i^t}{t}\right]$. First, we prove that θ^* lies in this range itself. The equation $L_1(\theta^*) = L_2(\theta^*)$ gives us the implicit equation

$$\left(\frac{\theta^*}{1-\theta^*}\right) = \left[\frac{s_i^t!(t-s_i^t)!}{s_j^t!(t-s_j^t)!}\right]^{\frac{1}{s_i^t - s_j^t}}.$$

We first prove $\theta^* \geq \frac{s_j^t}{t}$. This is the same as showing that $\frac{\theta^*}{1-\theta^*} \geq \frac{s_j^t}{t - s_j^t}$. We show that

$$\begin{aligned} \frac{s_j^t}{t-s_j^t} &\leq \left[\frac{s_i^t!(t-s_i^t)!}{s_j^t!(t-s_j^t)!}\right]^{\frac{1}{s_i^t - s_j^t}} \\ \Leftrightarrow \left(\frac{s_j^t}{t-s_j^t}\right)^{s_i^t - s_j^t} &\leq \left[\frac{(s_j^t+1)(s_j^t+2)...(t-s_i^t)}{(s_i^t+1)(s_i^t+2)...(t-s_j^t)}\right]. \end{aligned}$$

To prove the above inequality, first note that

$$\frac{s_j^t}{t-s_j^t} \leq \frac{s_j^t+1}{t-s_j^t}.$$

Hence, it is sufficient now to show

$$\left(\frac{s_j^t}{t-s_j^t}\right)^{s_i^t - s_j^t - 1} \leq \left[\frac{(s_j^t+2)...(t-s_i^t)}{(s_i^t+1)(s_i^t+2)...(t-s_j^t-1)}\right].$$

We know that $s_j^t < s_i^t$. Suppose that $s_j^t = s_i^t - 1$; then the above equation has both LHS = 1 and RHS = 1, hence it is true. Otherwise, if $s_j^t < s_i^t - 1$, we have

$$\frac{s_j^t}{t-s_j^t} \leq \frac{s_j^t+2}{t-s_j^t-1}.$$

And proceeding in a similar fashion, we can show inductively that

$$\frac{s_j^t}{t-s_j^t} \leq \left[\frac{s_i^t!(t-s_i^t)!}{s_j^t!(t-s_j^t)!}\right]^{\frac{1}{s_i^t - s_j^t}}.$$

In the same way, we can also show that

$$\frac{s_i^t}{t-s_i^t} \geq \left[\frac{s_i^t!(t-s_i^t)!}{s_j^t!(t-s_j^t)!}\right]^{\frac{1}{s_i^t - s_j^t}}.$$

This proves that $\theta^* \in \left[\frac{s_j^t}{t}, \frac{s_i^t}{t}\right]$. Moreover, since $L_1(\theta)$ is increasing and $L_2(\theta)$ is decreasing, we have that

$$\begin{aligned} L_1(\theta^*) &\leq L_1(\theta), \theta \geq \theta^*, \\ L_2(\theta^*) &\geq L_2(\theta), \theta \leq \theta^*. \end{aligned}$$

Combining the above two equations and noting that $L_1(\theta^*) = L_2(\theta^*)$, we get that

$$\max\left(L_1(\theta^*), L_2(\theta^*)\right) \leq \max\left(L_1(\theta), L_2(\theta)\right), \theta \in \left[\frac{s_j^t}{t}, \frac{s_i^t}{t}\right].$$

Thus, if we show that

$$\max\left(L_1(\theta^*), L_2(\theta^*)\right) \geq \frac{K-1}{K 2^{s_i^t + s_j^t} B(s_i^t + 1, s_j^t + 1)}$$

we are done. We need to show that

$$\frac{(\theta^*)^{s_i^t} (1-\theta^*)^{t - s_i^t}}{B(s_i^t + 1, t - s_i^t + 1)} \geq \frac{K-1}{K 2^{s_i^t + s_j^t} B(s_i^t + 1, s_j^t + 1)},$$

$$\text{where } \left(\frac{\theta^*}{1-\theta^*}\right) = \left[\frac{s_i^t!(t - s_i^t)!}{s_j^t!(t - s_j^t)!}\right]^{\frac{1}{s_i^t - s_j^t}}.$$

We ran computer simulations to verify the correctness of the above equation exhaustively for all values of s_i^t, s_j^t, t in $[1, 200]$. Thus, we conjecture that the above equation is true for all $s_i^t, s_j^t, t > 0$ such that $s_i^t > s_j^t$.

C.3 Comparison with Multidimensional PPR (PPR-MD)

The Dirichlet distribution is the conjugate prior of the Multinomial distribution, and hence can be used to implement a PPR-based stopping rule, which is similar in principle to the multi-variate PPR given in Appendix C in Waudby-Smith and Ramdas (2020). In specific, the PDF of the Dirichlet distribution is given as

$$\begin{aligned} pdf(\bar{p}) &= \frac{\prod_{i=1}^{K} \bar{p}_i^{\alpha_i - 1}}{B(\alpha)}, \\ B(\alpha) &= \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}. \end{aligned}$$

At each time-step t, we maintain a confidence set C^t with $\bar{p} \in [0,1]^k$, such that $C^t \overset{\text{def}}{=} \{\bar{p} : R^t(\bar{p}) < \frac{1}{\delta}\}$. We stop at time t when all the $\bar{p} \in C^t$ have the same unique mode. We call this stopping rule PPR-MD (Multi-dimensional). We note that, with prior $\alpha = [1, ..., 1]$,

$$R^t(\bar{p}) = \frac{1}{(K-1)!} \times \frac{B(\alpha)}{\prod_{i=1}^{K} \bar{p}_i^{\alpha_i - 1}}$$

We will show that if the PPR-MD stopping rule stops at a point, then PPR-1v1 would also have stopped. In other words, PPR-MD stopping *implies* PPR-1v1 stopping on every run.

Let the number of observations from $v_1, v_2, ..., v_K$ until timestep t be $s_1^t, s_2^t, ..., s_K^t$. Without loss of generality, order them such that $s_1^t \geq s_2^t \geq ... \geq s_K^t$. We denote the empirical means as $\hat{p}_i^t = \frac{s_i^t}{t}$. We note that, when PPR-MD stops, $x_1 = (\frac{\hat{p}_1^t + \hat{p}_2^t}{2}, \frac{\hat{p}_1^t + \hat{p}_2^t}{2}, \hat{p}_3^t, ..., \hat{p}_K^t)$ must not lie in the confidence set. Explicitly,

$$\frac{\left(\frac{\hat{p}_1^t + \hat{p}_2^t}{2}\right)^{s_1^t + s_2^t} (\hat{p}_3^t)^{s_3^t} ... \hat{p}_K^{s_K^t}}{s_1^t! s_2^t! ... s_K^t!} \leq \frac{\delta(s_1^t + s_2^t + \ldots + s_K^t + K - 1)!}{(K-1)!}.$$

We divide our proof into two parts. First, we show the implication for $K = 3$, and then we extend our proof to general $K \geq 3$.

C.3.1 Proof for $K = 3$

When $K = 3$, PPR-1v1 stops when

$$\frac{\left(\frac{1}{2}\right)^{s_1^t+s_2^t}(s_1^t+s_2^t+1)!}{s_1^t!s_2^t!} \leq \frac{\delta}{2}$$

and PPR-MD stopping implies that (using that $K = 3$)

$$\frac{\left(\frac{\hat{p}_1^t+\hat{p}_2^t}{2}\right)^{s_1^t+s_2^t}(\hat{p}_3^t)^{s_3^t}(s_1^t+s_2^t+s_3^t+2)!}{s_1^t!s_2^t!s_3^t!} \leq \frac{\delta}{2}.$$

If we can show that

$$\frac{\left(\frac{\hat{p}_1^t+\hat{p}_2^t}{2}\right)^{s_1^t+s_2^t}(\hat{p}_3^t)^{s_3^t}(s_1^t+s_2^t+s_3^t+2)!}{s_1^t!s_2^t!s_3^t!} \geq \frac{\left(\frac{1}{2}\right)^{s_1^t+s_2^t}(s_1^t+s_2^t+1)!}{s_1^t!s_2^t!},$$

for all s_1^t, s_2^t, s_3^t such that $s_1^t \geq s_2^t \geq s_3^t$, then PPR-MD stopping will imply PPR-1v1 stopping for $K = 3$.

We note that this is equivalent to showing

$$\begin{aligned}
(\hat{p}_1^t+\hat{p}_2^t)^{s_1^t+s_2^t}(\hat{p}_3^t)^{s_3^t} &\geq \frac{s_3^t!(s_1^t+s_2^t+1)!}{(s_1^t+s_2^t+s_3^t+2)!} \\
\Leftrightarrow \frac{(s_1^t+s_2^t)^{s_1^t+s_2^t}(s_3^t)^{s_3^t}}{(s_1^t+s_2^t+s_3^t)^{s_1^t+s_2^t+s_3^t}} &\geq \frac{s_3^t!(s_1^t+s_2^t+1)!}{(s_1^t+s_2^t+s_3^t+2)!} \\
\Leftrightarrow \frac{x^x y^y}{(x+y)^{x+y}} &\geq \frac{(x+1)!y!}{(x+y+2)!}, \text{ where } x = s_1^t+s_2^t, y = s_3^t
\end{aligned}$$

for all x, y, such that $y \geq 0, x \geq 2y$. Note that, when $y = 0$, the y^y term tends to 1; hence, the above statement is true. Subsequently, we will consider $y > 0$. Let

$$f(x,y) = \frac{x^x y^y (x+2)....(x+y+2)}{y!(x+y)^{x+y}}$$

We want to show that

$$f(x,y) \geq 1 \text{ when } y > 0, x \geq 2y.$$

We show this in two steps. First, we show that $f(x,y)$ is an increasing function of x when y is fixed. Since $f(x,y)$ is increasing in x, we then choose the minimum value of x with fixed y (i.e, $x = 2y$) and prove that $f(2y,y) \geq 1$ for all $y > 0$, which implies that $f(x,y) \geq 1$ for all $y > 0, x \geq 2y$.

