An Efficient Algorithm for PAC Mode Estimation

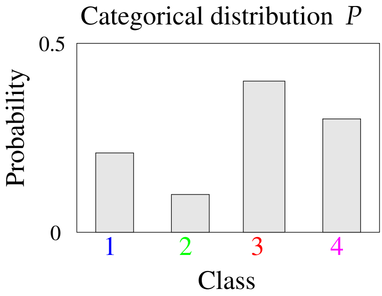
Shubham Anand Jain [2] Rohan Shah [1] Sanit Gupta [1]
Denil Mehta [1] Inderjeet Nair [1] Jian Vora [2] Sushil Khyalia [1]
Sourav Das [3] Vinay J. Ribeiro [1] Shivaram Kalyanakrishnan [1]

[1]Indian Institute of Technology Bombay

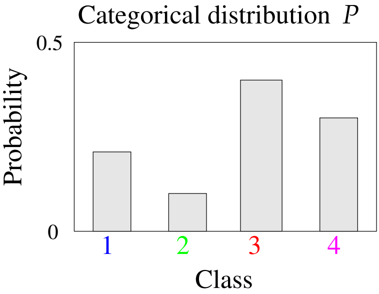
[2]Stanford University

[3]University of Illinois Urbana-Champaign

August 2023

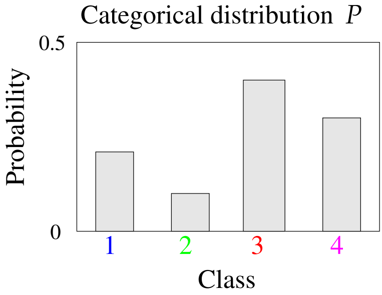

Categorical distribution P
0.5
Probability
0
1
2
3
4
Class

Mode Estimation



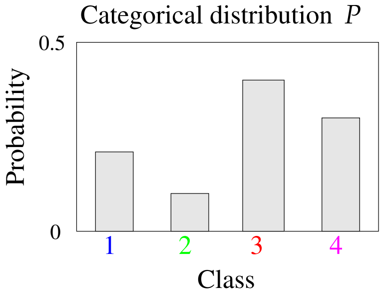
What is the **mode** of $\mathcal{P}$?

Mode Estimation



What is the **mode** of $\mathcal{P}$? Have to find out by **sampling** $\mathcal{P}$.

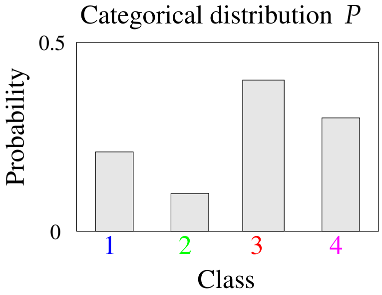
Mode Estimation



What is the **mode** of $\mathcal{P}$? Have to find out by **sampling** $\mathcal{P}$.

Sequence of samples from $\mathcal{P}$: 3

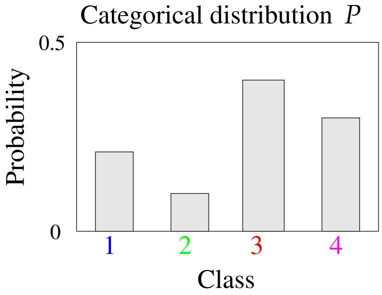
Mode Estimation



What is the **mode** of $\mathcal{P}$? Have to find out by **sampling** $\mathcal{P}$.

Sequence of samples from $\mathcal{P}$: 3, 1

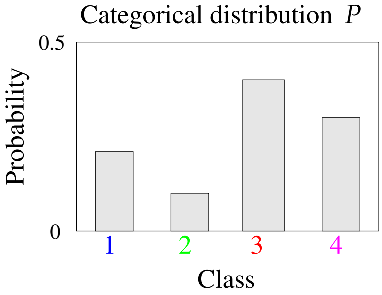
Mode Estimation



What is the **mode** of $\mathcal{P}$? Have to find out by **sampling** $\mathcal{P}$.

Sequence of samples from $\mathcal{P}$: 3, 1, 3

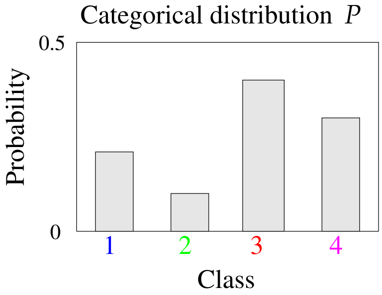
Mode Estimation



What is the **mode** of $\mathcal{P}$? Have to find out by **sampling** $\mathcal{P}$.

Sequence of samples from $\mathcal{P}$: 3, 1, 3, 4

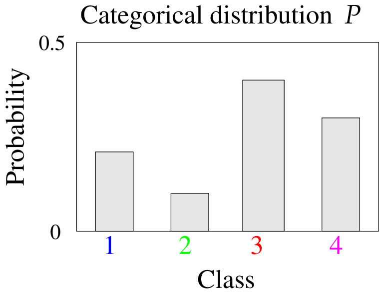
Mode Estimation



What is the **mode** of $\mathcal{P}$? Have to find out by **sampling** $\mathcal{P}$.

Sequence of samples from $\mathcal{P}$: 3, 1, 3, 4, 2, 1, 3, ...

Mode Estimation



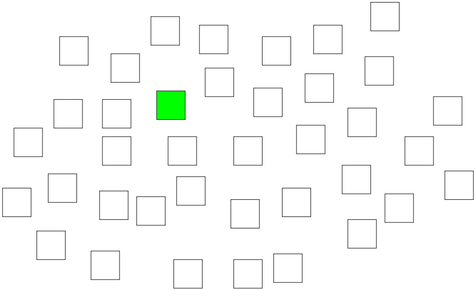
What is the **mode** of $\mathcal{P}$? Have to find out by **sampling** $\mathcal{P}$.

