Sparse Dueling Bandits

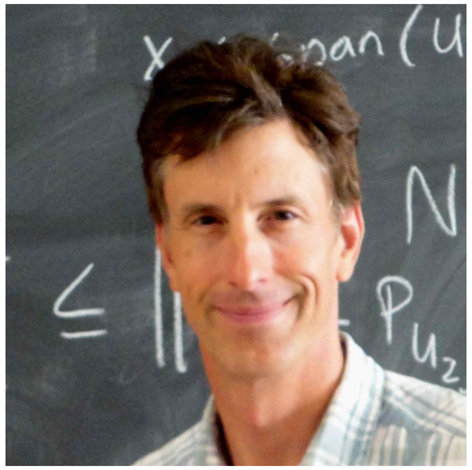
Sumeet Katariya

Electrical and Computer Engineering
University of Wisconsin-Madison

May 5, 2015

Kevin Jamieson

Atul Deshpande



Robert Nowak

The Multi-Armed Bandit Problem

Formulation

- n stochastic arms (items e.g. tshirts)
- Unknown (subgaussian) reward distribution with means $\mu_1 > \mu_2 \geq \mu_3 \cdots \geq \mu_n$
- $X_{i,t} \sim P_{\mu_i} \quad i = 1, 2, \cdots, n \quad t = 1, 2, \cdots$

Best arm Identification Problem

Given probability of error δ, find an algorithm that identifies the best arm using as few samples as possible while satisfying

$$\sup_{\mu_1 > \mu_2 \geq \cdots \geq \mu_n} \mathbb{P}(\hat{i} \neq 1) \leq \delta$$

The Multi-Armed Bandit Problem

Formulation

- n stochastic arms (items e.g. tshirts)
- Unknown (subgaussian) reward distribution with means $\mu_1 > \mu_2 \geq \mu_3 \cdots \geq \mu_n$
- $X_{i,t} \sim P_{\mu_i} \quad i = 1, 2, \cdots, n \quad t = 1, 2, \cdots$

Best arm Identification Problem

Given probability of error δ, find an algorithm that identifies the best arm using as few samples as possible while satisfying

$$\sup_{\mu_1 > \mu_2 \geq \cdots \geq \mu_n} \mathbb{P}(\hat{i} \neq 1) \leq \delta$$

Dueling Bandits Problem

Formulation (Yue et al 2012)

- Rather than observing rewards, observe binary comparisons between arms e.g: Is tshirt i better than tshirt j?
$$p_{ij} = \mathbb{P}(\text{arm } i \succ \text{arm } j)$$
- Binary samples $X_{ij,t} \sim \text{Bernoulli}(p_{ij})$

Best Arm Problem

Many criteria for how to decide the best arm (e.g. Condorcet, Copeland, Borda, etc)

Dueling Bandits Problem

Formulation (Yue et al 2012)

- Rather than observing rewards, observe binary comparisons between arms e.g: Is tshirt *i* better than tshirt *j*?

$$p_{ij} = \mathbb{P}(\text{arm } i \succ \text{arm } j)$$

- Binary samples $X_{ij,t} \sim$Bernoulli(p_{ij})

Best Arm Problem

Many criteria for how to decide the best arm (e.g. Condorcet, Copeland, Borda, etc)

Best Arm Criteria

$$P = \begin{array}{c|cccc|c} & 1 & 2 & 3 & 4 & \text{Borda Score} \\ 1 & - & 0.51 & 0.51 & 0.51 & 0.51 \\ 2 & 0.49 & - & 0.99 & 0.99 & 0.82 \\ 3 & 0.49 & 0.01 & - & 0.6 & 0.36 \\ 4 & 0.49 & 0.01 & 0.4 & - & 0.3 \end{array}$$

$\left(\frac{0.49+0.99+0.99}{3}\right)$

- Condorcet winner: Arm that beats every other arm.
- Borda winner: Arm with the highest Borda score.

  Borda score of arm i: Probability of arm i beating a random other arm $J \sim$Uniform$([n] \setminus i)$.

$$\text{Borda score } \mu_i = \frac{1}{n-1} \sum_{j \neq i} p_{ij}$$

Best Arm Criteria

$$P = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} - & 0.51 & 0.51 & 0.51 \\ 0.49 & - & 0.99 & 0.99 \\ 0.49 & 0.01 & - & 0.6 \\ 0.49 & 0.01 & 0.4 & - \end{pmatrix} \end{array} \quad \begin{array}{c} \text{Borda Score} \\ 0.51 \\ 0.82 \\ 0.36 \\ 0.3 \end{array} \quad \left(\frac{0.49+0.99+0.99}{3}\right)$$

- Condorcet winner: Arm that beats every other arm.
- Borda winner: Arm with the highest Borda score.