Showing that $f(x,y)$ is Increasing in x

Recapping, we need to show that

$$f(x,y) = \frac{y^y(x+2)...(x+y+2)}{y!(x+y)^y\left(1+\frac{y}{x}\right)^x}$$

is increasing with respect to x while keeping y fixed. We write down the partial derivative of $\ln(f(x,y))$ with respect to x as follows:

$$\frac{\partial \ln(f(x,y))}{\partial x} = \sum_{j=2}^{y} \frac{1}{x+j} - \frac{y}{x+y} + \frac{1}{x+y+2} + \frac{1}{x+y+1} + 1 - \frac{x}{x+y} - \ln\left(\frac{x+y}{x}\right)$$

$$\implies \frac{\partial \ln(f(x,y))}{\partial x} = \sum_{j=2}^{y} \frac{1}{x+j} + \frac{1}{x+y+2} + \frac{1}{x+y+1} - \ln\left(\frac{x+y}{x}\right).$$

We note that

$$\frac{1}{x+j} \geq \int_{w=j}^{w=j+1} \frac{1}{x+w} dw = \ln\left(\frac{x+j+1}{x+j}\right)$$
$$\implies \sum_{j=2}^{y} \frac{1}{x+j} \geq \ln\left(\frac{x+y+1}{x+2}\right).$$

Thus, if we show that

$$g(x,y) = \ln\left(\frac{x+y+1}{x+2}\right) + \frac{1}{x+y+2} + \frac{1}{x+y+1} - \ln\left(\frac{x+y}{x}\right) \geq 0 \text{ when } y > 0, x \geq 2y$$

this is sufficient to show that $\frac{\partial \ln(f(x,y))}{\partial x} \geq 0$ when $y \geq 0, x \geq 2y$. We combine terms to get

$$g(x,y) = \frac{1}{x+y+2} + \frac{1}{x+y+1} - \ln\left(\frac{(x+y)(x+2)}{(x+y+1)x}\right)$$

We will now show that keeping x fixed, $g(x,y)$ is a decreasing function of y. Hence, it will be sufficient to consider the maximum possible value of y to show that $g(x,y) \geq 0$. We see that

$$\frac{\partial g(x,y)}{\partial y} = -\left(\frac{3x^3 + x^2(9y+11) + x(9y^2+22y+13) + 3y^3 + 11y^2 + 13y + 4}{(x+y)(x+y+1)^2(x+y+2)^2}\right)$$
$$\implies \frac{\partial g(x,y)}{\partial y} \leq 0, y > 0, x \geq 2y.$$

Hence, $g(x,y)$ is decreasing in y. Thus, it is enough to consider the maximal possible value of y for a fixed x to show that $g(x,y) \geq 0$. For a fixed x, the maximum possible y is $\frac{x}{2}$. Hence, we have that

$$h(x) = g(x, x/2) = \frac{2}{3x+4} + \frac{2}{3x+2} - \ln\left(\frac{3(x+2)}{(3x+2)}\right),$$

and we need to show that $h(x) \geq 0$ when $x > 0$. We have

$$h'(x) = -\left(\frac{8(9x^2+21x+14)}{(x+2)(3x+2)^2(3x+4)^2}\right)$$
$$\implies h'(x) \leq 0 \text{ when } x > 0.$$

Thus, $h(x)$ is a decreasing function of x. We see that, as $x \to \infty, h(x) \to 0$. Thus, $h(x) \geq 0$ when $x > 0$. This means that $g(x, x/2) \geq 0 \implies g(x,y) \geq 0 \implies \frac{\partial f(x,y)}{\partial x} \geq 0$ which implies that $f(x,y)$ is an increasing function of x.

Showing that $f(x,y) \geq 1$

Since $f(x,y)$ is an increasing function of x when y is fixed, it is sufficient to show that $f(x,y) \geq 1$ for the minimum possible value of x. We note that the minimum possible value of x is $x = 2y$. We have

$$f(2y,y) - 1 = \frac{y^y(2y+2)...(3y+2) - y!(3y)^y\left(\frac{3}{2}\right)^{2y}}{y!(3y)^y\left(\frac{3}{2}\right)^{2y}}.$$

The denominator is positive. We show that the numerator is positive for all $y > 0$.

$$h(2y,y) = y^y(2y+2)...(3y+2) - y!(3y)^y\left(\frac{3}{2}\right)^{2y}$$
$$\implies h(2y,y) = y^y\left((2y+2)...(3y+2) - y!\left(\frac{27}{4}\right)^y\right).$$

Let

$$c(y) = (2y+2)...(3y+2) - y!\left(\frac{27}{4}\right)^y.$$

We will show that $c(y) > 0$ when $y > 0$. First, We note that $c(1) > 0$. We now use induction; assume that $c(y) > 0$ for some y. We will show that $c(y+1) > 0$. We have that

$$c(y) = (2y+2)...(3y+2) - y!\left(\frac{27}{4}\right)^y > 0$$
$$\implies d(y) = \frac{(2y+2)...(3y+2)}{y!\left(\frac{27}{4}\right)^y} > 1.$$

We want to show that

$$c(y+1) = (2y+4)...(3y+5) - (y+1)!\left(\frac{27}{4}\right)^{y+1} > 0$$
$$\Leftrightarrow d(y+1) = \frac{(2y+4)...(3y+5)}{(y+1)!\left(\frac{27}{4}\right)^{y+1}} > 1.$$

We will show that

$$\frac{d(y+1)}{d(y)} = \frac{(3y+3)(3y+4)(3y+5)}{(2y+2)(2y+3)(y+1)\frac{27}{4}} > 1$$
$$\Leftrightarrow (3y+3)(3y+4)(3y+5) - (2y+2)(2y+3)(y+1)\frac{27}{4} > 0.$$

We note that the last expression turns out to be a quadratic in y which is always positive when $y > 0$. Hence, $\frac{d(y+1)}{d(y)} > 1$ when $y > 0$. Thus, since by our induction hypothesis we have that $d(y) > 1, y > 0$ and by our proof above we have that $\frac{d(y+1)}{d(y)} > 1, y > 0$, we note that this implies $d(y+1) = d(y)\frac{d(y+1)}{d(y)} > 1$, which is what we wanted to show in our induction step. Hence, proved.

C.3.2 Extending to General K

For a general K, PPR-1v1 stops when

$$\frac{\left(\frac{1}{2}\right)^{s_1^t+s_2^t}(s_1^t+s_2^t+1)!}{s_1^t!s_2^t!} \leq \frac{\delta}{K-1}$$

and when PPR-MD stops, it is true that (by choosing $x = \left(\frac{\hat{p}_1^t+\hat{p}_2^t}{2}, \frac{\hat{p}_1^t+\hat{p}_2^t}{2}, \hat{p}_3^t, ...\hat{p}_K^t\right)$

$$\frac{\left(\frac{\hat{p}_1^t+\hat{p}_2^t}{2}\right)^{s_1^t+s_2^t}(\hat{p}_3^t)^{s_3^t}(\hat{p}_4^t)^{s_4^t}...(\hat{p}_K^t)^{s_K^t}(s_1^t+s_2^t+s_3^t+s_4^t+...+s_K^t+K-1)!}{s_1^t!s_2^t!s_3^t!s_4^t!...s_K^t!} \leq \frac{\delta}{(K-1)!}.$$

If we show that

$$\frac{(K-2)!\left(\frac{\hat{p}_1^t+\hat{p}_2^t}{2}\right)^{s_1^t+s_2^t}(\hat{p}_3^t)^{s_3^t}(\hat{p}_4^t)^{s_4^t}..(\hat{p}_K^t)^{s_K^t}(s_1^t+s_2^t+s_3^t+s_4^t+...+s_K^t+K-1)!}{s_1^t!s_2^t!s_3^t!s_4^t!...s_K^t!} \geq$$
$$\frac{\left(\frac{1}{2}\right)^{s_1^t+s_2^t}(s_1^t+s_2^t+1)!}{s_1^t!s_2^t!},$$

then PPR-MD stopping will imply PPR-1v1 stopping. We will show this by using a sequence of inequalities as follows;

$$\frac{(K-2)!\left(\frac{\hat{p}_1^t+\hat{p}_2^t}{2}\right)^{s_1^t+s_2^t}(\hat{p}_3^t)^{s_3^t}(\hat{p}_4^t)^{s_4^t}..(\hat{p}_K^t)^{s_K^t}(s_1^t+s_2^t+s_3^t+s_4^t+...+s_K^t+K-1)!}{s_1^t!s_2^t!s_3^t!s_4^t!...s_K^t!}\geq$$

$$\frac{(K-3)!\left(\frac{\hat{p}_1^t+\hat{p}_2^t}{2}\right)^{s_1^t+s_2^t}(\hat{p}_3^t)^{s_3^t}...(\hat{p}_{K-1}^t)^{s_{K-1}^t}(s_1^t+s_2^t+s_3^t+...+s_{K-1}^t+K-2)!}{s_1^t!s_2^t!s_3^t!...s_{K-1}^t!}\geq...\geq$$

$$\frac{\left(\frac{1}{2}\right)^{s_1^t+s_2^t}(s_1^t+s_2^t+1)!}{s_1^t!s_2^t!}.$$

We will show the first inequality, and the rest follow the same structure except the last (for $K=3$), which was already shown in the subsection above. Note that the last inequality does not follow the same structure due to the lack of $\frac{\hat{p}_1^t+\hat{p}_2^t}{2}$ term in the RHS, since this does not show up in the PPR-1v1 expression. Showing the first inequality is equivalent to showing

$$(K-2)(\hat{p}_K^t)^{s_K^t}\geq\frac{(s_1^t+s_2^t+...+s_{K-1}^t+K-2)!s_K^t!}{(s_1^t+s_2^t+...+s_K^t+K-1)!}$$

$$\Leftrightarrow(K-2)\frac{(s_K^t)^{s_K^t}}{(s_1^t+s_2^t+...+s_K^t)^{s_K^t}}\geq\frac{(s_1^t+s_2^t+...+s_{K-1}^t+K-2)!s_K^t!}{(s_1^t+s_2^t+...+s_K^t+K-1)!}$$

$$\Leftrightarrow(K-2)\frac{y^y}{(x+y)^y}\geq\frac{(x+K-2)!y!}{(x+y+K-1)!},x\geq(K-1)y,\text{ where }x=s_1^t+s_2^t+..+s_{K-1}^t,y=s_K^t.$$