Sequence of samples from $\mathcal{P}$: 3, 1, 3, 4, 2, 1, 3, ...

When can we confidently stop sampling and declare the mode?

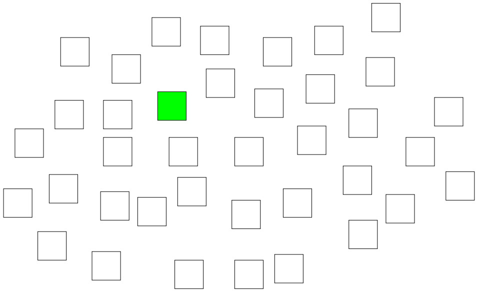
Verifying Smart Contracts in Permissionless Blockchains

- Ask a node for the result of a hard computation.
- Node returns "Result = ■".



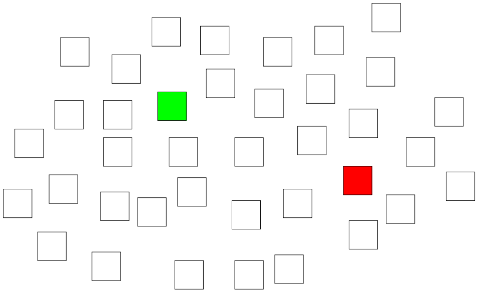
Verifying Smart Contracts in Permissionless Blockchains

- Ask a node for the result of a hard computation.
- Node returns "Result = ■".
- Should you trust this result?



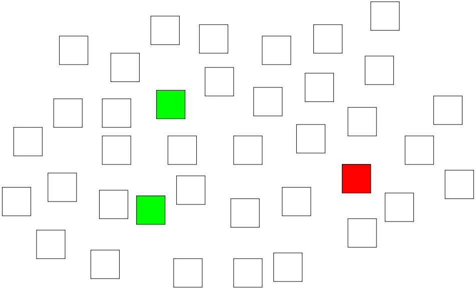
Verifying Smart Contracts in Permissionless Blockchains

- Ask a node for the result of a hard computation.
- Node returns "Result = ".
- Should you trust this result?



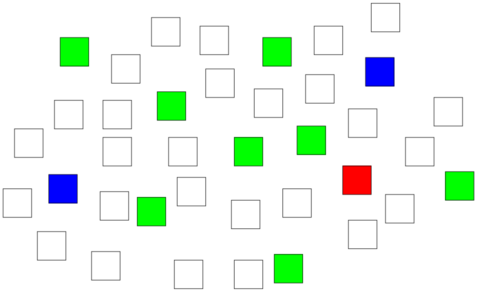
Verifying Smart Contracts in Permissionless Blockchains

- Ask a node for the result of a hard computation.
- Node returns "Result = ■".
- Should you trust this result?



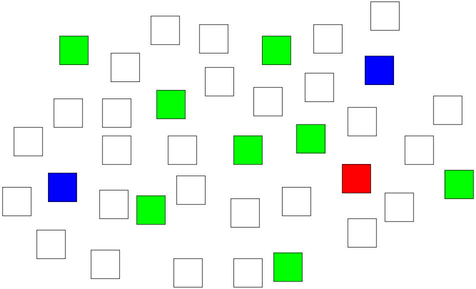
Verifying Smart Contracts in Permissionless Blockchains

- Ask a node for the result of a hard computation.
- Node returns "Result = ■".
- Should you trust this result?



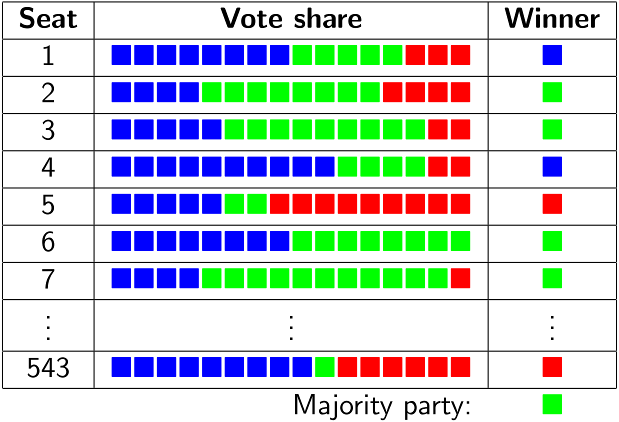
Verifying Smart Contracts in Permissionless Blockchains

- Ask a node for the result of a hard computation.
- Node returns "Result = ■".
- Should you trust this result?

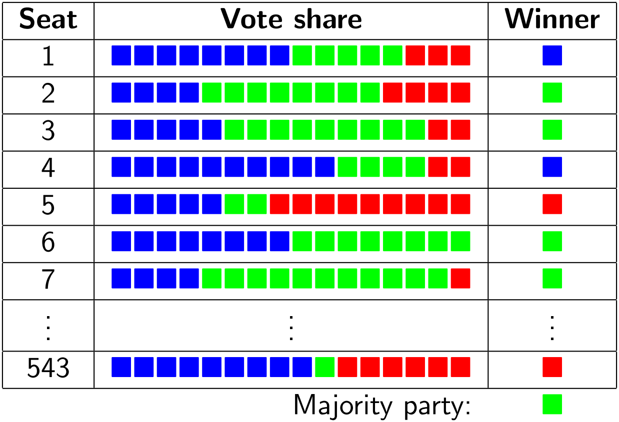


- Assume that most frequently returned result is correct.
- Determine most frequently returned result with sufficient confidence, while minimising number of queries.

