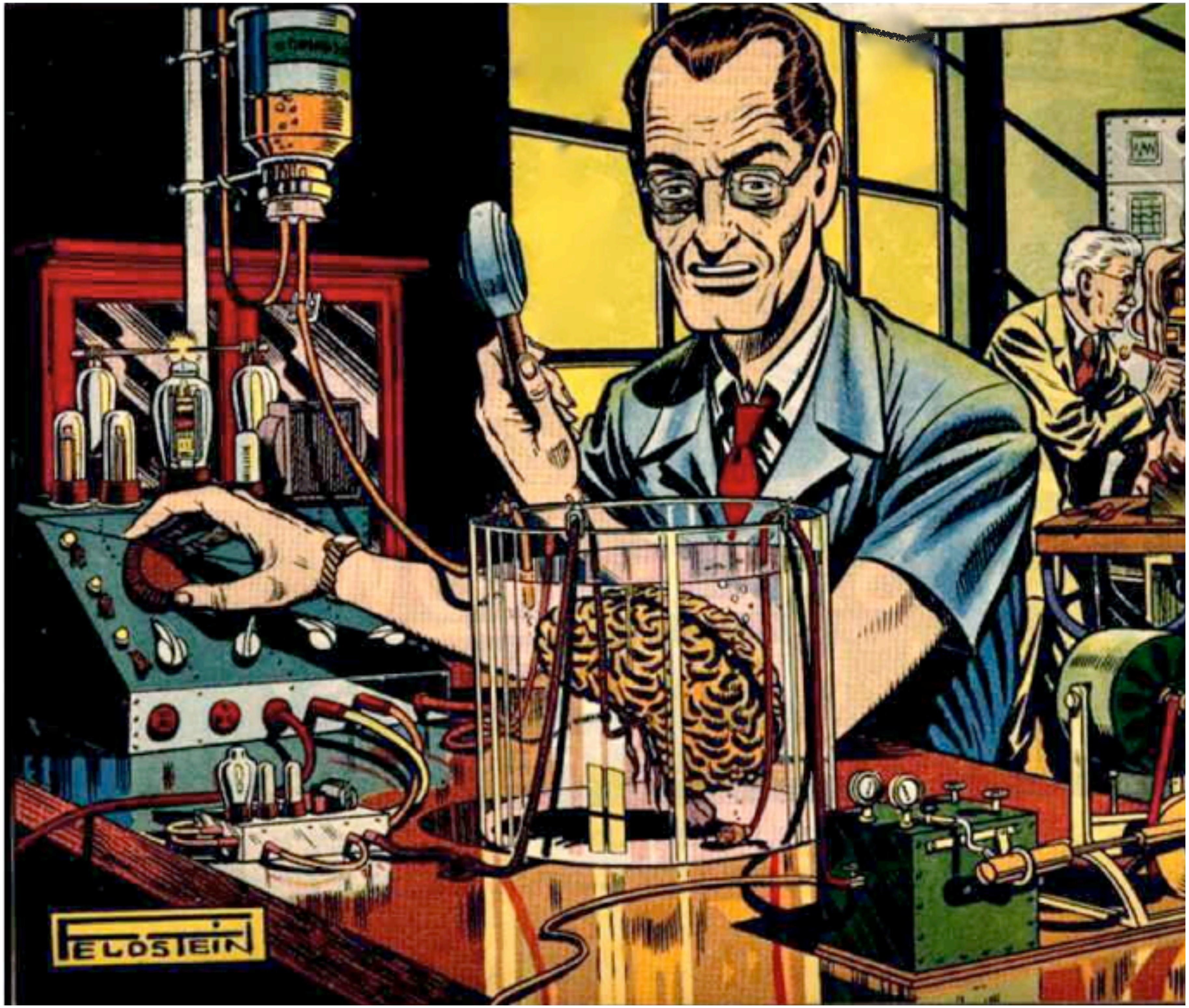
Picking Brains with Bandits



Simons, March 2015

Rob Nowak
University of Wisconsin-Madison

Motivation

Imagine we have n items and we want to know which one people prefer

n beers
n papers
n resumes

We also have a large pool of people who can judge the items.
How should we allocate the judges?

uniform distribution of judges over items

adaptive allocation of judges to (hopefully) focus on top items

Multi-Armed Bandit

n arms (one for each item)

$\mu_1 > \mu_2 \geq \cdots \geq \mu_n$, expected rating of each item

(order is unknown)

$x_{ij} \sim P_{\mu_i}$, random rating from judge j

$x_{ij} \sim P_{\mu_i}$, random rating from judge j

(assume judges are iid)

$\widehat{\mu}_{i,t_i} = \frac{1}{t_i} \sum_{j=1}^{t_i} x_{ij}$, empirical mean from t_i ratings

$$\boxed{\text{Use } \{\widehat{\mu}_{i,t_i}\} \text{ to choose } \widehat{i} \text{ so that } \mathbb{P}(\widehat{i} \neq 1) \leq \delta}$$

Confidence Intervals

Assume P_{μ_i} are subGaussian:

$$\mathbb{P}(|\widehat{\mu}_{i,t} - \mu_i| \geq \epsilon) \ \leq \ 2e^{-ct\epsilon^2} \ , \ \text{for some } c > 0$$

x_i iid Bernoulli → Chernoff

x_i iid bounded → Hoeffding

x_i iid Gaussian → $e^{-t\epsilon^2/2}$

With probability at least $1 - \delta$

$$\widehat{\mu}_{i,t} - \sqrt{\frac{c}{t}\log\frac{2}{\delta}} \ \leq \ \mu_i \ \leq \ \widehat{\mu}_{i,t} + \sqrt{\frac{c}{t}\log\frac{2}{\delta}}$$