We already know that

$$\frac{x^xy^y}{(x+y)^{x+y}}\geq\frac{(x+1)!y!}{(x+y+2)!},\text{ where }x\geq2y,y>0$$

from our proof for $K=3$. In this equation, mutltiplying the LHS by $\frac{(K-2)(x+y)^x}{x^x}$ which is a factor greater than 1, and multiplying the RHS by $\frac{(x+2)...(x+K-2)}{(x+y+3)...(x+y+K-1)}$, which is a factor lesser than 1, we have the inequality that

$$\frac{x^x}{(x+y)^y}\geq\frac{(x+K-2)!y!}{(x+y+K-1)!},x\geq2y$$

which directly implies what we wanted to show. Hence, even in the case of K parties, PPR-MD stopping implies PPR-1v1 stopping.

C.4 PPR-1v1 and $\mathcal{A}_1$-1v1

Both PPR-1v1 and $\mathcal{A}_1$-1v1 reduce to a single application of PPR-Bernoulli and $\mathcal{A}_1$, respectively, between $v_{\text{first}(t)}$ and $v_{\text{second}(t)}$. Thus it suffices to prove $\mathcal{G}(\text{PPR-Bernoulli},\mathcal{A}_1,X,\delta)$, assuming X is generated by distribution $\mathcal{P}$ over two values.

After any arbitrary t timesteps, without loss of generality, assume $\text{first}(t)=1$ and $\hat{p}_1^t=\frac{s_1^t}{t}$. We prove the following:

$$\begin{aligned}\neg TERM_{PPR}^t&\implies\neg TERM_{\mathcal{A}1}^t,\text{ where}\\TERM_{PPR}^t&\iff\text{Beta}(0.5;s_1^t+1,t-s_1^t+1)\leq\delta,\text{ and}\\TERM_{\mathcal{A}1}^t&\iff\hat{p}_1^t-\beta(t,\delta)>0.5,\qquad\beta(t,\delta)=\sqrt{\frac{2V^t\ln(4t^2/\delta)}{t}}+\frac{7\ln(4t^2/\delta)}{3(t-1)}.\end{aligned}$$

From Lemma 6 (see Appendix D), $\neg TERM_{PPR}^t$ implies,

$$tKL\left(\hat{p}_1^t||0.5\right)<\ln\left(\frac{t+1}{\delta}\right).$$

Applying Pinsker's inequality, we get,

$$t\frac{(2(\hat{p}_1^t - 0.5))^2}{2\ln(2)} < \ln\left(\frac{t+1}{\delta}\right). \tag{2}$$

Case 1: $0.50 \leq \hat{p}_1^t \leq 0.77$
Rearranging equation 2 gives us,

$$\begin{aligned}
&\hat{p}_1^t < 0.5 + \sqrt{\frac{\ln(2)}{2t}\ln\left(\frac{t+1}{\delta}\right)} \\
\implies &\hat{p}_1^t < 0.5 + \sqrt{\frac{2\hat{p}_1^t(1-\hat{p}_1^t)\ln(4t^2/\delta)}{(t-1)}} \\
\implies &\hat{p}_1^t < 0.5 + \left(\sqrt{\frac{2\hat{p}_1^t(1-\hat{p}_1^t)\ln(4t^2/\delta)}{(t-1)}} + \frac{7\ln(4t^2/\delta)}{3(t-1)}\right) \\
\implies &\hat{p}_1^t < 0.5 + \beta(t,\delta) \\
\implies &\neg TERM_{\mathcal{A}1}^t.
\end{aligned}$$

Case 2: $0.77 < \hat{p}_1^t \leq 1$
The lower confidence bound given by the $\mathcal{A}_1$ algorithm is

$$\begin{aligned}
&\hat{p}_1^t - \beta(t,\delta) = \hat{p}_1^t - \left(\sqrt{\frac{2V^t\ln(4t^2/\delta)}{t}} + \frac{7\ln(4t^2/\delta)}{3(t-1)}\right) \\
\implies\ &\hat{p}_1^t - \beta(t,\delta) \leq \hat{p}_1^t - \frac{7}{3}\ln\left(\frac{t+1}{\delta}\right) \\
\implies\ &\hat{p}_1^t - \beta(t,\delta) \leq \hat{p}_1^t - \frac{14(0.5-\hat{p}_1^t)^2}{3\ln(2)} \quad \text{(using equation 2).}
\end{aligned}$$

Since $f(\hat{p}_1^t) = \hat{p}_1^t - \frac{14(0.5-\hat{p}_1^t)^2}{3\ln 2}$ is a decreasing function for $\hat{p}_1^t \geq 0.77$, and $f(0.77) = 0.28$, we have $\neg TERM_{PPR}^t$ implies

$$\begin{aligned}
&\hat{p}_1^t - \beta(t,\delta) \leq 0.5 \\
\implies &\neg TERM_{\mathcal{A}1}^t.
\end{aligned}$$

D PROOF OF ASYMPTOTIC OPTIMALITY OF PPR-1v1

We present a full proof of Theorem 3. Consider a categorical distribution $\mathcal{P}(p, v, K)$ having parameters $p_1 > p_2 \ldots \geq p_K$ for $K \geq 2$. We show that PPR-1v1 asymptotically matches the lower bound given in Theorem 1 when $\mathcal{P}'$ is defined as follows: $p'_1 = p'_2 = \frac{p_1+p_2}{2}$ and $\forall i > 2, p'_i = p_i$. Observe that although $\mathcal{P}'$ itself does not have v_2 as its mode, it yields the supremum defined in $\text{LB}(\mathcal{P}, \delta)$ among distributions that do have v_2 as mode. Hence it is a sufficient choice for our proof, which proceeds in two major steps.

1. In subsection D.1 we fix arbitrary $\epsilon > 0$ as input, and define T^* such that $\lim_{\delta \to 0} \frac{T^*}{\ln(1/\delta)} = \frac{(1+\epsilon)}{KL(\mathcal{P}||\mathcal{P}')}$. Then assuming we have T^* samples in total, we lower bound the number of samples obtained for each of the 1v1 tests involving v_1.

2. In subsection D.2, we show that the number of samples obtained for each test is sufficient for PPR-Bernoulli to terminate and declare v_1 as the winner. The arguments in this subsection are along the lines of those used by Garivier and Kaufmann (2016).

D.1 Lower Bounding the Number of Samples Obtained for each 1v1 Test

Given arbitrary $\epsilon > 0$ and $\alpha > 0$, let us introduce T^*:

$$T^* \stackrel{\text{def}}{=} \frac{1+\epsilon}{KL(\mathcal{P}||\mathcal{P}')} \left(\ln\left(\frac{K-1}{\delta}\right) + \max_{j \in \{2,3\ldots K\}} H_j \right) + \frac{K}{1-K\xi} N(\mathcal{P}, \xi, \epsilon, \alpha) + \frac{K}{1-K\xi} T_0(\gamma).$$

We start by defining the notation used:

- $\xi(\mathcal{P}, \epsilon) : 0 < \xi < \min |p_1 - p_2|/4$ is small enough so that $\left(1 - \frac{\xi}{p_1+p_K}\right)(1+\frac{\epsilon}{2}) > 1$.

- $\gamma(\epsilon)$: $\gamma > 0$ is small enough so that $(1+\gamma)(1+\frac{\epsilon}{2}) < (1+\epsilon)$.

- $T_0(\gamma) : T_0$ is large enough such that $\forall t \geq T_0(\gamma), t^{1+\gamma} > t+1$.

- $H_j = \ln\left(e \left(\frac{(1+\epsilon/2)(1-\xi/(p_1+p_j))}{KL(p_1/(p_1+p_j)||0.5)} \right)^{1+\gamma} \right) + \ln\left(\ln\left(\frac{K-1}{\delta} \left(\frac{(1+\epsilon/2)(1-\xi/(p_1+p_j))}{KL(p_1/(p_1+p_j)||0.5)} \right)^{1+\gamma} \right) \right)$.