Election Forecasting



Election Forecasting



- Confidently identify winner of each seat by sampling.
- Aggregate wins to predict overall winner.

Overview of Talk

1. PAC Formulation
2. Solutions for $K = 2$ classes
3. Generalising to $K \geq 2$ classes
4. Theoretical guarantees
5. Summary and outlook

Overview of Talk

1. PAC Formulation
2. Solutions for $K = 2$ classes
3. Generalising to $K \geq 2$ classes
4. Theoretical guarantees
5. Summary and outlook

Formal Problem Statement

- **Categorical distribution** $\mathcal{P} = (p, K)$:
  - $K \geq 2$ classes,
  - probability vector $p = \{p_1, p_2, \ldots, p_K\}$.

Formal Problem Statement

- **Categorical distribution** $\mathcal{P} = (p, K)$:
  - $K \geq 2$ classes,
  - probability vector $p = \{p_1, p_2, \ldots, p_K\}$.

- **Algorithm** $\mathcal{L}$: Set of sample sequences $\rightarrow \{\text{Continue}, 1, 2, \ldots, K\}$.

Formal Problem Statement

- **Categorical distribution** $\mathcal{P} = (p, K)$:
  - $K \geq 2$ classes,
  - probability vector $p = \{p_1, p_2, \ldots, p_K\}$.

- **Algorithm** $\mathcal{L}$: Set of sample sequences $\rightarrow \{\text{Continue}, 1, 2, \ldots, K\}$.

- **Require** algorithm $\mathcal{L}$ that
  - for every categorical distribution $\mathcal{P}$ with a unique mode,
  - for every mistake probability $\delta > 0$,

  stops and declares the mode of $\mathcal{P}$ with probability at least $1 - \delta$.

Formal Problem Statement

- **Categorical distribution** $\mathcal{P} = (p, K)$:
  - $K \geq 2$ classes,
  - probability vector $p = \{p_1, p_2, \ldots, p_K\}$.

- **Algorithm** $\mathcal{L}$: Set of sample sequences $\rightarrow \{\text{Continue}, 1, 2, \ldots, K\}$.

- **Require** algorithm $\mathcal{L}$ that
  - for every categorical distribution $\mathcal{P}$ with a unique mode,
  - for every mistake probability $\delta > 0$,

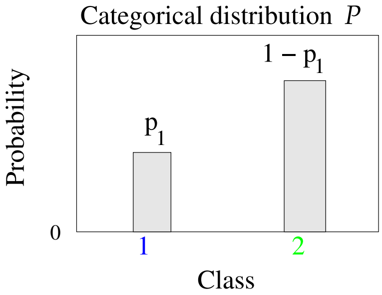
  stops and declares the mode of $\mathcal{P}$ with probability at least $1 - \delta$.

- **Prefer** δ-correct algorithm with low sample complexity.

Overview of Talk

1. PAC Formulation
2. Solutions for $K = 2$ classes
3. Generalising to $K \geq 2$ classes
4. Theoretical guarantees
5. Summary and outlook

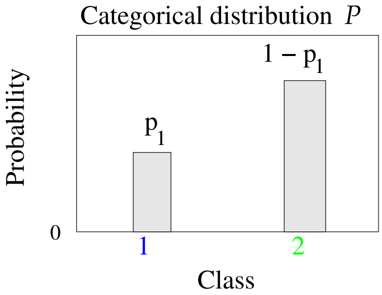
Solving the $K = 2$ Special Case



- Have to ascertain whether $p_1 > 0.5$ or $p_1 < 0.5$ based on samples

$$2, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1 \ldots$$

Solving the $K = 2$ Special Case



- Have to ascertain whether $p_1 > 0.5$ or $p_1 < 0.5$ based on samples

$$2, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1 \ldots$$

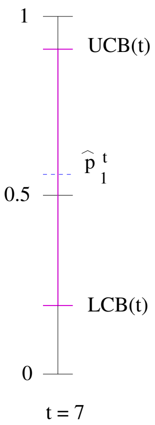
- Maintain lower confidence bound LCB and upper confidence bound UCB on p_1, stop if $LCB > 0.5$ or $UCB < 0.5$.

Using Confidence Bounds

- After t samples, let $\hat{p}_1^t$ be the empirical fraction of class 1.

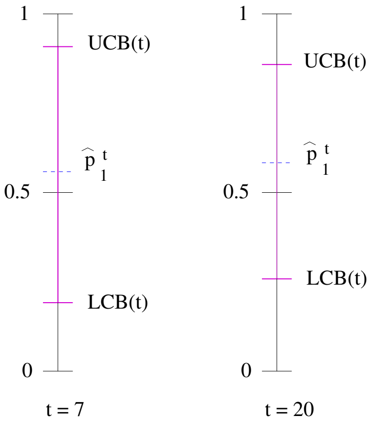
Using Confidence Bounds

- After t samples, let $\hat{p}_1^t$ be the empirical fraction of class 1.