Confidence Intervals

With probability at least $1 - \delta$

for fixed i and t

$$\widehat{\mu}_{i,t} - \sqrt{\frac{c}{t} \log \frac{2}{\delta}} \ \leq \ \mu_i \ \leq \ \widehat{\mu}_{i,t} + \sqrt{\frac{c}{t} \log \frac{2}{\delta}}$$

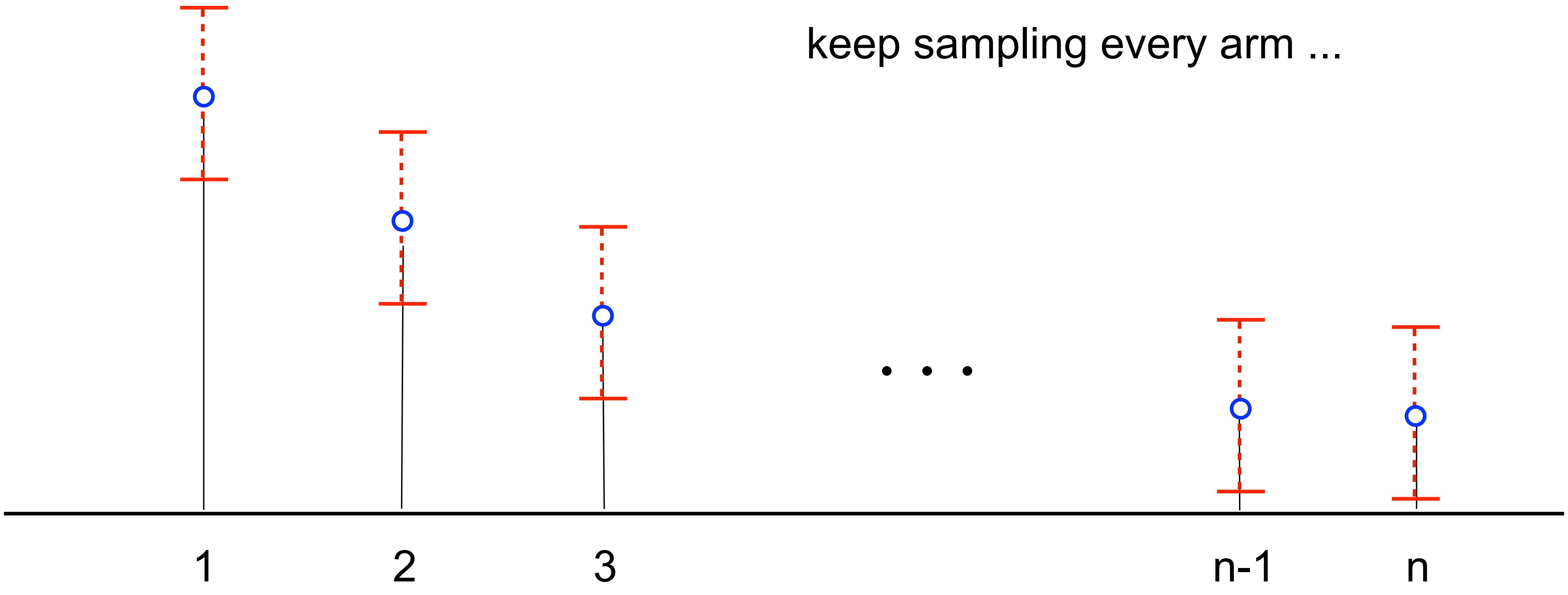
$\delta \to \delta/n$: for all i and fixed t

$$\widehat{\mu}_{i,t} - \sqrt{\frac{c}{t} \log \frac{2n}{\delta}} \ \leq \ \mu_i \ \leq \ \widehat{\mu}_{i,t} + \sqrt{\frac{c}{t} \log \frac{2n}{\delta}}$$

$\delta \to \delta/(2nt^2)$: for all i and all t

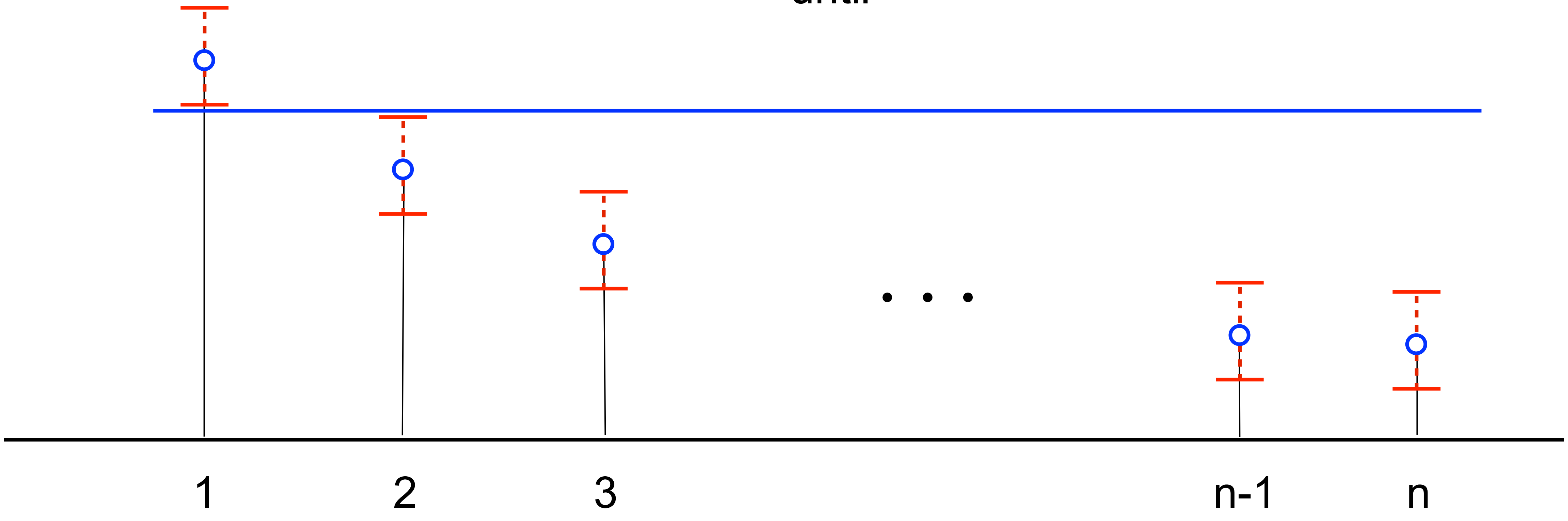
$$\widehat{\mu}_{i,t} - \sqrt{\frac{c}{t} \log \frac{4nt^2}{\delta}} \ \leq \ \mu_i \ \leq \ \widehat{\mu}_{i,t} + \sqrt{\frac{c}{t} \log \frac{4nt^2}{\delta}}$$