- $\xi'(\mathcal{P}, \xi, \epsilon)$: ξ' is such that $|\hat{p}_i^t - p_i| < \xi' \implies D\left(\frac{\hat{p}_1^t}{\hat{p}_1^t + \hat{p}_i^t}||0.5\right) \geq \frac{D(p_1/(p_1+p_i)||0.5)}{(1-\xi/(p_1+p_K))(1+\epsilon/2)}$ for $i \in \{2, 3, \ldots, K\}$

- $N(\mathcal{P}, \xi, \epsilon, \alpha) :$ First, we introduce $\sigma = \max\left\{ t \in N^* : |\hat{p}_i^t - p_i| \geq \min(\frac{\xi}{2}, \xi') \right\}$. By the law of large numbers, for every $\alpha \in (0,1)$, there exists $N(\mathcal{P}, \xi, \epsilon, \alpha)$, such that $\mathbb{P}\left(\sigma \leq N(\mathcal{P}, \xi, \epsilon, \alpha)\right) > 1 - \alpha$.

- E_α: We define the event $E_\alpha = \left\{ \forall t \geq N(\mathcal{P}, \xi, \epsilon, \alpha), \forall i \in \{1, 2 \ldots K\} \quad |\hat{p}_i^t - p_i| < \min(\frac{\xi}{2}, \xi') = \xi'' \right\}$. From the definition of N, $\mathbb{P}(E_\alpha) > 1 - \alpha$.

At time step $t > 0$ let s_i^t denote the number of samples of v_i, and $\hat{p}_i^t = \frac{s_i^t}{t}$ denote the empirical mean for value v_i. For convenience we use $KL(q_1||q_2)$ to denote $KL(\text{Bernoulli}(q_1)||\text{Bernoulli}(q_2))$. We will now show that conditioned on E_α, we have enough samples to separate each class $i \in \{2, 3, \ldots, K\}$ from the mode 1, i.e. $s_1^t + s_i^t$ is sufficiently large.

$$\begin{aligned}
s_1^{T^*} + s_i^{T^*} &= (\hat{p}_1^{T^*} + \hat{p}_i^{T^*})T^* \\
&\geq (p_1 + p_i - \xi)T^* \\
&= \frac{(p_1+p_i)\left(1 - \xi/(p_1+p_i)\right)(1+\epsilon)}{KL(\mathcal{P}||\mathcal{P}')}\left(\ln\left(\frac{K-1}{\delta}\right) + \max_{j\in\{2,3\ldots K\}} H_j\right) \\
&+ K(p_1+p_i)\left(\frac{1-\xi/(p_1+p_i)}{1-K\xi}\right)(N(\mathcal{P},\xi,\epsilon,\alpha) + T_0(\gamma)) \\
&\geq \frac{\left(1-\xi/(p_1+p_i)\right)(1+\epsilon)}{KL(p_1/(p_1+p_i)||0.5)}\left(\ln\left(\frac{K-1}{\delta}\right) + H_i\right) + N(\mathcal{P},\xi,\epsilon,\alpha) + T_0(\gamma) \\
&\geq \frac{\left(1-\xi/(p_1+p_i)\right)(1+\epsilon/2)(1+\gamma)}{KL(p_1/(p_1+p_i)||0.5)}\left(\ln\left(\frac{K-1}{\delta}\right) + H_i\right) + N(\mathcal{P},\xi,\epsilon,\alpha) + T_0(\gamma)
\end{aligned}$$

where the second inequality follows from Lemma 4.

D.2 Termination of the 1v1 Tests

We have shown that conditional on the event E_α,

$$s_1^{T^*} + s_i^{T^*} \geq \frac{\left(1-\xi/(p_1+p_i)\right)(1+\epsilon/2)(1+\gamma)}{KL(p_1/(p_1+p_i)||0.5)}\left(\ln\left(\frac{K-1}{\delta}\right) + H_i\right) + N(\mathcal{P},\xi,\epsilon,\alpha) + T_0(\gamma).$$

Consider $c_1 = \frac{KL(p_1/(p_1+p_i)||0.5)}{(1-\xi/(p_1+p_i))(1+\epsilon/2)}, c_2 = \frac{K-1}{\delta}, \alpha = 1+\gamma$. Since for $p \in (0.5,1], KL(p||0.5) \leq \ln(2) < 1$ we have $c_1^\alpha \leq \ln(2)$. If $\delta \leq 0.5$, we can see that $\frac{c_2}{c_1^\alpha} \geq \frac{2}{\ln(2)} > e$. This allows us to use Lemma 5 and state the following for $\delta \leq 0.5$,

$$\begin{aligned}
(s_1^{T^*} + s_i^{T^*})\frac{KL(p_1/(p_1+p_i)||0.5)}{(1-\xi/(p_1+p_i))(1+\epsilon/2)} &\geq \ln\left(\frac{(K-1)(s_1^{T^*} + s_i^{T^*})^{1+\gamma}}{\delta}\right) \\
&\geq \ln\left(\frac{(K-1)(s_1^{T^*} + s_i^{T^*}+1)}{\delta}\right).
\end{aligned}$$

By the definition of $N(\mathcal{P},\xi,\epsilon,\alpha)$, for $\delta \leq 0.5$ we know that,

$$\begin{aligned}
(s_1^{T_i^*} + s_i^{T^*})KL(\hat{p}_1^{T^*}/(\hat{p}_1^{T^*} + \hat{p}_i^{T^*})||0.5) &\geq \ln\left(\frac{(K-1)(s_1^{T^*} + s_i^{T^*}+1)}{\delta}\right) \\
\implies \text{Beta}(0.5; s_1^{T^*}+1, s_i^{T^*}+1) &\leq \frac{\delta}{K-1}
\end{aligned}$$

where the last implication follows from Lemma 6. At this stage the PPR-Bernoulli Stopping Rule declares that $p_1^{T^*} > p_i^{T^*}$.

The above proof tells us that under the event E_α, after T^* samples, PPR-Bernoulli decides that for $i \in \{2,3,\ldots,K\}$, $p_1 > p_i$, i.e. v_1 is the mode of the distribution. Therefore, if τ denotes the stopping time of PPR-1v1, we have the following bound.

$$\mathbb{P}(\tau \leq T^*) > 1-\alpha \implies \mathbb{P}\left(\lim_{\delta\to 0}\frac{\tau}{\ln(1/\delta)} \leq \frac{1+\epsilon}{KL\left(\mathcal{P}||\mathcal{P}'\right)}\right) > 1-\alpha.$$

To show that the lower bound of Shah et al. (2020) is matched, we upper bound the expected stopping time, using a commonly-known[5] result to bound the tail probability for a binomial random variable.

$$\begin{aligned}
\mathbb{P}(\tau > T^*) &\leq \mathbb{P}(E_\alpha^{\complement}). \\
\mathbb{E}(\tau) &\leq T^* + \sum_{i=1}^{K} \sum_{t=N(\mathcal{P},\xi,\epsilon,\alpha)}^{\infty} \mathcal{P}(|\hat{p}_i^t - p_i| > \xi'') \\
&\leq T^* + \sum_{i=1}^{K} \sum_{t=N(\mathcal{P},\xi,\epsilon,\alpha)}^{\infty} \exp\left(-2tKL(p_i - \xi''||\xi)\right) \\
&\leq T^* + \sum_{i=1}^{K} \sum_{t=0}^{\infty} \exp\left(-2tKL(p_i - \xi''||\xi)\right) \\
&\leq T^* + \sum_{i=1}^{K} \frac{1}{1 - \exp(2KL(p_i - \xi''||p_i)}.
\end{aligned}$$

Hence, we have

$$\lim_{\delta \to 0} \frac{\mathbb{E}_{\mathcal{P}}(\tau)}{\ln(1/\delta)} \leq \frac{1+\epsilon}{KL\left(\mathcal{P}||\mathcal{P}'\right)},$$

which completes our proof.

We now furnish proofs of the lemmas that were used in the proofs above.

Lemma 4. *For $x \in [0, p_2]$,*

$$(p_1 + x)KL\left(\frac{p_1}{p_1 + x}||0.5\right) \geq KL(\mathcal{P}||\mathcal{P}').$$

Proof. By definition,

$$\begin{aligned}
(p_1 + x)KL\left(\frac{p_1}{p_1 + x}||0.5\right) &= p_1 \ln\left(\frac{p_1}{(p_1 + x)/2}\right) + x \ln\left(\frac{x}{(p_1 + x)/2}\right) \\
&\geq p_1 \ln\left(\frac{p_1}{(p_1 + p_2)/2}\right) + p_2 \ln\left(\frac{p_2}{(p_1 + p_2)/2}\right) \\
&= KL(\mathcal{P}||\mathcal{P}').
\end{aligned}$$

The second line follows from from the fact that $p_2 \geq x$ and that $F(x) = p_1 \ln\left(\frac{p_1}{(p_1+x)/2}\right) + p \ln\left(\frac{x}{(p_1+x)/2}\right), x \in [0, p_2]$ is a decreasing function ($F'(x) = \ln\left(\frac{2x}{p_1+x}\right) < 0$). □

Lemma 5. *For $\alpha \in [1, e/2]$ for any two constants c_1 and c_2 such that $\frac{c_2}{c_1^\alpha} \geq e$, we have*

$$c_1 x \geq \ln(c_2 x^\alpha), \forall x \geq T_0(c_1, c_2, \alpha) \text{ where } T_0(c_1, c_2, \alpha) = \frac{\alpha}{c_1}\left[\ln\left(\frac{c_2 e}{c_1^\alpha}\right) + \ln\ln\left(\frac{c_2}{c_1^\alpha}\right)\right].$$

Proof. Consider $F(x) = c_1 x - \ln(c_2 x^\alpha)$. We have $F'(x) = c_1 - \frac{\alpha}{x}$ which means that $F(x)$ is increasing for $x > \frac{\alpha}{c_1}$. Using Lemma 18 by Garivier and Kaufmann (2016) we know that for $x = T_0(c_1, c_2, \alpha), c_1 x \geq c_2 x^\alpha$, and $\frac{c_2}{c_1^\alpha} \geq e \implies T_0(c_1, c_2, \alpha) \geq \frac{\alpha}{c_1}[\ln(e^2) + \ln\ln(e)] = 2\frac{\alpha}{c_1}$. The fact that $F(x)$ is increasing at this value of x completes our proof. □

Lemma 6. *For $i \in \{2, 3, \ldots, K\}$,*

$$(s_1^t + s_i^t)KL(\hat{p}_1^t/(\hat{p}_1^t + \hat{p}_i^t)||0.5) \geq \ln\left(\frac{s_1^t + s_i^t + 1}{\delta}\right) \implies Beta(0.5; s_1^t + 1, s_i^t + 1) \leq \delta.$$

[5] See Theorem 2.1 by Mulzer (2018).