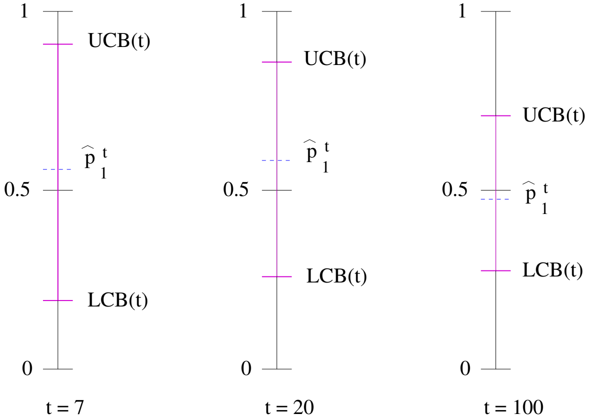
Using Confidence Bounds

- After t samples, let $\hat{p}_1^t$ be the empirical fraction of class 1.



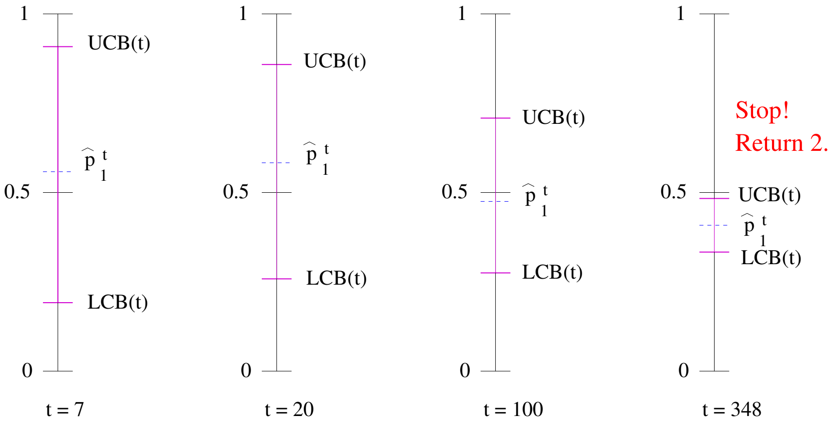
Using Confidence Bounds

- After t samples, let $\hat{p}_1^t$ be the empirical fraction of class 1.



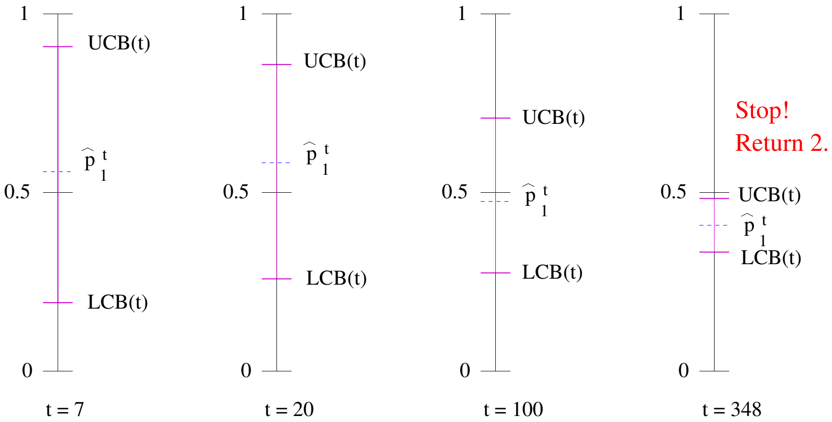
Using Confidence Bounds

- After t samples, let $\hat{p}_1^t$ be the empirical fraction of class 1.



Using Confidence Bounds

- After t samples, let $\hat{p}_1^t$ be the empirical fraction of class 1.



- Note that the stopping time is random.

Some Baselines

- Using Hoeffding's Inequality.

$$LCB(t), UCB(t) = \hat{p}_1^t \pm \beta(t, \delta),$$
$$\beta(t, \delta) = \sqrt{\frac{1}{2t} \ln\left(\frac{kt^\alpha}{\delta}\right)}.$$

Some Baselines

- Using Hoeffding's Inequality.

$$LCB(t), UCB(t) = \hat{p}_1^t \pm \beta(t, \delta),$$
$$\beta(t, \delta) = \sqrt{\frac{1}{2t} \ln\left(\frac{kt^\alpha}{\delta}\right)}.$$

- Using Empirical Bernstein bound ($\mathcal{A}_1$ algorithm (Shah et al., 2020)).

$$LCB(t), UCB(t) = \hat{p}_1^t \pm \beta(t, \delta),$$
$$\beta(t, \delta) = \sqrt{\frac{2V^t \ln(4t^2/\delta)}{t}} + \frac{7\ln(4t^2/\delta)}{3(t-1)},$$

where V^t is the empirical variance.

Some Baselines

- Using Hoeffding's Inequality.

$$LCB(t), UCB(t) = \hat{p}_1^t \pm \beta(t, \delta),$$
$$\beta(t, \delta) = \sqrt{\frac{1}{2t} \ln\left(\frac{kt^\alpha}{\delta}\right)}.$$

- Using Empirical Bernstein bound ($\mathcal{A}_1$ algorithm (Shah et al., 2020)).

$$LCB(t), UCB(t) = \hat{p}_1^t \pm \beta(t, \delta),$$
$$\beta(t, \delta) = \sqrt{\frac{2V^t \ln(4t^2/\delta)}{t}} + \frac{7\ln(4t^2/\delta)}{3(t-1)},$$

where V^t is the empirical variance.

- "Self-normalised" inequality to avoid naïve union bound over t. KL-SN, Garivier (2013).

PPR Martingale Confidence Sequences–1

- Introduced by Waudby-Smith and Ramdas (2020).

PPR Martingale Confidence Sequences–1

- Introduced by Waudby-Smith and Ramdas (2020).
- Let us maintain a belief π on p_1, starting with uniform prior $\pi^0(q) = 1$ for $q \in [0, 1]$.