Non-Adaptive Scheme

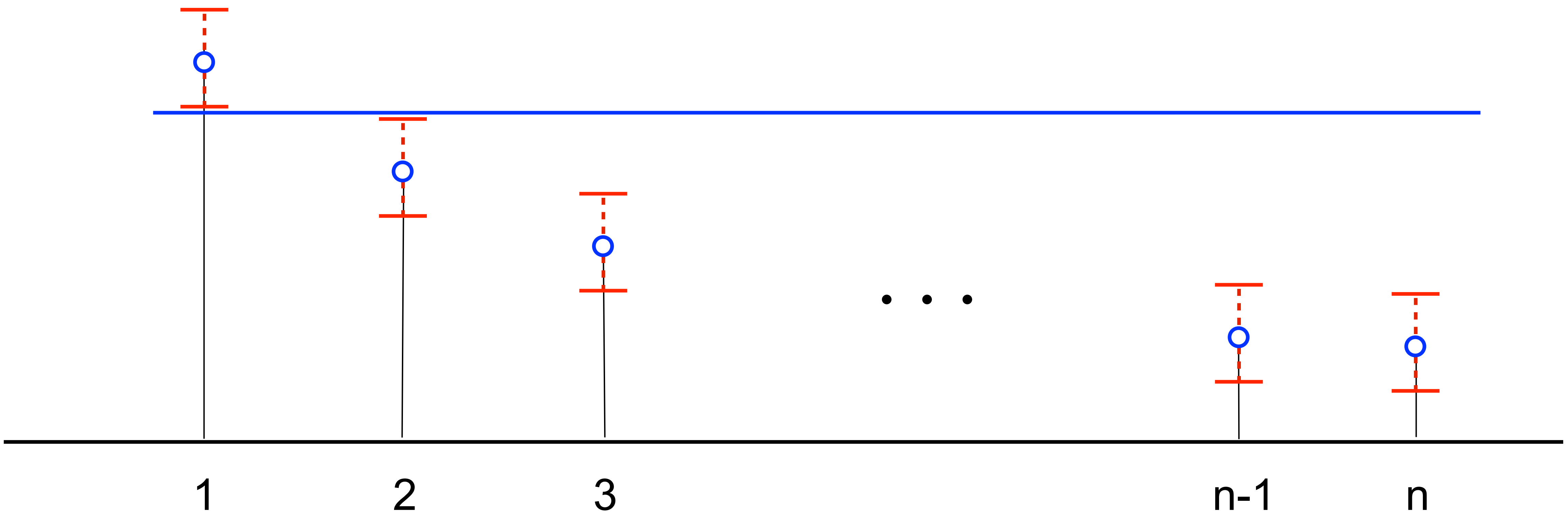


keep sampling every arm ...