Proof.

$$
\begin{aligned}
&(s_1^t + s_i^t) KL(\hat{p}_1^t/(\hat{p}_1^t + \hat{p}_i^t)||0.5) \geq \ln\left(\frac{s_1^t + s_i^t + 1}{\delta}\right) \\
\implies &(s_1^t + s_i^t)\frac{\hat{p}_1^t}{\hat{p}_1^t + \hat{p}_i^t}\ln\left(\frac{\hat{p}_1^t}{\hat{p}_1^t + \hat{p}_i^t}\right) + (s_1^t + s_i^t)\frac{\hat{p}_i^t}{\hat{p}_1^t + \hat{p}_i^t}\ln\left(\frac{\hat{p}_i^t}{\hat{p}_1^t + \hat{p}_i^t}\right) - (s_1^t + s_i^t)\ln(0.5) \geq \ln\left(\frac{s_1^t + s_i^t + 1}{\delta}\right) \\
\implies &(s_1^t + s_i^t) H_b\left(\frac{\hat{p}_i^t}{\hat{p}_1^t + \hat{p}_i^t}\right) + (s_1^t + s_i^t)\log_2(0.5) \leq \log_2\left(\frac{\delta}{s_1^t + s_i^t + 1}\right).
\end{aligned}
$$

The last step follows from the definition of binary entropy function H_b for $p \in [0, 1]$:

$$
H_b(p) = \begin{cases} 0 & p \in \{0, 1\}, \\ -p\log_2 p - (1-p)\log_2(1-p) & \text{otherwise.} \end{cases}
$$

Writing the condition in this form allows us to use the following well known[6] inequality, with $\alpha = \frac{\hat{p}_i^t}{\hat{p}_1^t + \hat{p}_i^t}$ ($\alpha < \frac{1}{2}$ as $s_1^t > s_i^t$) and $t = s_1^t + s_i^t$.

$$
\binom{t}{\alpha t} \leq 2^{tH_b(\alpha)} \text{ where } t \in \mathbb{N} \text{ and } \alpha \in \left[0, \frac{1}{2}\right].
$$

So we have

$$
\begin{aligned}
&(s_1^t + s_i^t) H_b\left(\frac{\hat{p}_1^t}{\hat{p}_1^t + \hat{p}_i^t}\right) + (s_1^t + s_i^t)\log_2(0.5) \leq \log_2\left(\frac{\delta}{s_1^t + s_i^t + 1}\right) \\
\implies &\log_2\left(\binom{s_1^t + s_i^t}{s_1^t}\right) + (s_1^t + s_i^t)\log_2(0.5) + \log_2(s_1^t + s_i^t + 1) \leq \log_2(\delta) \\
\implies &\log_2\left(\frac{(s_1^t + s_i^t + 1)!(0.5)^{s_1^t}(0.5)^{s_i^t}}{(s_1^t)!(s_i^t)!}\right) \leq \log_2(\delta) \\
\implies &\text{Beta}(0.5; s_1^t + 1, s_i^t + 1) \leq \delta.
\end{aligned}
$$

□

[6] See, for example, Galvin (2014).

E NON-ASYMPTOTIC UPPER BOUND FOR PPR-1v1

In Appendix D we showed that the PPR-1v1 stopping rule is asymptotically optimal, in the regime that $\delta \to 0$. In this section, we show a non-asymptotic upper bound on its sample complexity: in other words, a result that holds for all $\delta \in (0, 1)$. To the best of our knowledge, the tightest such upper bound given yet for the PAC mode estimation problem is the recent result of Shah et al. (2020), which we reproduce below.

Theorem 7 ($\mathcal{A}_1$ upper bound Shah et al. (2020))**.** *Fix $\delta \in (0,1)$, $K \geq 2$, and problem instance $\mathcal{P} = (p, v, K)$. When $\mathcal{A}_1$ is run on $\mathcal{P}$, with probability $1 - \delta$, the number of samples it observes is at most*

$$\frac{592}{3} \frac{p_1}{(p_1 - p_2)^2} \ln \left(\frac{592}{3} \sqrt{\frac{K}{\delta}} \frac{p_1}{(p_1 - p_2)^2} \right).$$

It is easy to show that the leading $\frac{p_1}{(p_1 - p_2)^2}$ factor is within a constant factor of $LB(\mathcal{P}, \delta)$ (Shah et al., 2020). In this appendix, we derive a similar upper bound for PPR-1v1, albeit one that is tighter by a small constant factor. Our first step is to show an upper bound for the special case of $K = 2$ (wherein PPR-1v1 reduces to PPR-Bernoulli) in Appendix E.1. In turn, this result is used to generalise to $K \geq 2$ in Appendix E.2. The final upper bound, is given in Theorem 9.

E.1 An Upper Bound for K = 2

Lemma 8 (PPR-Bernoulli upper bound)**.** *Fix $\delta \in (0,1)$ and problem instance $\mathcal{P} = (p, v, 2)$. When PPR-Bernoulli is run on $\mathcal{P}$, with probability $1 - \delta$, the number of samples it observes is at most*

$$\frac{20.775 p_1}{(p_1 - \frac{1}{2})^2} \ln \left(\frac{2.49}{(p_1 - \frac{1}{2})^2 \delta} \right).$$

Proof. For a problem instance $\mathcal{P} = (p, v, 2)$, our parameters are p_1 and $p_2 = 1 - p_1$, with $p_1 > 0.5$. It suffices to maintain a confidence sequence on p_1; termination is achieved when this confidence sequence no longer contains $\frac{1}{2}$. To upper-bound the number of samples needed for termination, we proceed in three steps.

1. At time t, there are 2^t possible 0-1 sequences that the Bernoulli variable can produce in t steps. Let X^t be a random variable denoting this t-length 0-1 sequence. Let s_1^t denote the number of times v_1 occurs in X^t. In Subsection E.1.1, we find the range of s_1^t for which $\mathbb{1}\left\{R^t(\frac{1}{2}) \geq \frac{1}{\delta}\right\}$.

2. Next, in Appendix E.1.2, we use the range of k derived in Appendix E.1.1 to derive a sufficient condition for t to be the sample complexity.

3. In Appendix E.1.3, we use the sufficient condition derived in Subsection E.1.2 to obtain a closed-form sample complexity upper bound. We separately take up two cases, $p_1 \leq 0.6$, and $p_1 > 0.6$, so as to tighten the constants in the upper bound.

E.1.1 Finding a Range of k for which $\mathbb{1}\left\{R^t(\frac{1}{2}) \geq \frac{1}{\delta}\right\}$

The stopping rule does not terminate at t so long as $R^t\left(\frac{1}{2}\right) < \frac{1}{\delta}$.

$$\mathbb{P}\left(R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right) = \sum_{X^t} \mathbb{1}\left\{R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right\} P_{p_1}(X^t).$$

The expression inside the indicator random variable can be simplified to obtain the following:

$$\mathbb{1}\left\{R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right\} = \mathbb{1}\left\{\frac{1}{\pi_t(1/2)} \geq \frac{1}{\delta}\right\} = \mathbb{1}\left\{\frac{\int_{\eta=0}^{1} P_\eta(X^t) d\eta}{P_{1/2}(X^t)} \geq \frac{1}{\delta}\right\}.$$

The sum inside the indicator function is independent of the position of 0-1 outcomes within the sequence X^t and only depends on the number of zeros and ones in the instance. Suppose we consider the instances which contain

s_1^t ones; then, we have

$$\mathbb{1}\left\{R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right\} = \mathbb{1}\left\{\delta \geq \frac{(t+1)!}{s_1^t!(t-s_1^t)!}\left(\frac{1}{2}\right)^{s_1^t}\left(\frac{1}{2}\right)^{t-s_1^t}\right\}.$$

From the above equations, we derive the exact expression for $\mathbb{P}(R^t(\frac{1}{2}) \geq \frac{1}{\delta})$:

$$\mathbb{P}\left(R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right) = \sum_{s_1^t=0}^{t}\binom{t}{s_1^t}p_1^{s_1^t}(1-p_1)^{t-s_1^t}\mathbb{1}\left\{\delta \geq \frac{(t+1)!}{s_1^t!(t-s_1^t)!}2^{-t}\right\}.$$

We now break the expression into two parts.

$$\begin{aligned}\mathbb{P}\left(R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right) &= \sum_{s_1^t=0}^{t}\binom{t}{s_1^t}p_1^{s_1^t}(1-p_1)^{t-s_1^t}\mathbb{1}\left\{\delta \geq \frac{(t+1)!}{s_1^t!(t-s_1^t)!}2^{-t}\right\}\\ &= \sum_{s_1^t=0}^{\lfloor\frac{t}{2}\rfloor}\binom{t}{s_1^t}p_1^{s_1^t}(1-p_1)^{t-s_1^t}\mathbb{1}\left\{\frac{\delta}{t+1} \geq \binom{t}{s_1^t}2^{-t}\right\} +\\ &\quad \sum_{s_1^t=\lfloor\frac{t}{2}\rfloor+1}^{t}\binom{t}{s_1^t}p_1^{s_1^t}(1-p_1)^{t-s_1^t}\mathbb{1}\left\{\frac{\delta}{t+1} \geq \binom{t}{t-s_1^t}2^{-t}\right\}.\end{aligned}$$