  Borda score of arm i: Probability of arm i beating a random other arm $J \sim$Uniform($[n] \setminus i$).

$$\text{Borda score } \mu_i = \frac{1}{n-1} \sum_{j \neq i} p_{ij}$$

Best Arm Criteria

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} - & 0.51 & 0.51 & 0.51 \\ 0.49 & - & 0.99 & 0.99 \\ 0.49 & 0.01 & - & 0.6 \\ 0.49 & 0.01 & 0.4 & - \end{pmatrix} \end{array} \begin{array}{c} \text{Borda Score} \\ 0.51 \\ 0.82 \\ 0.36 \\ 0.3 \end{array} \quad \left(\frac{0.49+0.99+0.99}{3}\right)$$

- Condorcet winner: Arm that beats every other arm.
- Borda winner: Arm with the highest Borda score.

Borda score of arm i: Probability of arm i beating a random other arm $J \sim$Uniform$([n] \setminus i)$.

$$\text{Borda score } \mu_i = \frac{1}{n-1} \sum_{j \neq i} p_{ij}$$

Best Arm Criteria

$$P = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} - & 0.51 & 0.51 & 0.51 \\ \color{red}{0.49} & - & \color{red}{0.99} & \color{red}{0.99} \\ 0.49 & 0.01 & - & 0.6 \\ 0.49 & 0.01 & 0.4 & - \end{pmatrix} \end{array} \quad \begin{array}{c} \text{Borda Score} \\ 0.51 \\ \color{red}{0.82} \\ 0.36 \\ 0.3 \end{array} \quad \left(\frac{0.49+0.99+0.99}{3}\right)$$

- Condorcet winner: Arm that beats every other arm.
- Borda winner: Arm with the highest Borda score.

  Borda score of arm i: Probability of arm i beating a random other arm $J \sim$Uniform($[n] \setminus i$).

$$\text{Borda score } \mu_i = \frac{1}{n-1} \sum_{j \neq i} p_{ij}$$

Best Arm Criteria

$$
P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{pmatrix}
- & 0.51 & 0.51 & 0.51 \\
\color{red}{0.49} & - & \color{red}{0.99} & \color{red}{0.99} \\
0.49 & 0.01 & - & 0.6 \\
0.49 & 0.01 & 0.4 & -
\end{pmatrix}
\end{array}
\begin{array}{c} \text{Borda Score} \\ 0.51 \\ \color{red}{0.82} \\ 0.36 \\ 0.3 \end{array}
\quad \left(\frac{0.49+0.99+0.99}{3}\right)
$$

- Condorcet winner: Arm that beats every other arm.
- Borda winner: Arm with the highest Borda score.

  Borda score of arm i: Probability of arm i beating a random other arm $J \sim$Uniform$([n] \setminus i)$.

$$\text{Borda score } \mu_i = \frac{1}{n-1} \sum_{j \neq i} p_{ij}$$

Why has Condorcet received more attention?

- In a duel between Condorcet and Borda arm, the Condorcet arm is preferred.
- The problem of finding the best Borda arm can be reduced to a multi-armed bandits(MAB) problem.
  Take your favorite best-arm MAB algorithm, simulate sample from arm i:

  $$X_{it}(\mathrm{MAB}) := X_{iJt}(\mathrm{Dueling})$$

  where $J \sim$ uniform over $[n] \setminus i$.
  Referred to as Borda Reduction.

Why has Condorcet received more attention?

- In a duel between Condorcet and Borda arm, the Condorcet arm is preferred.
- The problem of finding the best Borda arm can be reduced to a multi-armed bandits(MAB) problem.
  Take your favorite best-arm MAB algorithm, simulate sample from arm i:

$$X_{it}(\text{MAB}) := X_{iJt}(\text{Dueling})$$

  where $J \sim$ uniform over $[n] \setminus i$.
  Referred to as Borda Reduction.

Why has Condorcet received more attention?

- In a duel between Condorcet and Borda arm, the Condorcet arm is preferred.
- The problem of finding the best Borda arm can be reduced to a multi-armed bandits(MAB) problem.
  Take your favorite best-arm MAB algorithm, simulate sample from arm i:

$$X_{it}(\text{MAB}) := X_{iJt}(\text{Dueling})$$

  where $J \sim$ uniform over $[n] \setminus i$.
  Referred to as Borda Reduction.

Why has Condorcet received more attention?

- In a duel between Condorcet and Borda arm, the Condorcet arm is preferred.
- The problem of finding the best Borda arm can be reduced to a multi-armed bandits(MAB) problem.
  Take your favorite best-arm MAB algorithm, simulate sample from arm i:

  $$X_{it}(\text{MAB}) := X_{iJt}(\text{Dueling})$$

  where $J \sim$ uniform over $[n] \setminus i$.
  Referred to as Borda Reduction.

Borda deserves more attention

- There may not be a Condorcet winner - a row with all entries $> \frac{1}{2}$.
- Cases where Borda winner may be more appropriate

$$
\begin{array}{c|cccc|cl}
 & 1 & 2 & 3 & 4 & \mu_i & \\
1 & - & 0.51 & 0.51 & 0.51 & 0.51 & \text{(Condorcet)} \\
2 & 0.49 & - & 0.99 & 0.99 & 0.82 & \text{(Borda)} \\
3 & 0.49 & 0.01 & - & 0.6 & 0.36 & \\
4 & 0.49 & 0.01 & 0.4 & - & 0.3 &
\end{array}
$$

- Sample complexity for Condorcet $\mathcal{O}(n^2)$.
- Better algorithms than Borda reduction?