PPR Martingale Confidence Sequences–1

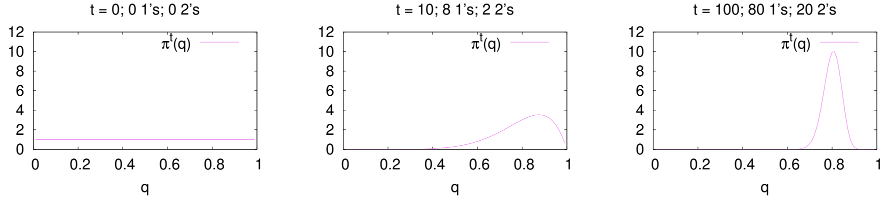
- Introduced by Waudby-Smith and Ramdas (2020).
- Let us maintain a belief π on p_1, starting with uniform prior $\pi^0(q) = 1$ for $q \in [0, 1]$.
- Bayesian update on receiving sample $x^t \in \{1, 2\}$:

$$\pi^t(q) = \frac{\pi^{t-1}(q) \cdot (q)^{\mathbf{1}[x^t=1]} \cdot (1-q)^{\mathbf{1}[x^t=2]}}{\int_{\rho=0}^{1} \pi^{t-1}(\rho) \cdot (\rho)^{\mathbf{1}[x^t=1]} \cdot (1-\rho)^{\mathbf{1}[x^t=2]} d\rho}.$$

PPR Martingale Confidence Sequences–1

- Introduced by Waudby-Smith and Ramdas (2020).
- Let us maintain a belief π on p_1, starting with uniform prior $\pi^0(q) = 1$ for $q \in [0, 1]$.
- Bayesian update on receiving sample $x^t \in \{1, 2\}$:

$$\pi^t(q) = \frac{\pi^{t-1}(q) \cdot (q)^{\mathbf{1}[x^t=1]} \cdot (1-q)^{\mathbf{1}[x^t=2]}}{\int_{\rho=0}^{1} \pi^{t-1}(\rho) \cdot (\rho)^{\mathbf{1}[x^t=1]} \cdot (1-\rho)^{\mathbf{1}[x^t=2]} d\rho}.$$



PPR Martingale Confidence Sequences–2

- The Prior-Posterior-Ratio is given by

$$R^t(q) \stackrel{\text{def}}{=} \frac{\pi^0(q)}{\pi^t(q)}.$$

PPR Martingale Confidence Sequences–2

- The Prior-Posterior-Ratio is given by

$$R^t(q) \stackrel{\text{def}}{=} \frac{\pi^0(q)}{\pi^t(q)}.$$

- Define the confidence set

$$C^t \stackrel{\text{def}}{=} \left\{ q : R^t(q) < \frac{1}{\delta} \right\}.$$

PPR Martingale Confidence Sequences–2

- The Prior-Posterior-Ratio is given by

$$R^t(q) \stackrel{\text{def}}{=} \frac{\pi^0(q)}{\pi^t(q)}.$$

- Define the confidence set

$$C^t \stackrel{\text{def}}{=} \left\{ q : R^t(q) < \frac{1}{\delta} \right\}.$$

- We have the “anytime” guarantee that

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

PPR Martingale Confidence Sequences–2

- The Prior-Posterior-Ratio is given by

$$R^t(q) \stackrel{\text{def}}{=} \frac{\pi^0(q)}{\pi^t(q)}.$$

- Define the confidence set

$$C^t \stackrel{\text{def}}{=} \left\{ q : R^t(q) < \frac{1}{\delta} \right\}.$$

- We have the "anytime" guarantee that

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

- Proof by establishing that $R^t(p_1)$ is a martingale, then applying Ville's inequality on nonnegative supermartingales.

PPR Martingale Confidence Sequences–2

- The Prior-Posterior-Ratio is given by

$$R^t(q) \stackrel{\text{def}}{=} \frac{\pi^0(q)}{\pi^t(q)}.$$

- Define the confidence set

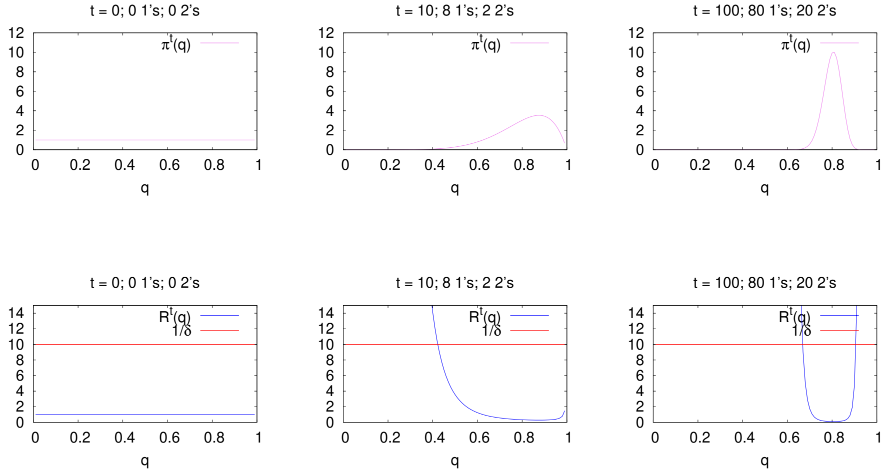
$$C^t \stackrel{\text{def}}{=} \left\{ q : R^t(q) < \frac{1}{\delta} \right\}.$$

- We have the "anytime" guarantee that

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

- Proof by establishing that $R^t(p_1)$ is a martingale, then applying Ville's inequality on nonnegative supermartingales.
- Method more generally applicable, even for estimating multiple parameters simultaneously.