We have split the summation in this manner as it allows to use the following commonly-known[7] inequality for the binomial coefficients inside the indicator function.

$$\binom{t}{\alpha t} \leq 2^{tH_b(\alpha)} \text{ where } t \in \mathbb{N} \text{ and } \alpha \in \left[0, \frac{1}{2}\right].$$

The definition of binary entropy function H_b for $p \in [0,1]$ is:

$$H_b(p) = \begin{cases} 0 & p \in \{0,1\},\\ -p\log_2 p - (1-p)\log_2(1-p) & \text{otherwise.}\end{cases}$$

Invoking this inequality in the probability expression, we arrive at the lower bound:

$$\begin{aligned}\mathbb{P}\left(R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right) \geq &\sum_{s_1^t=0}^{\lfloor\frac{t}{2}\rfloor}\binom{t}{s_1^t}p_1^{s_1^t}(1-p_1)^{t-s_1^t}\mathbb{1}\left\{\frac{1}{t}\log_2\left(\frac{\delta}{t+1}\right)+1 \geq H_b\left(\frac{s_1^t}{t}\right)\right\} +\\ &\sum_{s_1^t=\lfloor\frac{t}{2}\rfloor+1}^{t}\binom{t}{s_1^t}p_1^{s_1^t}(1-p_1)^{t-s_1^t}\mathbb{1}\left\{\frac{1}{t}\log_2\left(\frac{\delta}{t+1}\right)+1 \geq H_b\left(\frac{t-s_1^t}{t}\right)\right\}.\end{aligned}$$

Using the fact that

$$H_b(p) \leq 2\sqrt{p(1-p)} \text{ if } p \leq \frac{1}{2},$$

the summation is further lower-bounded:

$$\mathbb{P}\left(R^t\left(\frac{1}{2}\right) \geq \frac{1}{\delta}\right) \geq \sum_{s_1^t=0}^{t}\binom{t}{s_1^t}p_1^{s_1^t}(1-p_1)^{t-s_1^t}\mathbb{1}\left\{\frac{1}{2}\left[\log_2\left(\frac{\delta}{t+1}\right)+t\right] \geq \sqrt{s_1^t(t-s_1^t)}\right\}.$$

We will later see that for the t we find as the bound, $\log_2\left(\frac{\delta}{t+1}\right)+t \geq 0$. Hence, we can take the square of both sides. By using that fact and solving the quadratic, the range of s_1^t which satisfies the condition inside the indicator function is:

$$s_1^t \in \left[0, \frac{t-\sqrt{t^2-4\beta^2}}{2}\right]\bigcup\left[\frac{t+\sqrt{t^2-4\beta^2}}{2}, t\right] \text{ where } \beta = \frac{1}{2}\left[\log_2\left(\frac{\delta}{t+1}\right)+t\right].$$

[7] See, for example, Galvin (2014).

Slightly loosening this bound, we get

$$s_1^t \in \left[0, \frac{t - \sqrt{2t\log_2\left(\frac{t+1}{\delta}\right)}}{2}\right] \bigcup \left[\frac{t + \sqrt{2t\log_2\left(\frac{t+1}{\delta}\right)}}{2}, t\right].$$

In short, if s_1^t lies in the range above, then it must satisfy $\mathbb{1}\left\{R^t(\frac{1}{2}) \geq \frac{1}{\delta}\right\}$. Call this range R.

E.1.2 Finding a Sufficient Condition for t to be the Sample Complexity

We will lower bound the probability of $\sum_{s_1^t \in R} p_1^{s_1^t}(1-p_1)^{t-s_1^t}$ by $1-\delta$, and thus find the values of t satisfying this equation. To that end, we use another commonly-known [8] result to bound the tail probability for a binomial random variable: for $\epsilon > 0$,

$$\mathbb{P}(H(t) \leq (p_1 - \epsilon)t) \leq e^{-D_{KL}(p_1-\epsilon||p_1)t}.$$

Using this in our working, considering $H(t)$ as a binomial random variable, we get

$$\begin{aligned} \sum_{s_1^t=(p_1-\epsilon)t}^{t} \binom{t}{s_1^t} p_1^{s_1^t}(1-p_1)^{t-s_1^t} &= \mathbb{P}\left((p_1-\epsilon)t \leq H(t)\right) \\ &\geq 1 - e^{-D_{KL}(p_1-\epsilon||p_1)t} \\ \implies \mathbb{P}\left(R^t\left(\frac{1}{2}\right) \leq \frac{1}{\delta}\right) &\leq e^{-D_{KL}(p_1-\epsilon||p_1)t}. \end{aligned}$$

We want $\mathbb{P}\left(R^t\left(\frac{1}{2}\right) > \frac{1}{\delta}\right) \geq 1-\delta$. A sufficient condition for the above to hold is

$$\begin{aligned} e^{-D_{KL}(p_1-\epsilon||p_1)t} &\leq \delta \\ \implies t \times D_{KL}(p_1-\epsilon||p_1) &\geq \ln\left(\frac{1}{\delta}\right) \end{aligned}$$

And ϵ is given by,

$$\epsilon = p_1 - \frac{1}{2}\left(1 + \sqrt{\frac{2\log_2\left(\frac{t+1}{\delta}\right)}{t}}\right), \epsilon \geq 0.$$

We will note that $\epsilon \geq 0$ holds later. For now, we try to find stronger sufficient conditions on t. To that end, we use that

$$\begin{aligned} D_{KL}(x||y) &\geq \frac{(x-y)^2}{2y}, x \leq y \\ \implies D_{KL}\left(p_1-\epsilon||p_1\right) &\geq \frac{\epsilon^2}{2p_1}. \end{aligned}$$

So, a sufficient condition for

$$t \times D_{KL}(p_1-\epsilon||p_1) \geq \ln\left(\frac{1}{\delta}\right)$$

is given as

$$\begin{aligned} \frac{t\epsilon^2}{2p_1} &\geq \ln\left(\frac{1}{\delta}\right) \\ \implies t &\geq \frac{2p_1}{\epsilon^2}\ln\left(\frac{1}{\delta}\right). \end{aligned}$$

[8] See Theorem 2.1 in Mulzer (2018).

To find a sufficient condition for t to satisfy the above, we can find a lower bound on ϵ^2 and use it in the expression above. We have

$$\begin{aligned}\epsilon^2 &= \left(\left(p_1 - \frac{1}{2}\right) - \sqrt{\frac{\log_2\left(\frac{t+1}{\delta}\right)}{2t}}\right)^2 \\ &= \left(p_1 - \frac{1}{2}\right)^2 + \frac{\log_2\left(\frac{t+1}{\delta}\right)}{2t} - 2\left(p_1 - \frac{1}{2}\right)\sqrt{\frac{\log_2\left(\frac{t+1}{\delta}\right)}{2t}} \\ \implies \epsilon^2 &\geq \left(p_1 - \frac{1}{2}\right)^2 - 2\left(p_1 - \frac{1}{2}\right)\sqrt{\frac{\log_2\left(\frac{t+1}{\delta}\right)}{2t}}.\end{aligned}$$

E.1.3 Constant Factor Tightening

To obtain our final bound, we consider two cases: $p_1 \leq 0.6, p_1 > 0.6$.

Case 1: $p_1 \leq 0.6$ We will find a sufficient condition on t for the fact that

$$\frac{t}{C_1}\left(p_1 - \frac{1}{2}\right)^2 \geq \log_2\left(\frac{t+1}{\delta}\right) \tag{3}$$

where $C_1 = 2.4488$.
Earlier in the proof, we have used the fact that $\log_2\left(\frac{\delta}{t+1}\right) + t \geq 0$. We first prove this fact. For $p_1 \in (\frac{1}{2}, 1]$, $\frac{(p_1-\frac{1}{2})^2}{C_1} \leq 1$. So, from (3), we can easily see that $\log_2\left(\frac{\delta}{t+1}\right) + t \geq 0$.
We find a sufficient t such that

$$\begin{aligned}\frac{t}{C_1}\left(p_1 - \frac{1}{2}\right)^2 &\geq \log_2\left(\frac{t+1}{\delta}\right) \\ \implies t &\geq \frac{C_1}{(p_1 - \frac{1}{2})^2}\log_2\left(\frac{t+1}{\delta}\right).\end{aligned}$$