until



Non-Adaptive Scheme



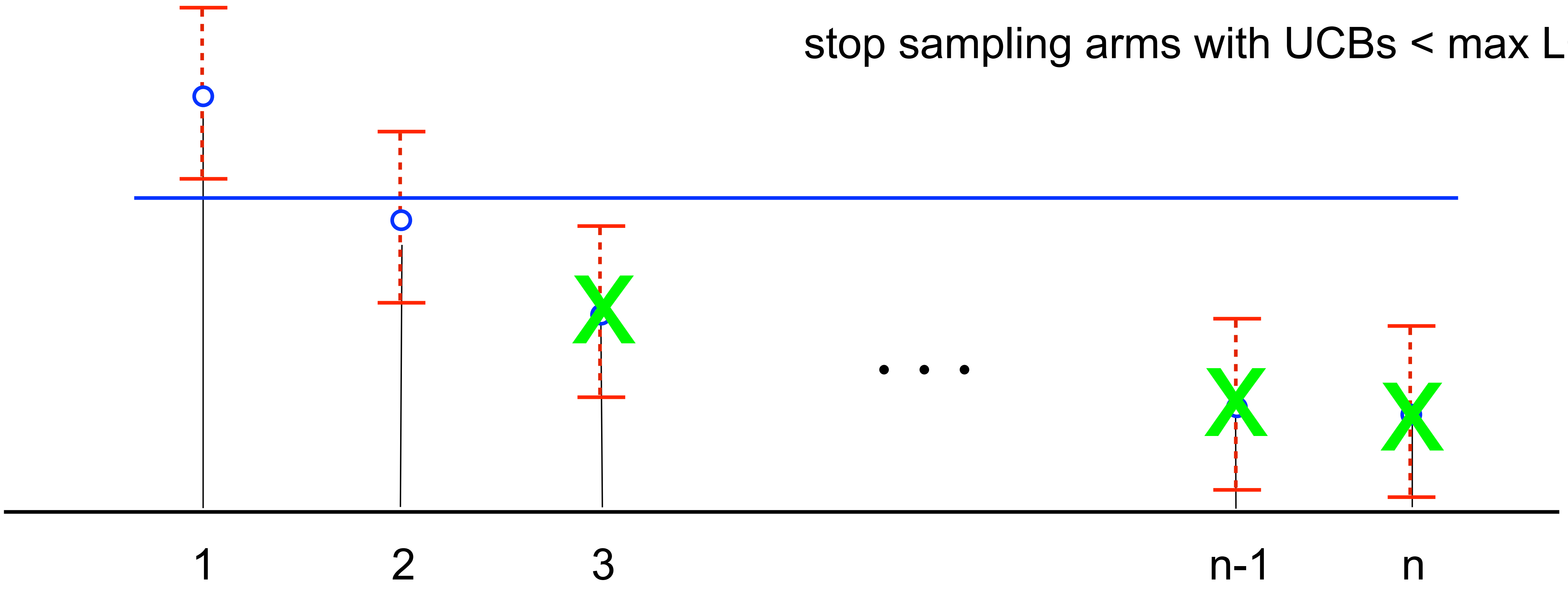
satisfied if

$$4\sqrt{\frac{c}{t}\log\frac{4nt^2}{\delta}} \leq \mu_1 - \mu_2 \;\; =: \;\; \Delta_2$$

$$\Rightarrow \;\; t = O\left(\Delta_2^{-2}\log\frac{n\Delta_2^{-2}}{\delta}\right) \text{ samples/arm suffice}$$

$$\text{Total samples} \;\; T \;\; = \;\; O\left(n\,\Delta_2^{-2}\log\frac{n\Delta_2^{-2}}{\delta}\right)$$

Adaptive Scheme



stop sampling arms with UCBs < max LCB

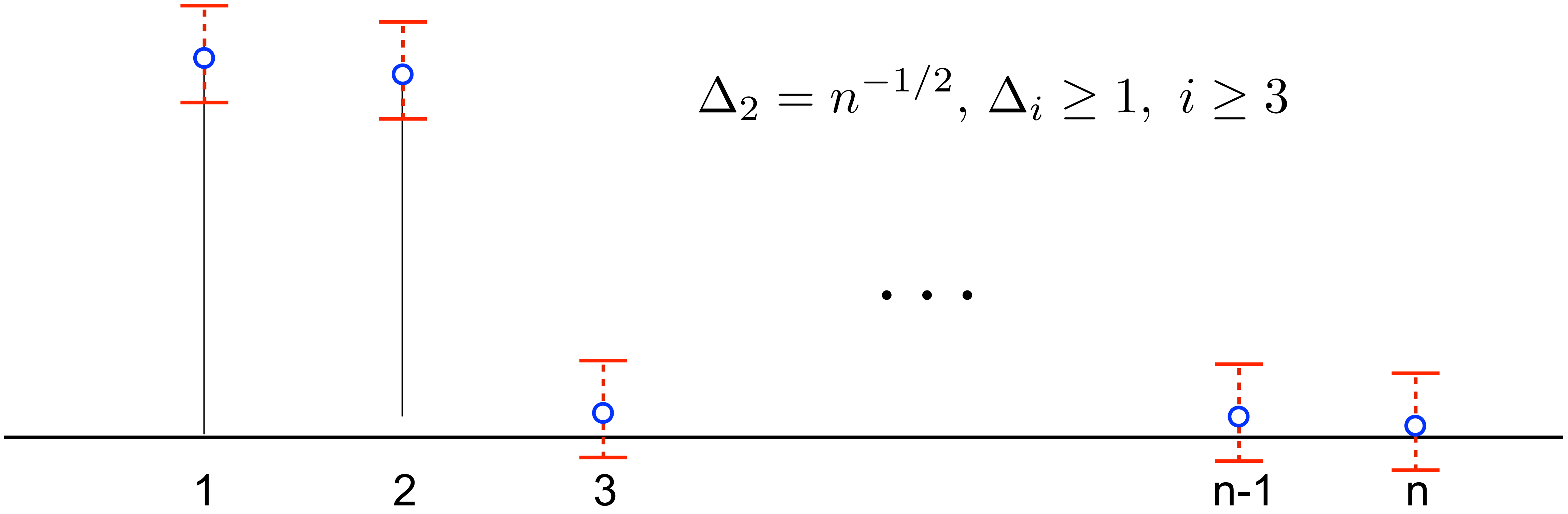
$$i\text{th arm removed after } t_i \;=\; O\left(\Delta_i^{-2}\log\frac{n\Delta_i^{-2}}{\delta}\right) \;\text{ samples}$$

$$\Delta_i := \mu_1 - \mu_i$$

$$\text{Total samples } \; T \;=\; O\left(\sum_{i\geq 2}\Delta_i^{-2}\log\frac{n\Delta_i^{-2}}{\delta}\right)$$

Even-Dar et al (2006)