PPR Martingale Confidence Sequences: Illustration



PPR-Bernoulli Stopping Rule

- For our $K = 2$ case, the belief distribution π^t is Beta.

PPR-Bernoulli Stopping Rule

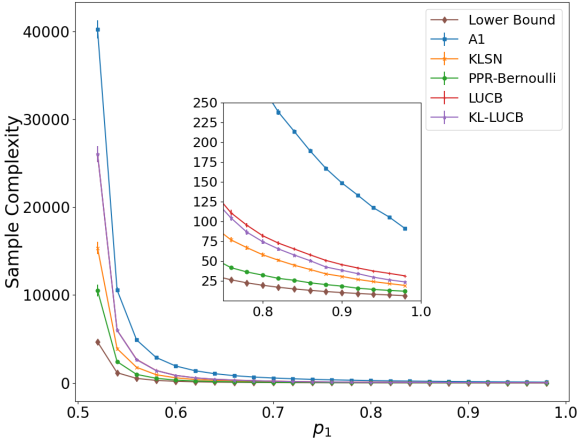
- For our $K = 2$ case, the belief distribution π^t is Beta.
- Let first$(t) \in \{1, 2\}$ denote the more frequent class in the data, and second$(t) \in \{1, 2\}$ the other class. Let s_i^t be the number of occurrences of class $i \in \{1, 2\}$. We obtain this stopping rule.

PPR-Bernoulli Stopping Rule

- For our $K = 2$ case, the belief distribution π^t is Beta.
- Let first$(t) \in \{1, 2\}$ denote the more frequent class in the data, and second$(t) \in \{1, 2\}$ the other class. Let s_i^t be the number of occurrences of class $i \in \{1, 2\}$. We obtain this stopping rule.

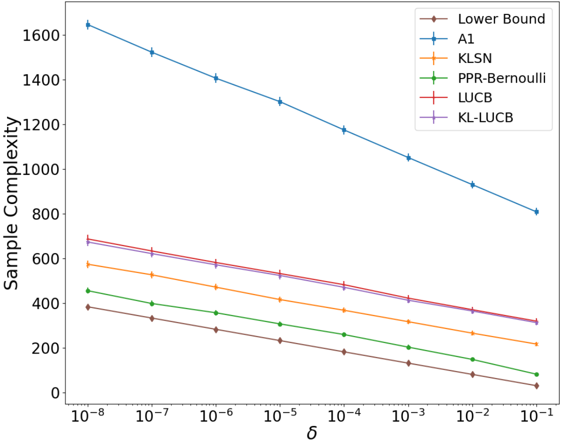
PPR-Bernoulli: Stop, declare first(t) as mode iff Beta $\left(\frac{1}{2}; s^t_{\text{first}(t)} + 1, s^t_{\text{second}(t)} + 1\right) \leq \delta$.

Comparison of Stopping Rules for $K = 2$



$$\delta = 0.01$$

Comparison of Stopping Rules for $K = 2$



$$p_1 = 0.65$$

Overview of Talk

1. PAC Formulation
2. Solutions for $K = 2$ classes
3. Generalising to $K \geq 2$ classes
4. Theoretical guarantees
5. Summary and outlook

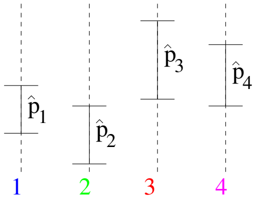
Two Approaches to Generalise

$$3, 1, 3, 4, 2, 1, 3, 2, 4, 3, 3, 1, 2, 3, \ldots$$

Two Approaches to Generalise

3, 1, 3, 4, 2, 1, 3, 2, 4, 3, 3, 1, 2, 3, ...

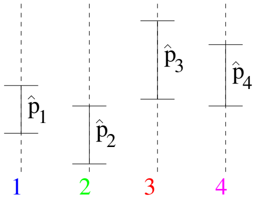
1-versus-rest approach



- For each class i, estimate underlying probability p_i.
- Stop and declare i as mode if LCB on p_i exceeds UCB on p_j for all $j \neq i$.

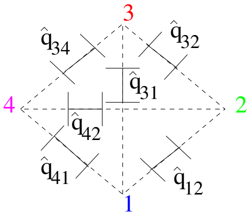
3, 1, 3, 4, 2, 1, 3, 2, 4, 3, 3, 1, 2, 3, . . .

1-versus-rest approach



- For each class i, estimate underlying probability p_i.
- Stop and declare i as mode if LCB on p_i exceeds UCB on p_j for all $j \neq i$.

1-versus-1 approach

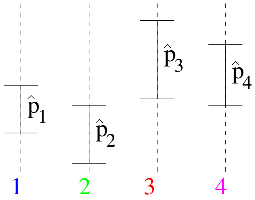


- For each pair of classes (i, j), estimate $q_{ij} = \frac{p_i}{p_i + p_j}$.
- Stop and declare i as mode if for all $j \neq i$, q_{ij} confidently exceeds 0.5.

Two Approaches to Generalise

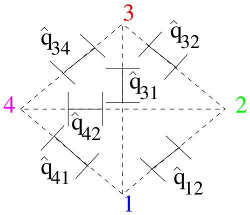
3, 1, 3, 4, 2, 1, 3, 2, 4, 3, 3, 1, 2, 3, . . .

1-versus-rest approach



- For each class i, estimate underlying probability p_i.
- Stop and declare i as mode if LCB on p_i exceeds UCB on p_j for all $j \neq i$.

1-versus-1 approach



- For each pair of classes (i, j), estimate $q_{ij} = \frac{p_i}{p_i + p_j}$.
- Stop and declare i as mode if for all $j \neq i$, q_{ij} confidently exceeds 0.5.