Borda deserves more attention

- There may not be a Condorcet winner - a row with all entries $> \frac{1}{2}$.
- Cases where Borda winner may be more appropriate

$$
\begin{array}{c|cccc|cl}
 & 1 & 2 & 3 & 4 & \mu_i & \\
\hline
1 & - & 0.51 & 0.51 & 0.51 & 0.51 & \text{(Condorcet)} \\
2 & 0.49 & - & 0.99 & 0.99 & 0.82 & \text{(Borda)} \\
3 & 0.49 & 0.01 & - & 0.6 & 0.36 & \\
4 & 0.49 & 0.01 & 0.4 & - & 0.3 &
\end{array}
$$

- Sample complexity for Condorcet $\mathcal{O}(n^2)$.
- Better algorithms than Borda reduction?

Borda deserves more attention

- There may not be a Condorcet winner - a row with all entries $> \frac{1}{2}$.
- Cases where Borda winner may be more appropriate

$$
\begin{array}{c|c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} & \mu_i & \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} - & 0.51 & 0.51 & 0.51 \\ 0.49 & - & 0.99 & 0.99 \\ 0.49 & 0.01 & - & 0.6 \\ 0.49 & 0.01 & 0.4 & - \end{pmatrix} & \begin{array}{c} \color{red}{0.51} \\ \color{red}{0.82} \\ 0.36 \\ 0.3 \end{array} & \begin{array}{c} \text{(Condorcet)} \\ \text{(Borda)} \\ \\ \\ \end{array}
\end{array}
$$

- Sample complexity for Condorcet $\mathcal{O}(n^2)$.
- Better algorithms than Borda reduction?

Borda deserves more attention

- There may not be a Condorcet winner - a row with all entries $> \frac{1}{2}$.
- Cases where Borda winner may be more appropriate

$$
\begin{array}{c c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} & \mu_i \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} &
\begin{pmatrix} - & 0.51 & 0.51 & 0.51 \\ 0.49 & - & 0.99 & 0.99 \\ 0.49 & 0.01 & - & 0.6 \\ 0.49 & 0.01 & 0.4 & - \end{pmatrix} &
\begin{array}{c} \color{red}{0.51} \\ \color{red}{0.82} \\ 0.36 \\ 0.3 \end{array}
\end{array}
\begin{array}{c} \text{(Condorcet)} \\ \text{(Borda)} \\ \\ \\ \end{array}
$$

- Sample complexity for Condorcet $\mathcal{O}(n^2)$.
- Better algorithms than Borda reduction?

Borda deserves more attention

- There may not be a Condorcet winner - a row with all entries $> \frac{1}{2}$.
- Cases where Borda winner may be more appropriate

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} - & 0.51 & 0.51 & 0.51 \\ 0.49 & - & 0.99 & 0.99 \\ 0.49 & 0.01 & - & 0.6 \\ 0.49 & 0.01 & 0.4 & - \end{pmatrix} \end{array} \begin{array}{cc} \mu_i & \\ \color{red}{0.51} & \text{(Condorcet)} \\ \color{red}{0.82} & \text{(Borda)} \\ 0.36 & \\ 0.3 & \end{array}$$

- Sample complexity for Condorcet $\mathcal{O}(n^2)$.
- Better algorithms than Borda reduction?

Is Borda Reduction the best?

$$P_1 = \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{matrix} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{matrix}$$

If P_1 were known upto a permutation of the indices, $\tilde{T}_1 = \mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$

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

If P_2 were known upto a permutation of the indices, $\tilde{T}_2 = \mathcal{O}\left(\frac{n^2}{\epsilon^2}\log\frac{1}{\delta}\right)$

The sample complexity of Borda reduction for both P_1 and P_2 is the same, because they have the same Borda scores.

Is Borda Reduction the best?

$$P_1 = \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{matrix} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{matrix}$$

If P_1 were known upto a permutation of the indices, $\tilde{T}_1 = \mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$

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

If P_2 were known upto a permutation of the indices, $\tilde{T}_2 = \mathcal{O}\left(\frac{n^2}{\epsilon^2}\log\frac{1}{\delta}\right)$

The sample complexity of Borda reduction for both P_1 and P_2 is the same, because they have the same Borda scores.

Is Borda Reduction the best?

$$P_1 = \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{matrix} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{matrix}$$

If P_1 were known upto a permutation of the indices, $\tilde{T}_1 = \mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$

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

If P_2 were known upto a permutation of the indices, $\tilde{T}_2 = \mathcal{O}\left(\frac{n^2}{\epsilon^2}\log\frac{1}{\delta}\right)$

The sample complexity of Borda reduction for both P_1 and P_2 is the same, because they have the same Borda scores.