Example



$\Delta_2 = n^{-1/2},\ \Delta_i \geq 1,\ \ i \geq 3$

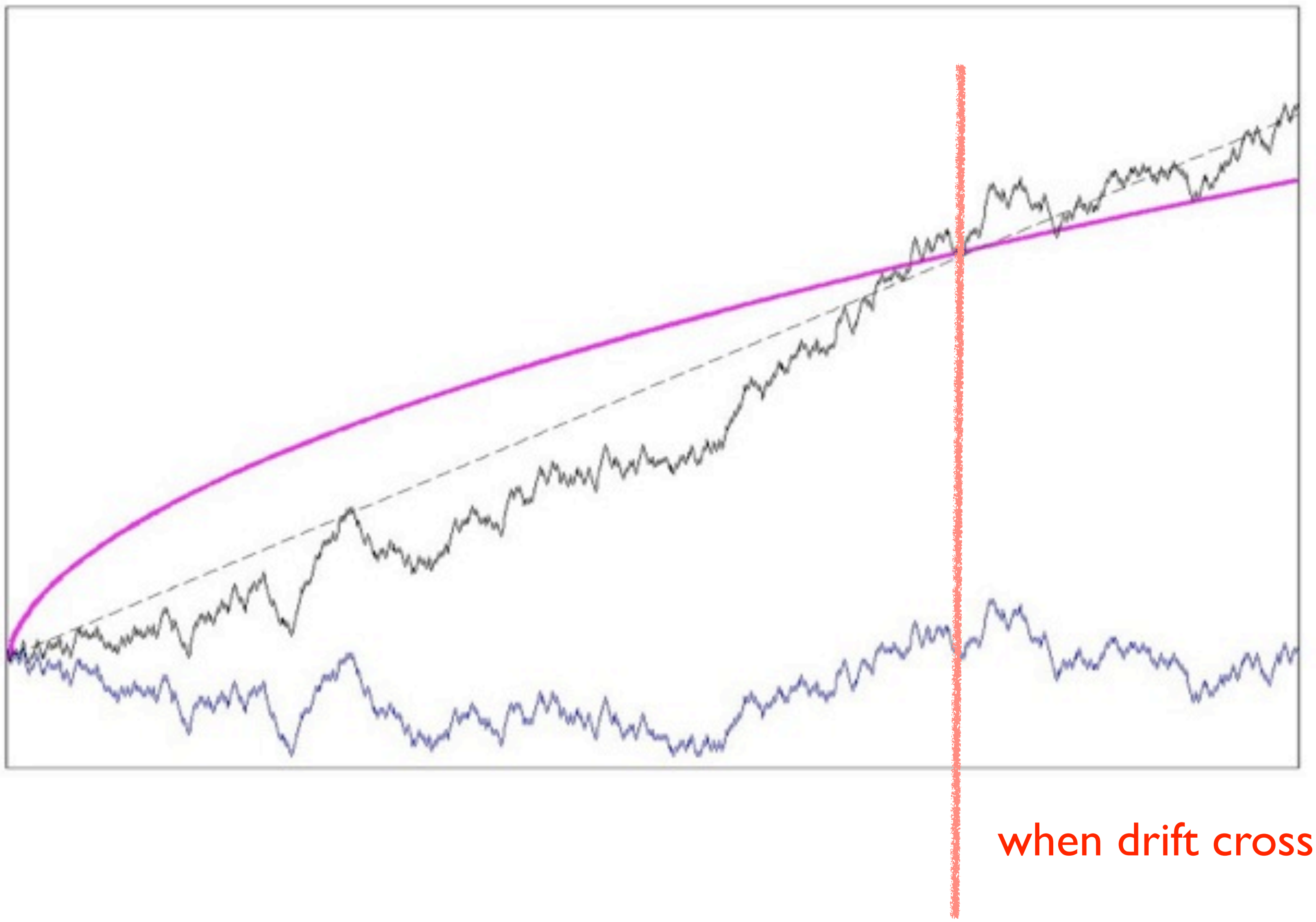
non-adaptive: $T = O(n^2 \log n)$

adaptive: $T = O(n \log n)$

is $\log n$ factor necessary?

Two-Arm Case

Test: $\sum_{j=1}^{t}(x_{1,j} - x_{2,j}) \geq 0$



$\Delta_2 > 0$

walk + drift $\Delta_2\, t$

$\sqrt{2t \log\log t}$

zero-mean walk

$\Delta_2 = 0$

when drift crosses LIL bound

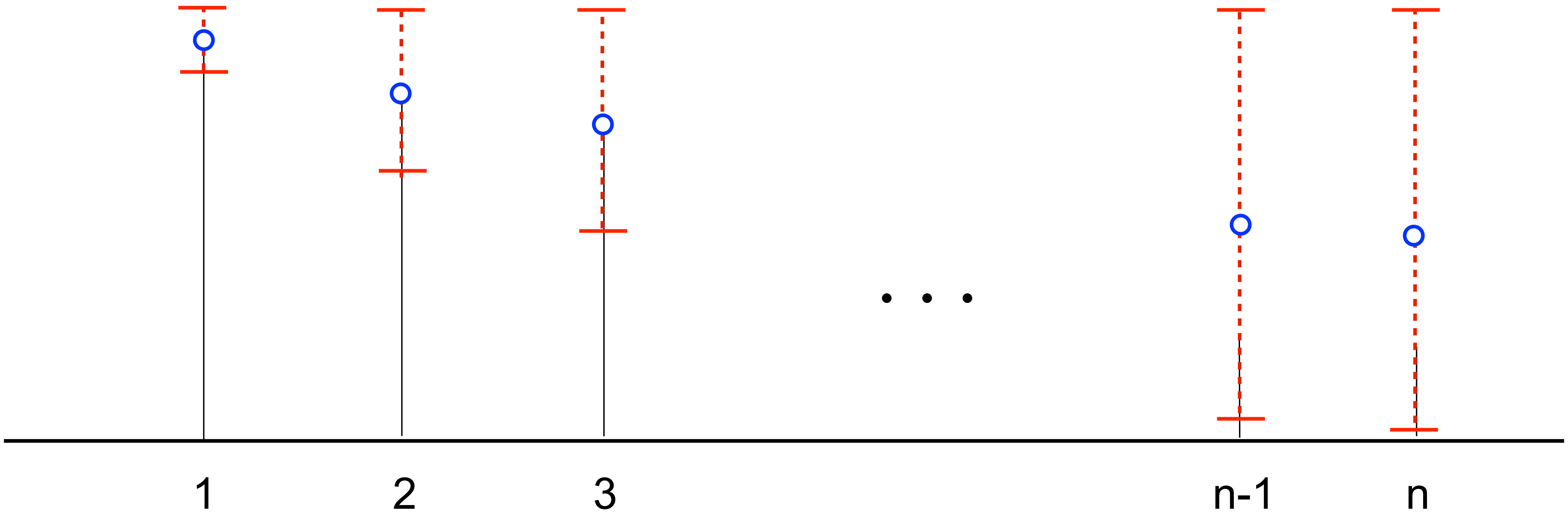
$$\Delta\, t = \sqrt{2t \log\log t} \quad \Rightarrow \quad t \approx \Delta_2^{-2} \log\log \Delta_2^{-2}$$

this suggests $T = O\left(\sum_{i\geq 2} \Delta_i^{-2} \log\left(\frac{\log \Delta_i^{-2}}{\delta}\right)\right)$