1-versus-1 appears to waste samples, but . . .

Empirical Comparison: 1vr and 1v1

Sample complexity

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

PPR-1v1 for PAC Mode Estimation

- We stop when some class i beats all classes $j \neq i$.

PPR-1v1 for PAC Mode Estimation

- We stop when some class i beats all classes $j \neq i$.
- When we stop, clearly i must be $\text{first}(t)$.

PPR-1v1 for PAC Mode Estimation

- We stop when some class i beats all classes $j \neq i$.
- When we stop, clearly i must be $\text{first}(t)$.
- If $\text{first}(t)$ beats all $j \neq \text{first}(t)$, then $\text{first}(t)$ must beat $\text{second}(t)$, and vice versa).

PPR-1v1 for PAC Mode Estimation

- We stop when some class i beats all classes $j \neq i$.
- When we stop, clearly i must be $\text{first}(t)$.
- If $\text{first}(t)$ beats all $j \neq \text{first}(t)$, then $\text{first}(t)$ must beat $\text{second}(t)$, and vice versa).
- It is sufficient for us check whether $\text{first}(t)$ beats $\text{second}(t)$!

PPR-1v1 for PAC Mode Estimation

- We stop when some class i beats all classes $j \neq i$.
- When we stop, clearly i must be first(t).
- If first(t) beats all $j \neq$ first(t), then first(t) must beat second(t), and vice versa).
- It is sufficient for us check whether first(t) beats second(t)!

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

PPR-1v1 for PAC Mode Estimation

- We stop when some class i beats all classes $j \neq i$.
- When we stop, clearly i must be first(t).
- If first(t) beats all $j \neq$ first(t), then first(t) must beat second(t), and vice versa).
- It is sufficient for us check whether first(t) beats second(t)!

PPR-1v1: Stop and declare first(t) as mode iff

$$\text{Beta}\left(\frac{1}{2}; s^t_{\text{first}(t)} + 1, s^t_{\text{second}(t)} + 1\right) \leq \frac{\delta}{K-1}.$$

- Computationally light compared to PPR-1vr, and other KL-divergence-based rules (which have to perform a numerical computation at each step).

Overview of Talk

1. PAC Formulation
2. Solutions for $K = 2$ classes
3. Generalising to $K \geq 2$ classes
4. Theoretical guarantees
5. Summary and outlook

Asymptotic Optimality

- **Lower bound** [Shah *et al.*, 2020]: Any δ-correct algorithm requires at least $\mathrm{LB}(\mathcal{P}, \delta)$ samples in expectation, where

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

Asymptotic Optimality

- **Lower bound** [Shah *et al.*, 2020]: Any δ-correct algorithm requires at least $\mathrm{LB}(\mathcal{P}, \delta)$ samples in expectation, where

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

- **PPR-1v1 upper bound**: Let $\tau(\mathcal{P}, \delta)$ be the expected stopping time of PPR-1v1 on $\mathcal{P}$ when run with mistake probability δ. Then,

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

Asymptotic Optimality

- **Lower bound** [Shah *et al.*, 2020]: Any δ-correct algorithm requires at least $\mathsf{LB}(\mathcal{P}, \delta)$ samples in expectation, where

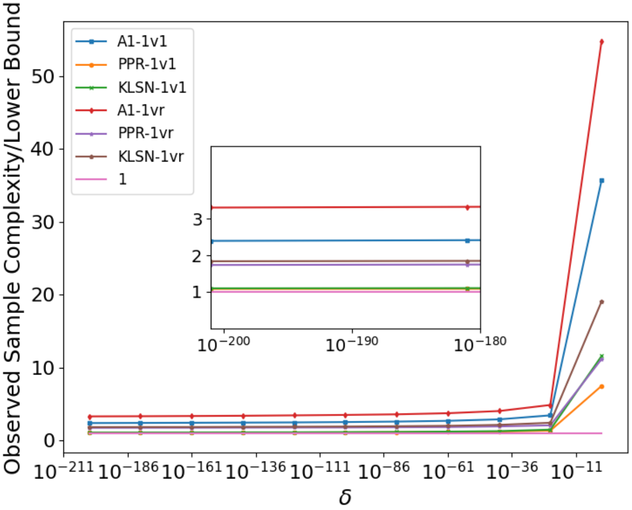
$$\mathsf{LB}(\mathcal{P}, \delta) \stackrel{\text{def}}{=} \sup_{\mathcal{P}' : \text{mode}(\mathcal{P}') \neq \text{mode}(\mathcal{P})} \frac{1}{\mathsf{KL}(\mathcal{P} || \mathcal{P}')} \ln\left(\frac{1}{2.4\delta}\right).$$

- **PPR-1v1 upper bound**: Let $\tau(\mathcal{P}, \delta)$ be the expected stopping time of PPR-1v1 on $\mathcal{P}$ when run with mistake probability δ. Then,

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

- Proof uses probability, functions, some results from Garivier and Kaufmann (2016) (who obtain similar result for PAC bandits).