Pseudo-Complexity of P_1

$$P_1 = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & n \end{array} \\ \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4}{\color{red}+\epsilon} & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \end{array} \begin{array}{cc} \mu_i & \Delta_i = \mu_1 - \mu_i \\ \frac{3}{4}+\frac{\epsilon}{n} & \\ \frac{3}{4} & \frac{\epsilon}{n} \\ \frac{1}{2} & \frac{1}{4} \\ \vdots & \vdots \\ \frac{1}{2} & \frac{1}{4} \end{array}$$

Imagine we know P_1 upto a permutation of the indices.

- Duel each arm with $\mathcal{O}\left(\log \frac{n}{\delta}\right)$ others. Complexity $\mathcal{O}\left(n \log \frac{n}{\delta}\right)$.
- Duel top 2 arms with each of the remaining $n-2$ arms $\mathcal{O}\left(\frac{1}{\epsilon^2} \log \frac{n}{\delta}\right)$ times. Complexity $\mathcal{O}\left(\frac{n}{\epsilon^2} \log \frac{n}{\delta}\right)$.

Pseudo-Complexity of P_1

$$\begin{array}{c} \\ P_1 = \end{array} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & n \\ \left(\begin{array}{ccccc} - & \frac{1}{2} & \frac{3}{4}{\color{red}+\epsilon} & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{array}\right) \end{array} \begin{array}{c} \mu_i \\ \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{array} \quad \begin{array}{c} \Delta_i = \mu_1 - \mu_i \\ \\ \frac{\epsilon}{n} \\ {\color{red}\frac{1}{4}} \\ {\color{red}\vdots} \\ {\color{red}\frac{1}{4}} \end{array}$$

Imagine we know P_1 upto a permutation of the indices.

- Duel each arm with $\mathcal{O}\left(\log \frac{n}{\delta}\right)$ others. Complexity $\mathcal{O}\left(n \log \frac{n}{\delta}\right)$.
- Duel top 2 arms with each of the remaining $n-2$ arms $\mathcal{O}\left(\frac{1}{\epsilon^2} \log \frac{n}{\delta}\right)$ times. Complexity $\mathcal{O}\left(\frac{n}{\epsilon^2} \log \frac{n}{\delta}\right)$.

Pseudo-Complexity of P_1

$$
\begin{array}{c} \\ P_1 = \end{array}
\begin{array}{ccccc|cc}
1 & 2 & 3 & \cdots & n & \mu_i & \Delta_i = \mu_1 - \mu_i \\
\end{array}
$$

$$
P_1 = \begin{pmatrix}
- & \frac{1}{2} & \frac{3}{4} + \epsilon & \cdots & \frac{3}{4} \\
\frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\
\frac{1}{4} - \epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & -
\end{pmatrix}
\begin{array}{cc}
\frac{3}{4} + \frac{\epsilon}{n} & \\
\frac{3}{4} & \frac{\epsilon}{n} \\
\frac{1}{2} & \frac{1}{4} \\
\vdots & \vdots \\
\frac{1}{2} & \frac{1}{4}
\end{array}
$$

Imagine we know P_1 upto a permutation of the indices.

- Duel each arm with $\mathcal{O}\left(\log \frac{n}{\delta}\right)$ others. Complexity $\mathcal{O}\left(n \log \frac{n}{\delta}\right)$.
- Duel top 2 arms with each of the remaining $n-2$ arms $\mathcal{O}\left(\frac{1}{\epsilon^2}\log \frac{n}{\delta}\right)$ times. Complexity $\mathcal{O}\left(\frac{n}{\epsilon^2}\log \frac{n}{\delta}\right)$.

Pseudo-Complexity of P_1

$$\begin{array}{cccc} & \begin{array}{ccccc} 1 & 2 & 3 & \cdots & n \end{array} & \mu_i & \Delta_i = \mu_1 - \mu_i \\ P_1 = & \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4} + \epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4} - \epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} & \begin{array}{c} \frac{3}{4} + \frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{array} & \begin{array}{c} \\ \frac{\epsilon}{n} \\ \frac{1}{4} \\ \vdots \\ \frac{1}{4} \end{array} \end{array}$$

Imagine we know P_1 upto a permutation of the indices.

- Duel each arm with $\mathcal{O}\left(\log \frac{n}{\delta}\right)$ others. Complexity $\mathcal{O}\left(n \log \frac{n}{\delta}\right)$.
- Duel top 2 arms with each of the remaining $n - 2$ arms $\mathcal{O}\left(\frac{1}{\epsilon^2} \log \frac{n}{\delta}\right)$ times. Complexity $\mathcal{O}\left(\frac{n}{\epsilon^2} \log \frac{n}{\delta}\right)$.