LIL UCB Algorithm

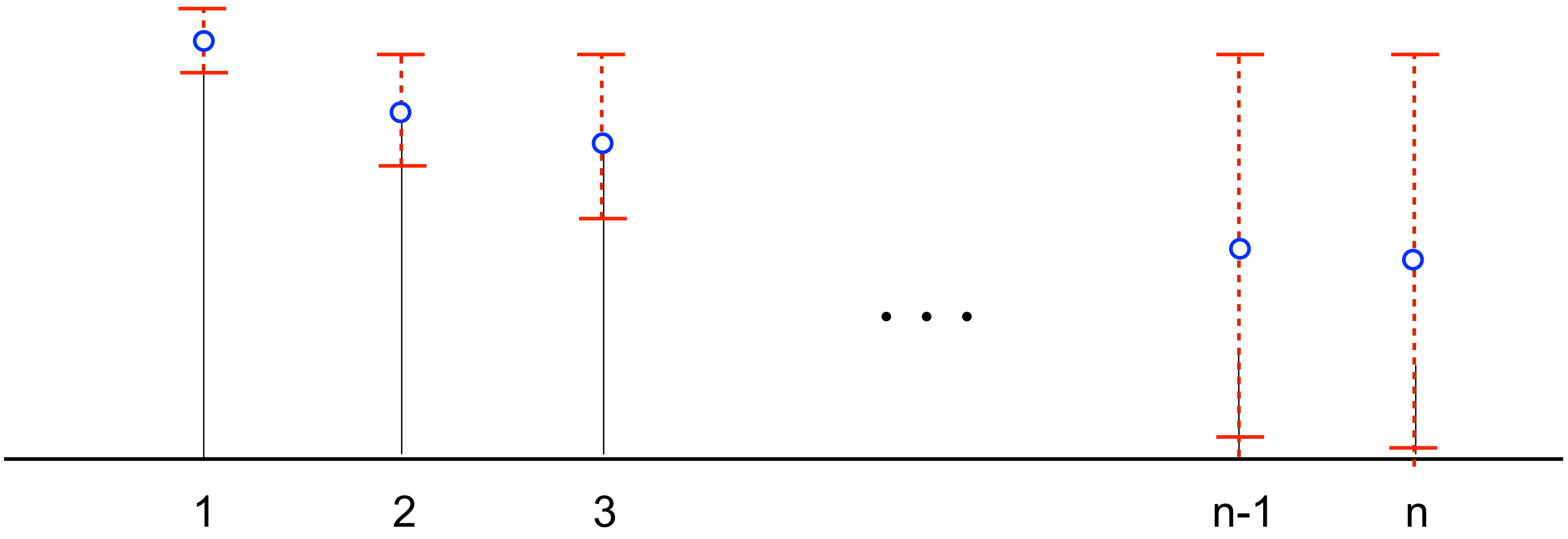
$$\widehat{\mu}_{i,t} - \sqrt{\frac{c}{t}\log\left(\frac{\log t}{\delta}\right)} \;\le\; \mu_i \;\le\; \widehat{\mu}_{i,t} + \sqrt{\frac{c}{t}\log\left(\frac{\log t}{\delta}\right)}$$