We let t be of the form

$$t = \frac{C_1 C_2}{(p_1 - \frac{1}{2})^2}\log_2\left(\frac{C_1}{(p_1 - \frac{1}{2})^2\delta}\right),$$

where C_2 is a constant. Note that, for this choice of t, (We will show later that $t \geq 4000$ is also true for the first line below to hold),

$$\begin{gathered} t + 1 \leq 1.00025t. \\ \frac{C_1}{(p_1 - \frac{1}{2})^2}\log_2\left(\frac{t+1}{\delta}\right) \leq \frac{C_1}{(p_1 - \frac{1}{2})^2}\log_2\left(\frac{1.00025t}{\delta}\right). \\ \frac{C_1}{(p_1 - \frac{1}{2})^2}\log_2\left(\frac{t+1}{\delta}\right) \leq \frac{C_1}{(p_1 - \frac{1}{2})^2}\log_2(1.00025C_2) + 0.3839\frac{C_1}{(p_1 - \frac{1}{2})^2}\log_2\left(\frac{C_1}{(\theta - \frac{1}{2})^2\delta}\right) + \\ \frac{C_1}{(p_1 - \frac{1}{2})^2}\log_2\left(\frac{C_1}{(p_1 - \frac{1}{2})^2\delta}\right),\end{gathered}$$

where the last step is true because

$$\log_2\left(\log_2\left(\frac{C_1}{(p_1 - \frac{1}{2})^2\delta}\right)\right) \leq 0.3839\log_2\left(\frac{C_1}{(p_1 - \frac{1}{2})^2\delta}\right)$$

using that, for $x \geq 200$,

$$\log_2\left(\log_2(x)\right) \leq 0.3839\log_2(x).$$

Thus, we get

$$\begin{aligned}\frac{C_1}{(p_1-\frac{1}{2})^2}\log_2\left(\frac{t+1}{\delta}\right) &\leq \frac{C_1}{(p_1-\frac{1}{2})^2}\log_2\left(\frac{C_1}{(p_1-\frac{1}{2})^2\delta}\right)(1.3839+\log_2(1.00025C_2))\\ &\leq \frac{C_1}{(p_1-\frac{1}{2})^2}\log_2\left(\frac{C_1}{(p_1-\frac{1}{2})^2\delta}\right)C_2\end{aligned}$$

where the last step is true when $C_2 = 2.9402$. Thus,

$$t = \frac{2C_1C_2p_1}{(p_1-\frac{1}{2})^2}\log_2\left(\frac{C_1}{(p_1-\frac{1}{2})^2\delta}\right) \implies t = \frac{C_3p_1}{(p_1-\frac{1}{2})^2}\ln\left(\frac{C_1}{(p_1-\frac{1}{2})^2\delta}\right),$$

where $C_3 = 20.775 \geq 2C_1C_2\log_2(e)$ satisfies the above inequality.

So now, if the condition on t is satisfied, we have

$$\begin{aligned}\epsilon^2 &\geq \left(p_1-\frac{1}{2}\right)^2 - 2\left(p_1-\frac{1}{2}\right)\sqrt{\frac{\log_2\left(\frac{t+1}{\delta}\right)}{2t}} \geq \left(p_1-\frac{1}{2}\right)^2 - 2\left(p_1-\frac{1}{2}\right)\frac{(p_1-\frac{1}{2})}{2.21305}\\ &\implies \epsilon^2 \geq C_4\left(p_1-\frac{1}{2}\right)^2,\end{aligned}$$

where $C_4 = 0.09627$.
Previously, we had remarked that it was sufficient to have

$$t \geq \frac{2p_1}{\epsilon^2}\ln\left(\frac{1}{\delta}\right).$$

We can loosen this by using the lower bound on ϵ to: it is sufficient to satisfy

$$t \geq \frac{C_5p_1}{\left(p_1-\frac{1}{2}\right)^2}\ln\left(\frac{1}{\delta}\right),$$

where $C_5 = 20.775$.
To make both the lower bound on ϵ and the above equation valid, we note that the following choice of t works:

$$t = \frac{C_5p_1}{\left(p_1-\frac{1}{2}\right)^2}\max\left(\left(\ln\left(\frac{C_1}{(p_1-\frac{1}{2})^2\delta}\right), \ln\left(\frac{1}{\delta}\right)\right)\right)$$

We note that this leaves us with the following choice of t:

$$t = \frac{20.775p_1}{\left(p_1-\frac{1}{2}\right)^2}\left(\ln\left(\frac{2.45}{(p_1-\frac{1}{2})^2\delta}\right)\right)$$

Note that $t \geq 4000$, since $p_1 \leq 0.6$.

Case 2: $p_1 > 0.6$ We will find a sufficient condition on t for the fact that

$$\frac{t}{C_1}\left(p_1-\frac{1}{2}\right)^2 \geq \log_2\left(\frac{t+1}{\delta}\right)$$

where $C_1 = 2.4877$.
Using the same argument as in case 1, here also we can prove that $\log_2\left(\frac{\delta}{t+1}\right) + t \geq 0$.
We find a sufficient t such that

$$\begin{aligned}\frac{t}{C_1}\left(p_1-\frac{1}{2}\right)^2 &\geq \log_2\left(\frac{t+1}{\delta}\right)\\ \implies t &\geq \frac{C_1}{(p_1-\frac{1}{2})^2}\log_2\left(\frac{t+1}{\delta}\right).\end{aligned}$$

We let t be of the form

$$t = \frac{C_1 C_2}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right),$$

where C_2 is a constant.
Note that, for this choice of t, (We will show later that $t \geq 100$ is also true for the first line to hold)

$$\begin{gathered} t + 1 \leq 1.01t. \\ \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{t+1}{\delta} \right) \leq \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{1.01t}{\delta} \right). \\ \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{t+1}{\delta} \right) \leq \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2(1.01C_2) + 0.5228 \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{C_1}{(\theta - \frac{1}{2})^2 \delta} \right) + \\ \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right), \end{gathered}$$

where the last step is true because

$$\log_2 \left(\log_2 \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right) \right) \leq 0.5228 \log_2 \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right)$$

using that, for $x \geq 9.64$,

$$\log_2 \left(\log_2 \left((x \right) \right) \leq 0.5228 \log_2 (x)$$

Thus, we get

$$\begin{aligned} \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{t+1}{\delta} \right) &\leq \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right) (1.5228 + \log_2 (1.01C_2)) \\ &\leq \frac{C_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right) C_2, \end{aligned}$$

where the last step is true when $C_2 = 3.228$. Thus, using that $p_1 > 0.6$ by writing that $\frac{p_1}{0.6} \geq 1$, we have that

$$t = \frac{1.67 C_1 C_2 p_1}{(p_1 - \frac{1}{2})^2} \log_2 \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right) \implies t = \frac{C_3 p_1}{(p_1 - \frac{1}{2})^2} \ln \left(\frac{C_1}{(p_1 - \frac{1}{2})^2 \delta} \right),$$

where $C_3 = 19.35 \geq 1.67 C_1 C_2 \log_2(e)$ satisfies the above inequality.

So now, if the condition on t is satisfied, we have

$$\begin{gathered} \epsilon^2 \geq \left(p_1 - \frac{1}{2} \right)^2 - 2 \left(p_1 - \frac{1}{2} \right) \sqrt{\frac{\log_2 \left(\frac{t+1}{\delta} \right)}{2t}} \geq \left(p_1 - \frac{1}{2} \right)^2 - 2 \left(p_1 - \frac{1}{2} \right) \frac{\left(p_1 - \frac{1}{2} \right)}{2.2305} \\ \implies \epsilon^2 \geq C_4 \left(p_1 - \frac{1}{2} \right)^2, \end{gathered}$$

where $C_4 = 0.1033$.
Previously, we had remarked that it was sufficient to have

$$t \geq \frac{2p_1}{\epsilon^2} \ln \left(\frac{1}{\delta} \right).$$

We can loosen this by using the lower bound on ϵ to: it is sufficient to satisfy

$$t \geq \frac{C_5 p_1}{\left(p_1 - \frac{1}{2} \right)^2} \ln \left(\frac{1}{\delta} \right),$$

where $C_5 = 19.36$.
To make both the lower bound on ϵ and the above equation valid, we note that the following choice of t works:

$$t = \frac{C_5 p_1}{\left(p_1 - \frac{1}{2}\right)^2} \max\left(\left(\ln\left(\frac{C_1}{(p_1 - \frac{1}{2})^2\delta}\right), \ln\left(\frac{1}{\delta}\right)\right)\right).$$

We note that this leaves us with the following choice of t:

$$t = \frac{19.36 p_1}{\left(p_1 - \frac{1}{2}\right)^2} \left(\ln\left(\frac{2.49}{(p_1 - \frac{1}{2})^2\delta}\right)\right)$$

Note that $t \geq 106$.

Combining both the above bounds, we have that

$$t = \frac{19.36 p_1}{\left(p_1 - \frac{1}{2}\right)^2} \left(\ln\left(\frac{2.49}{(p_1 - \frac{1}{2})^2\delta}\right)\right)$$

is sufficient.

Looking over both cases, we note

$$t = \frac{20.775 p_1}{\left(p_1 - \frac{1}{2}\right)^2} \left(\ln\left(\frac{2.49}{(p_1 - \frac{1}{2})^2\delta}\right)\right)$$

is sufficient. We denote $C_6 = 20.775, C_7 = 2.49$ to get

$$t = \frac{C_6 p_1}{\left(p_1 - \frac{1}{2}\right)^2} \left(\ln\left(\frac{C_7}{(p_1 - \frac{1}{2})^2\delta}\right)\right)$$

as an upper bound on the sample complexity of PPR-Bernoulli with probability atleast $1 - \delta$. □

E.2 An Upper Bound for General K

Theorem 9 (PPR-1v1 upper bound)**.** *Fix $\delta \in (0, 1)$, $K \geq 2$, and problem instance $\mathcal{P} = (p, v, K)$. When PPR-1v1 is run on $\mathcal{P}$, with probability $1 - \delta$, the number of samples it observes is at most*

$$t^\star = \frac{194.07 p_1}{(p_1 - p_2)^2} \ln\left(\sqrt{\frac{79.68(K-1)}{\delta}} \frac{p_1}{(p_1 - p_2)}\right).$$