Pseudo-complexity of P_2

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

Complexity $\Omega\left(\frac{n^2}{\epsilon^2}\log\frac{1}{\delta}\right)$

$$P_1 = \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4} + \epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4} - \epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{array}{c} \frac{3}{4} + \frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{array} \qquad \tilde{T}_1 = \mathcal{O}\left(\frac{n}{\epsilon^2} \log \frac{n}{\delta}\right)$$

$$P_2 = \begin{pmatrix} - & \frac{1}{2} + \frac{\epsilon}{n} & \frac{3}{4} + \frac{\epsilon}{n} & \cdots & \frac{3}{4} + \frac{\epsilon}{n} \\ \frac{1}{2} - \frac{\epsilon}{n} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4} - \frac{\epsilon}{n} & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} - \frac{\epsilon}{n} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{array}{c} \frac{3}{4} + \frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{array} \qquad \tilde{T}_2 = \Omega\left(\frac{n^2}{\epsilon^2} \log \frac{1}{\delta}\right)$$

Sparsity helps!
Can we adaptively learn sparsity and achieve better results?

Distribution dependent lower bound

$$\text{Borda score } = \mu_i = \frac{1}{n-1}\sum_{j\neq i} p_{ij}$$

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

Theorem 1: Consider class of 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

$$E_P[T] \geq C \log\left(\tfrac{1}{\delta}\right) \sum_{i\geq 2} \Delta_i^{-2}$$

using techniques from Kaufmann et al (2014)

Upper bound on sample complexity using Borda reduction and lilUCB

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

Jamieson et al (2014)
Karnin et al (2013)

⇒Impossible to agnostically exploit sparsity for much, if any, gain

$$P_1 = \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{array}{c} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{array} \qquad \tilde{T}_1 = \mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$$

$$P_2 = \begin{pmatrix} - & \frac{1}{2}+\frac{\epsilon}{n} & \frac{3}{4}+\frac{\epsilon}{n} & \cdots & \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{1}{2}-\frac{\epsilon}{n} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\frac{\epsilon}{n} & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4}-\frac{\epsilon}{n} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{array}{c} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{array} \qquad \tilde{T}_2 = \Omega\left(\frac{n^2}{\epsilon^2}\log\frac{1}{\delta}\right)$$

Sparsity helps!

Can we adaptively learn sparsity and achieve better results? No!

Can we do better if we assume sparsity?

$$P_1 = \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4}+\epsilon & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{matrix} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{matrix} \qquad \tilde{T}_1 = \mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$$

$$P_2 = \begin{pmatrix} - & \frac{1}{2}+\frac{\epsilon}{n} & \frac{3}{4}+\frac{\epsilon}{n} & \cdots & \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{1}{2}-\frac{\epsilon}{n} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\frac{\epsilon}{n} & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4}-\frac{\epsilon}{n} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} \begin{matrix} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{matrix} \qquad \tilde{T}_2 = \Omega\left(\frac{n^2}{\epsilon^2}\log\frac{1}{\delta}\right)$$

Sparsity helps!
Can we adaptively learn sparsity and achieve better results? No!
Can we do better if we assume sparsity?

Real World Evidence

Suppose we try to decide if arm 1 is better than arm $i > 1$ using only the k most discriminating duels between arm 1 and arm i

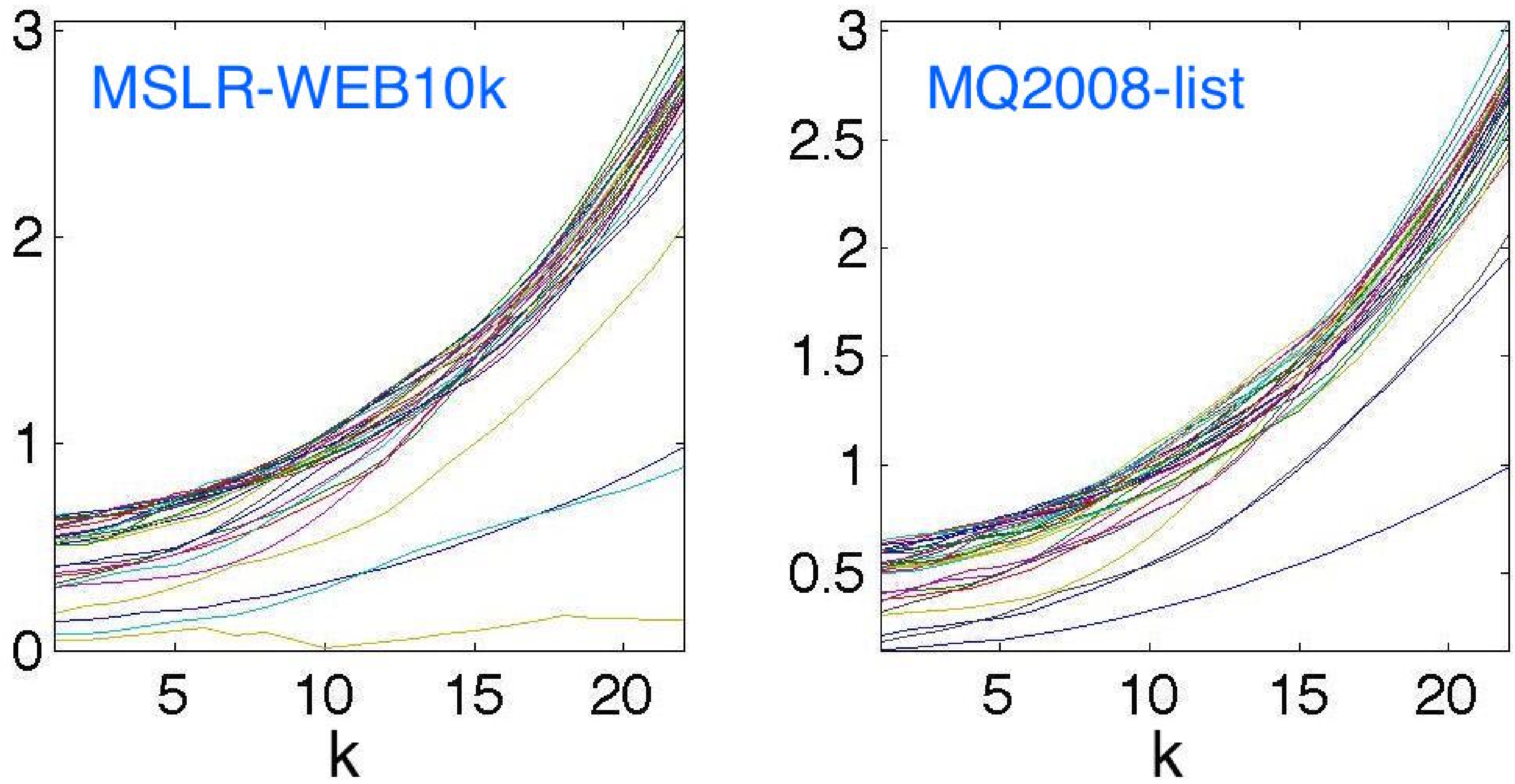
i.e. the duels with arms j that have largest values of $|p_{1,j} - p_{i,j}|$