sample arm with largest LIL upper confidence bound



LIL UCB Algorithm

... eventually, algorithm will stop sampling suboptimal arms



key steps in analysis are to show

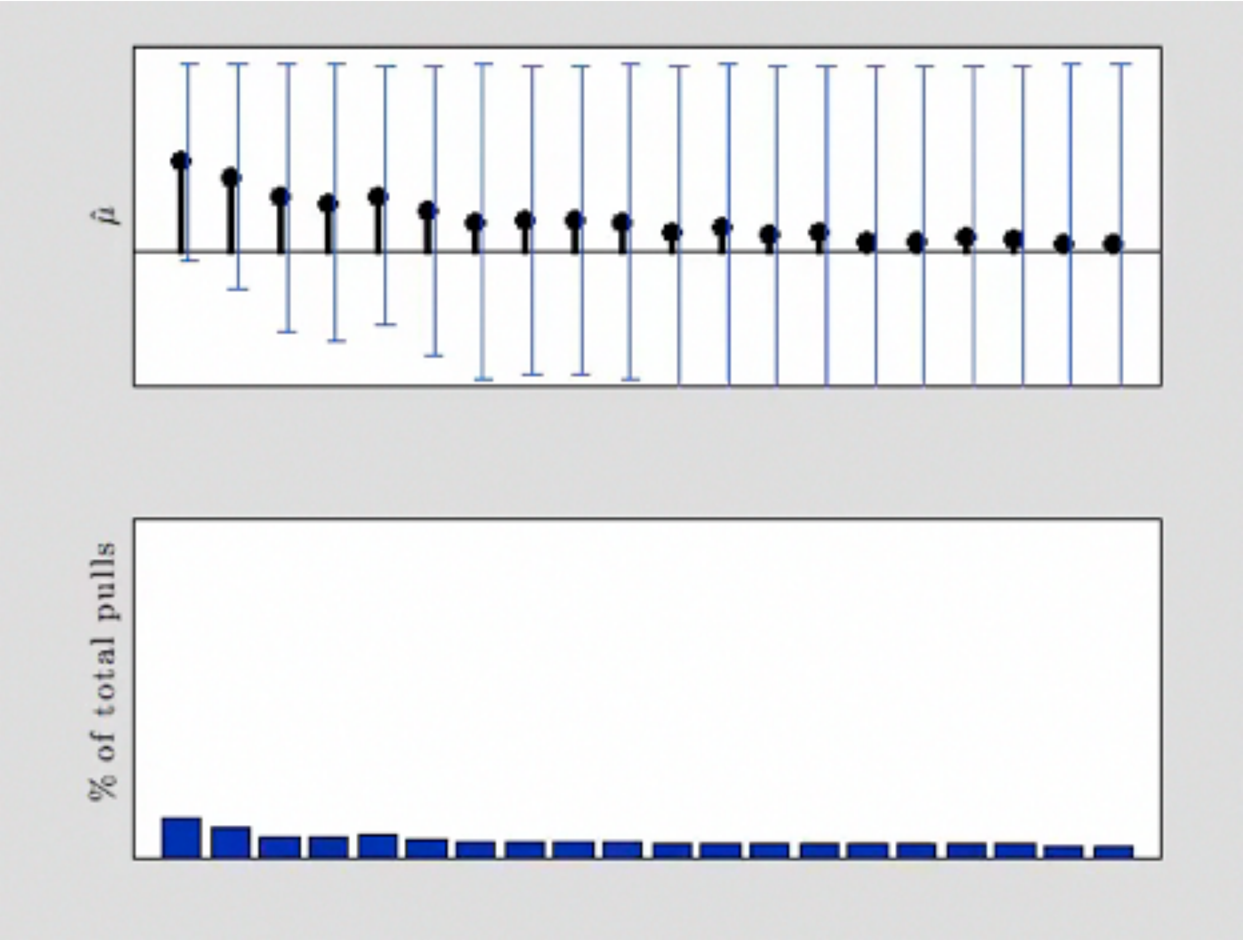
1. suboptimal arms sampled finitely many times

$$\sum_{i\geq 2} t_i \;\leq\; c_0 \sum_{i\geq 2} \Delta_i^{-2} \log\left(\frac{\log \Delta_i^{-2}}{\delta}\right)$$

2. no suboptimal arm sampled more than all others

$$t_i \;\leq\; c_1 \sum_{j\neq i} t_j + 1 \;,\; \forall\, i \geq 2$$

lil' UCB



Theorem 1 *Assume arms are sub-Gaussian. For any* $\delta \leq 0.10$*, there exist (small) universal constants* $c_0, c_1 > 0$ *such that with probability at least* $1 - c_0\delta$ *the lil' UCB algorithm stops after at most*

$$c_1 \sum_{i=2}^{n} \Delta_i^{-2} \log(\log(\Delta_i^{-2})/\delta)$$

samples and outputs the optimal arm.

Jamieson et al (2014)

Dueling Bandits

Rather than collecting ratings, collect binary comparisons between pairs of items; e.g., Do you prefer Beer A or Beer B ?

$$p_{ij} \;=\; \mathbb{P}(\text{arm } i \succ \text{arm } j), \text{ probability person prefers } i \text{ to } j$$

$$\text{samples } x_{ij} \sim \text{Bernoulli}(p_{ij})$$

Yue et al (2012)

Many criteria for how to decide which item is most prefered (e.g., Condorcet, **Borda**, etc.)

$$\text{Borda score: } \mu_i \;:=\; \tfrac{1}{n-1}\textstyle\sum_{j\neq i} p_{ij}$$

Simulate sample from arm i:

$$x_i = x_{iJ}, \;\text{ where } J \sim \text{uniform over } [n]/i$$

from here we can apply all the algorithms for the usual best arm problem

$$P_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \end{bmatrix}$$

$$P_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{3}{4}+\epsilon/n & \cdots & \frac{3}{4}+\epsilon/n \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon/n & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{4}-\epsilon/n & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \end{bmatrix}$$

Assume p_{ij} are known up to permutation of the arms

P_1 and P_2 have roughly the same Borda scores, but very different sample complexities:

$$T_1 = O\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$$

$$T_2 \gtrsim \frac{n^2}{\epsilon^2}\log\frac{1}{\delta}$$

$$P_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \end{bmatrix}$$

$$T_1 = O\left(\frac{n}{\epsilon^2} \log \frac{n}{\delta}\right)$$

1. Duel each arm with $O(\log \frac{n}{\delta})$ others, chosen uniformly at random

2. Duel arms 1 and 2 against each other arm $O\left(\frac{1}{\epsilon^2} \log \frac{n}{\delta}\right)$ times

$$P_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{3}{4}+\epsilon/n & \cdots & \frac{3}{4}+\epsilon/n \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon/n & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{4}-\epsilon/n & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \end{bmatrix} \qquad T_2 \gtrsim \frac{n^2}{\epsilon^2} \log \frac{1}{\delta}$$

1. Duel each arm with $O(\log \frac{n}{\delta})$ others, chosen uniformly at random

2. Duel arms 1 and 2 against **any** other arm $O\left(\frac{n^2}{\epsilon^2} \log \frac{n}{\delta}\right)$ times

$$P_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \end{bmatrix}$$

Assume p_{ij} are known up to permutation of the arms

P_1 and P_2 have roughly the same Borda scores, but very different sample complexities:

$$P_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{3}{4}+\epsilon/n & \cdots & \frac{3}{4}+\epsilon/n \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon/n & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{4}-\epsilon/n & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \end{bmatrix}$$

$$T_1 = O\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$$

$$T_2 \gtrsim \frac{n^2}{\epsilon^2}\log\frac{1}{\delta}$$

sparsity helps !