Asymptotic Optimality (continued)



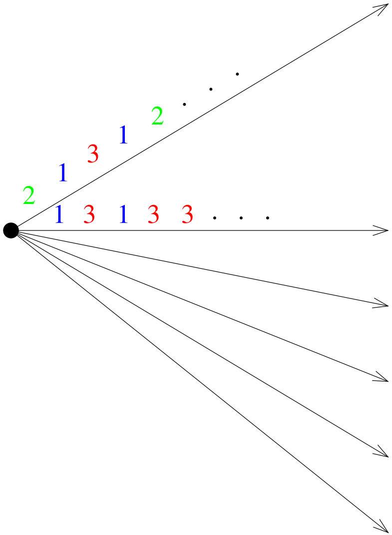
Result from $\mathcal{P}_3$

Uniform Domination

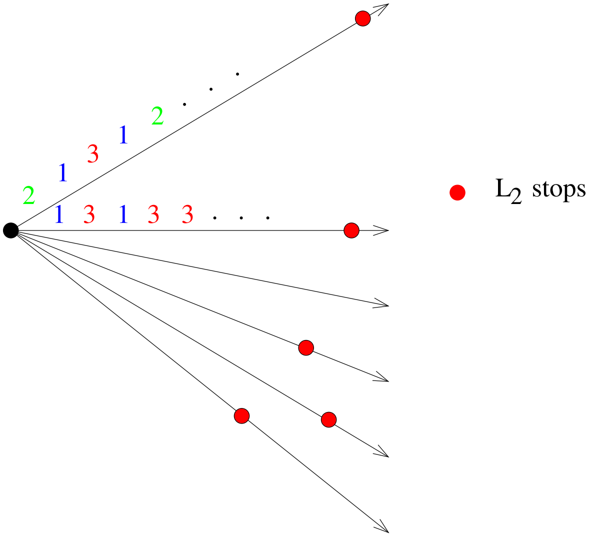
- Let $X = x^1, x^2, \ldots$ be an infinite sequence of samples from $\mathcal{P}$.
- For algorithm L, let $T(L, X)$ denote the stopping time of L on X.
- Define algorithm L_1 to uniformly dominate algorithm L_2 if for every sequence X, $\delta \in (0, 1)$,

$$T(L_2, X) < \infty \implies T(L_1, X) \leq T(L_2, X).$$

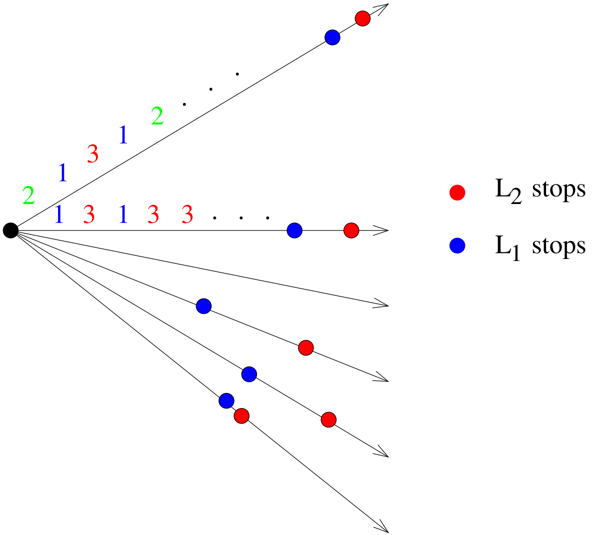
Uniform Domination



Uniform Domination



Uniform Domination



Uniform Domination: Results

- Hoeffding-1v1 uniformly dominates Hoeffding-1vr.

Uniform Domination: Results

- Hoeffding-1v1 uniformly dominates Hoeffding-1vr.
- $\mathcal{A}_1$-1v1 uniformly dominates $\mathcal{A}_1$-1vr.

Uniform Domination: Results

- Hoeffding-1v1 uniformly dominates Hoeffding-1vr.
- $\mathcal{A}_1$-1v1 uniformly dominates $\mathcal{A}_1$-1vr.
- PPR-1v1 uniformly dominates $\mathcal{A}_1$-1v1.

Uniform Domination: Results

- Hoeffding-1v1 uniformly dominates Hoeffding-1vr.
- $\mathcal{A}_1$-1v1 uniformly dominates $\mathcal{A}_1$-1vr.
- PPR-1v1 uniformly dominates $\mathcal{A}_1$-1v1.
- Does PPR-1v1 uniformly dominate PPR-1vr?

Uniform Domination: Results

- Hoeffding-1v1 uniformly dominates Hoeffding-1vr.
- $\mathcal{A}_1$-1v1 uniformly dominates $\mathcal{A}_1$-1vr.
- PPR-1v1 uniformly dominates $\mathcal{A}_1$-1v1.
- Does PPR-1v1 uniformly dominate PPR-1vr? Yet a conjecture!

Uniform Domination: Results

- Hoeffding-1v1 uniformly dominates Hoeffding-1vr.
- $\mathcal{A}_1$-1v1 uniformly dominates $\mathcal{A}_1$-1vr.
- PPR-1v1 uniformly dominates $\mathcal{A}_1$-1v1.
- Does PPR-1v1 uniformly dominate PPR-1vr? Yet a conjecture!
- These proofs done by showing

$$L_2 \text{ terminates on } X \implies L_1 \text{ terminates on } X.$$

Overview of Talk

1. PAC Formulation
2. Solutions for $K = 2$ classes
3. Generalising to $K \geq 2$ classes
4. Theoretical guarantees
5. Summary and outlook

Concluding Notes

- Have addressed PAC mode estimation problem.
- PPR an elegant approach to handle random stopping time.
- 1v1 outperforms 1vr to generalise to $K \geq 2$ classes.
- PPR-1v1 simple, asymptoticaly optimal, efficient in practice.
- Flexible to apply in practice, validated on two distinct applications.

- Code base released along with AISTATS 2022 paper.

- Conjecture on PPR-1v1 termination preceding PPR-1vr termination remains unresolved.
- Can PPR help in bandits, too?
- Do you have a practical application of mode estimation?