Real World Evidence

Suppose we try to decide if arm 1 is better than arm $i > 1$ using only the k most discriminating duels between arm 1 and arm i

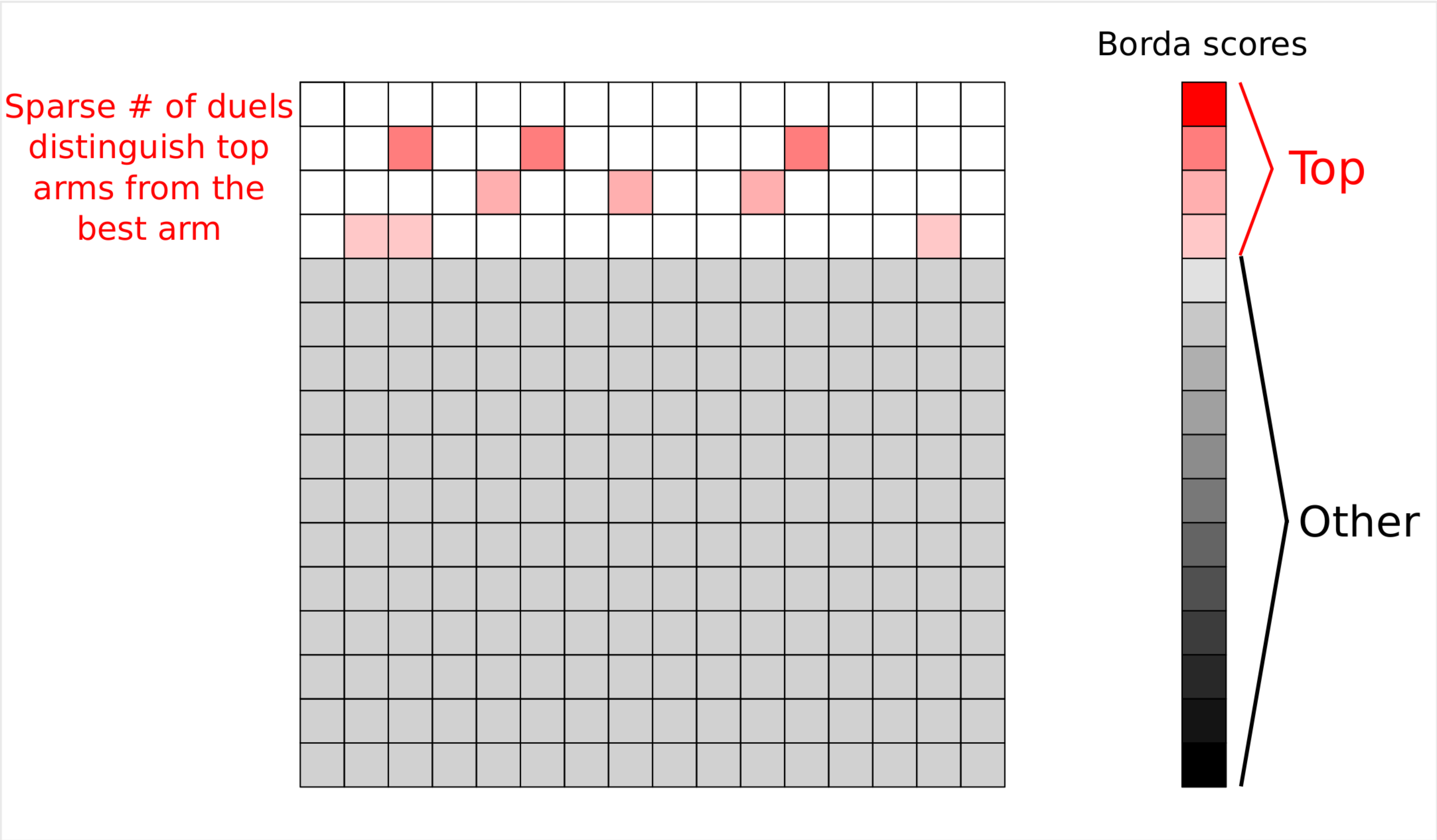
i.e. the duels with arms j that have largest values of $|p_{1,j} - p_{i,j}|$