Bounds for Borda Dueling Bandits

$$\textbf{Borda score:}\ \mu_i \ := \ \tfrac{1}{n-1} \textstyle\sum_{j \neq i} p_{ij}$$

$$\textbf{Borda gaps:}\ \Delta_i = \mu_i - \mu_1 \ , \ i \geq 2$$

general upper bound on sample complexity: $T = O\left(\sum_{i \geq 2} \Delta_i^{-2} \log\left(\frac{\log \Delta_i^{-2}}{\delta} \right) \right)$

... but maybe it is possible to automatically adapt to sparsity to achieve better results

Consider class problems $\mathcal{P} := \{P : \frac{3}{8} \leq p_{ij} \leq \frac{5}{8}\ \forall\ ij\}$ and class $\mathcal{A}$ of procedures that are guaranteed to find Borda winner with probability at least $1-\delta\ \forall\ P \in \mathcal{P}$.

Then for every $P \in \mathcal{P}$ and every procedure in $\mathcal{A}$, the expected number of samples satisfies

$$\mathbb{E}_P[T] \ \geq \ C \log\left(\frac{1}{2\delta}\right) \sum_{i \geq 2} \Delta_i^{-2}$$

using techniques from Kaufmann, Cappe, & Garivier (2014)

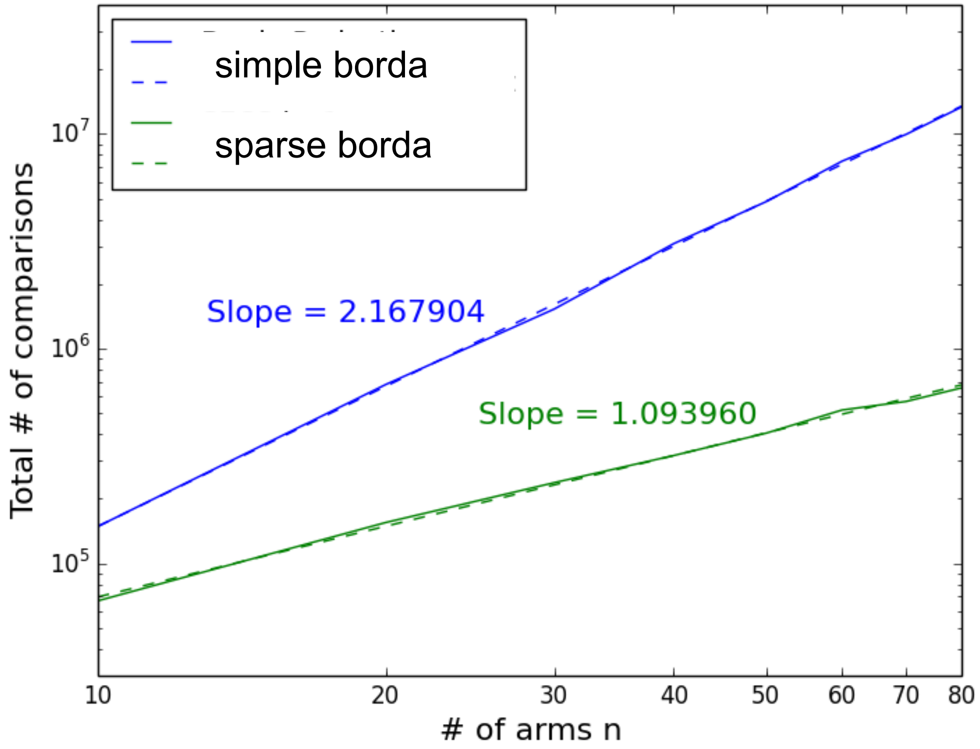
=> impossible to agnostically exploit sparsity for much, if any, gain

Sparse Borda Algorithm

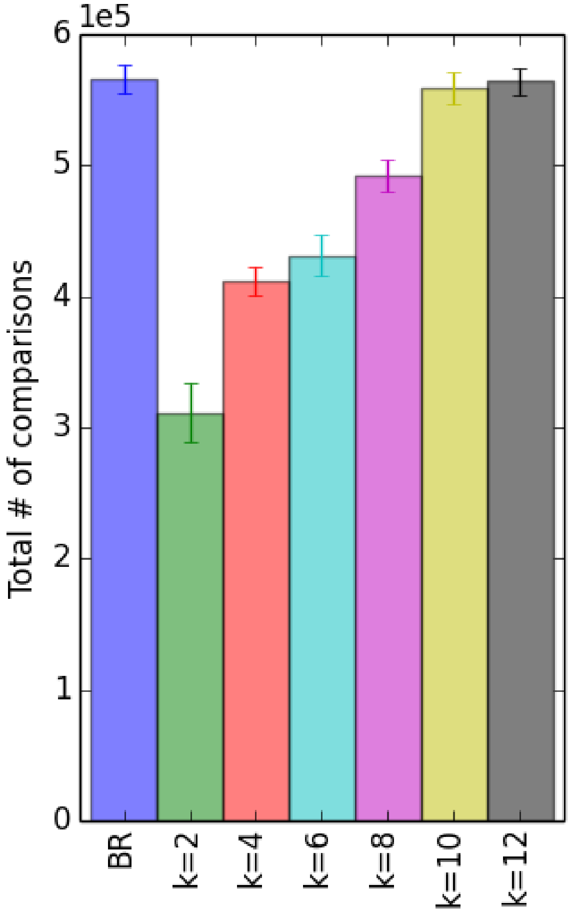
Assumption: best arm is differentiated from any suboptimal arm by a small subset (of size at most k) of all possible duels

Algorithmic idea: Successive elimination of arms and duels

Results: provably improves on sample complexity of simple Borda reduction



$$P_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} \end{bmatrix}$$



(a) MSLR-WEB10k

Jamieson, Katariya and others (2012)

Thanks!

Kevin Jamieson

Matt Malloy

Sumeet Katariya

Sebastien Bubeck