Proof. Fix arbitrary $t > t^\star$ and $j \geq 2$. Our strategy is to show that with sufficiently high probability, v_1 and v_j will have separated before t pulls. First, let s^t_{ij} denote the sum of the number of occurrences of v_1 and v_j in the first t samples: that is, $s^t_{ij} = s^t_1 + s^t_j$. Clearly s^t_{ij} is a binomial random variable with parameters t and $(p_1 + p_j)$, We argue that s^t_{ij} cannot fall too far below its mean. Concretely, take $\delta' = \frac{\delta}{2(K-1)}$ and $l = \sqrt{\frac{2\ln(\frac{1}{\delta'})}{(p_1+p_2)t}}$. A Chernoff bound yields

$$\mathbb{P}\{s^t_{ij} \leq (1 - l)(p_1 + p_j)t\} \leq \exp\left(-\frac{l^2(p_1 + p_j)t}{2}\right) \leq \delta'$$

for our choice of l. Thus, with probability at least $1-\delta'$, v_1 and v_j together have more than $(1-l)(p_1+p_j)t$ samples. Now, the test to separate v_1 and v_j is PPR-Bernoulli on a Bernoulli variable with parameter $q_1 = \frac{p_1}{p_1+p_j} > \frac{1}{2}$. Although PPR-1v1 runs this test with mistake probability $\frac{\delta}{K-1}$, imagine running it with mistake probability $\delta' = \frac{\delta}{2(K-1)}$. The latter test would necessarily incur equal or more samples on *every* run. Yet, from Lemma 8, we know that with probability at least $1 - \delta'$, the latter will terminate after at most $u = \frac{20.775 q_1}{(q_1 - \frac{1}{2})^2} \ln\left(\frac{2.49}{(q_1 - \frac{1}{2})^2\delta'}\right)$ samples. It can be verified that for $t > t^\star$, $u < (1 - l)(p_1 + p_j)t$ (calculation shown in Appendix E.2.1). In other words, the probability that v_1 and v_j have *not* separated before t pulls is at most $2\delta'$.

Since the argument above holds for arbitrary $j \geq 2$, a union bound establishes that with probability at least $1 - \delta$, v_1 must have separated from all other values—implying termination—before t pulls. □

E.2.1 Calculation in Proof of Theorem 9

We have to show $u < (1-l)(p_1+p_j)t$ for $t > t^\star$, where

$$\begin{aligned} u &= \frac{20.775 q_1}{(q_1-\frac{1}{2})^2} \ln\left(\frac{2.49}{(q_1-\frac{1}{2})^2\delta'}\right), \\ q_1 &= \frac{p_1}{p_1+p_j}, \\ l &= \sqrt{\frac{2\ln(1/\delta')}{(p_1+p_j)t}}, \text{ and} \\ t^\star &= \frac{194.07 p_1}{(p_1-p_2)^2} \ln\left(\sqrt{\frac{79.68(k-1)}{\delta}} \frac{p_1}{(p_1-p_2)}\right). \end{aligned}$$

First, observe that

$$u = \frac{83.1 p_1(p_1+p_j)}{(p_1-p_j)^2} \ln\left(\frac{9.96(p_1+p_j)^2}{(p_1-p_j)^2\delta'}\right) \leq \frac{83.1 p_1(p_1+p_j)}{(p_1-p_2)^2} \ln\left(\frac{39.84 p_1^2}{(p_1-p_2)^2\delta'}\right).$$

Thus, we have

$$\frac{u}{(p_1+p_j)t^*} < 0.85640 \frac{\ln\left(\frac{39.84 p_1^2}{(p_1-p_2)^2\delta'}\right)}{\ln\left(\frac{79.68(k-1)}{\delta}\frac{p_1^2}{(p_1-p_2)^2}\right)} = 0.85640.$$

It suffices to show that $l \leq 0.1436$, which is established by the steps below.

$$\begin{aligned} l^2 &\leq \frac{2\ln(1/\delta')(p_1-p_2)^2}{(p_1+p_j)97.035 p_1 \ln\left(\frac{79.68(k-1)p_1^2}{\delta(p_1-p_2)^2}\right)} \leq \frac{2(p_1-p_2)^2}{(p_1+p_j)97.035 p_1} \leq \frac{2}{97.035} \\ \implies l &\leq \frac{1}{\sqrt{97.035/2}} < 0.1436. \end{aligned}$$

F DCB ALGORITHM

We describe the DCB ("Difference in Confidence Bounds") algorithm for sampling constituencies in the indirect election winner-forecasting problem from Section 5.1.

Suppose we have C constituencies. For ease of explanation, we assume that all the constituencies have the same set of $K \geq 2$ parties (in reality we maintain a separate list for each constituency). Although constituencies have finite populations, these are usually large enough to ignore the benefit of without-replacement samples; we simply view each response as a sample from a discrete distribution.

After $t \geq 1$ samples have been obtained from the population, let $\mathrm{LCB}^t(c, i)$ and $\mathrm{UCB}^t(c, i)$ denote (1vr) lower and upper confidence bounds, respectively, on the the (true) fraction of votes to be cast for party $i \in \{1, 2, \dots, K\}$ in constituency $c \in \{1, 2, \dots, C\}$. If applying a 1v1 procedure for mode estimation, we have separate confidence bounds $\mathrm{LCB}^t(c, i, j)$ and $\mathrm{UCB}^t(c, i, j)$ for each pair of parties $i, j \in \{1, 2, \dots, K\}$, $i \neq j$. The permitted mistake probability δ is divided equally among the constituencies.

The key idea in the DCB algorithm is to keep lower and upper confidence bounds on the *wins of each party* (across constituencies), and to use this information to guide sampling. For party $i \in \{1, 2, \dots, K\}$, the current number of wins (wins_i^t) and losses (losses_i^t) can be obtained by verifying whether its confidence bounds within each constituency have separated accordingly from other parties. We also use leads_i^t to denote the number of constituencies in which i has polled the most votes yet, but has not yet won. Thus, we have $\mathrm{LCB}_i^t = \mathrm{wins}_i^t$, and $\mathrm{UCB}_i^t = C - \mathrm{losses}_i^t$.

Observe that the overall winner can be declared as soon as one party's LCB exceeds the UCB of all the other parties. On the other hand, when the winner is yet to be identified, one would ideally like to focus on "potential" winners, rather than query a constituency whose result does not seem relevant to the big picture. Taking cue from the LUCB algorithm for bandits Kalyanakrishnan et al. (2012), the first step under DCB is to identify two contenders for the top position: party $a^t = \mathrm{argmax}_{i \in \{1,2,\dots,K\}}(\mathrm{wins}_i^t + \mathrm{leads}_i^t)$, and party $b^t = \mathrm{argmax}_{i \in \{1,2,\dots,K\} \setminus \{a^t\}} \mathrm{UCB}_i^t$. Optimistic that sampling can reveal a win for a^t and a loss for b^t—which would take us closer to termination—DCB picks one "promising" constituency each for a^t and b^t. These constituencies, denoted c_1^t and c_2^t, are defined below for use with both 1v1 and 1vr confidence bounds. The idea is the same: c_1^t is the constituency in which a^t appears poised to win by a large margin, and c_2^t is the constituency in which b^t appears poised to lose by a large margin.

$$c_1^t = \begin{cases} \mathrm{argmax}_c \min_{j \in \{1,2,\dots,K\}} (\mathrm{UCB}(c, a^t, j) - \mathrm{LCB}(c, a^t, j)) & \text{(1v1)}, \\ \mathrm{argmax}_c \min_{j \in \{1,2,\dots,K\}} (\mathrm{UCB}(c, a^t) - \mathrm{LCB}(c, j)) & \text{(1vr)}, \end{cases}$$

$$c_2^t = \begin{cases} \mathrm{argmax}_c \max_{j \in \{1,2,\dots,K\}} (\mathrm{UCB}(c, j, b^t) - \mathrm{LCB}(c, j, b^t)) & \text{(1v1)}, \\ \mathrm{argmax}_c \max_{j \in \{1,2,\dots,K\}} (\mathrm{UCB}(c, j) - \mathrm{LCB}(c, b^t)) & \text{(1vr)}. \end{cases}$$

The outer "argmax" in all the definitions above is over all constituencies where the concerned party (a^t or b^t) is still in contention: that is, it has not yet won or lost the constituency. The DCB algorithm queries c_1^t and c_2^t at each time step t, and terminates once an overall winner has been identified.

G Winner Forecasting in Bihar Elections

Below we provide results from the 2015 Bihar state elections, a closer contest than the 2014 Indian national elections whose results were shown in Table 2 in Section 5.1. Of a total 242 seats in this election, 80 went to the winner, 71 to the second largest party, and 53 to the third largest.

Table 4: Sample complexity of various stopping rules when coupled with (1) round-robin (RR) polling of constituencies and (2) DCB. All experiments are run with mistake probability $\delta = 0.01$. Values shown are averages from 10 runs, and show one standard error. "Seats resolved" indicates the number of constituencies in which a winner was identified before the overall procedure terminated.

Algorithm	Bihar-2015 (242 seats)	
	Samples	Seats Resolved
RR-$\mathcal{A}_1$-1v1	4201429 ± 43932	221 ± 1
RR-$\mathcal{A}_1$-1vr	4841406 ± 58188	222 ± 1
RR-KLSN-1v1	2198678 ± 44514	221 ± 1
RR-KLSN-1vr	2668729 ± 47535	222 ± 2
RR-PPR-1v1	**1813213 ± 42081**	221 ± 1
RR-PPR-1vr	2054171 ± 40394	221 ± 1
DCB-$\mathcal{A}_1$-1v1	2301936 ± 36399	135 ± 1
DCB-$\mathcal{A}_1$-1vr	2481552 ± 21483	134 ± 1
DCB-KLSN-1v1	1127963 ± 22191	139 ± 2
DCB-KLSN-1vr	1312027 ± 22892	139 ± 1
DCB-PPR-1v1	**883389 ± 15581**	142 ± 2
DCB-PPR-1vr	993495 ± 18859	139 ± 1