Difference between partial Borda scores $\mu_1(k) - \mu_i(k)$ for each $i > 1$.



$\mu_1(k) - \mu_i(k) \geq 0$ based on small number (k) of most discriminating duels

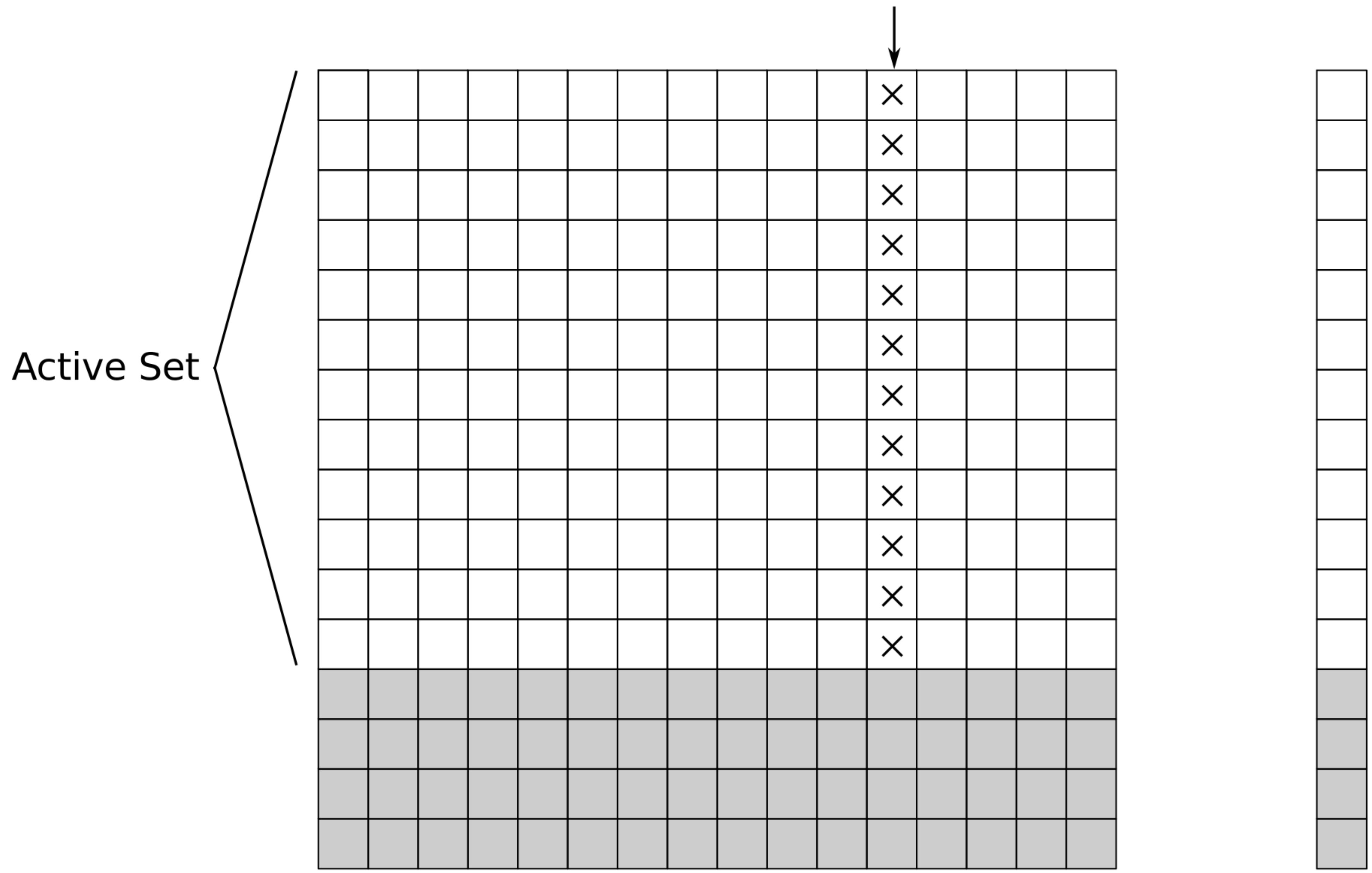
Sparsity Model



Sparse Borda Algorithm

Phase 1:

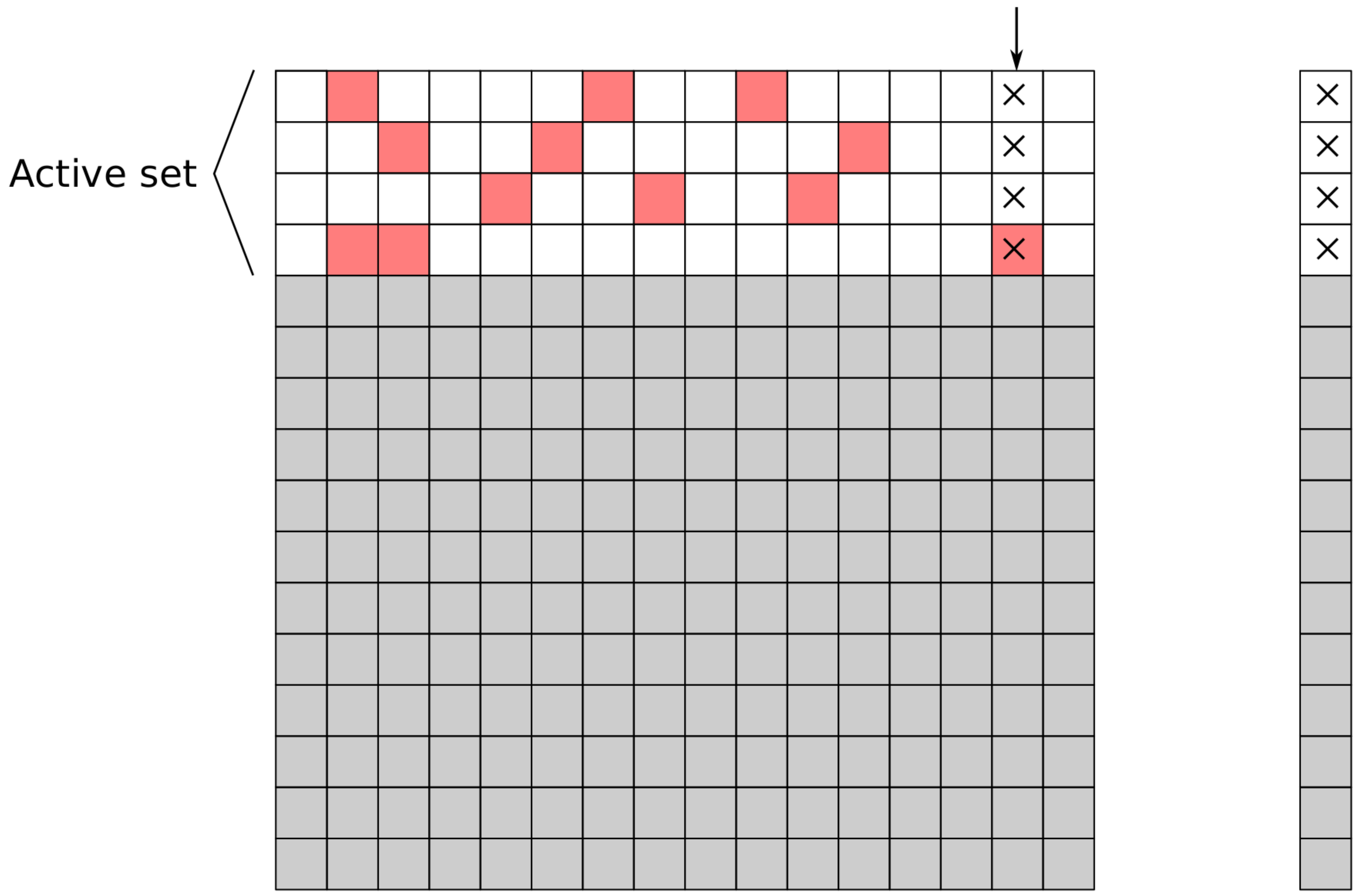
- Duel arms in “active set”, selecting duels at random.
- Estimate Borda scores
- Successively remove lowest scoring arms from “active set”



Sparse Borda Algorithm

Phase 2:

- Duel arms in “active set”, selecting duels at random.
- Successively remove lowest scoring arms from “active set” based on k-subset of Borda scores with largest gaps.



Sparse Borda Algorithm

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

Algorithm: Successive elimination of arms based on k most "discriminating" duels.

Results: Provably improves on sample complexity of simple Borda reduction.

Sample Complexity:

$$\text{Borda reduction and lilUCB:} T_{br} = \mathcal{O}\left(\sum_{i=2}^{n} \Delta_i^{-2} \log\left(\log \frac{\Delta_i^{-2}}{\delta}\right)\right)$$

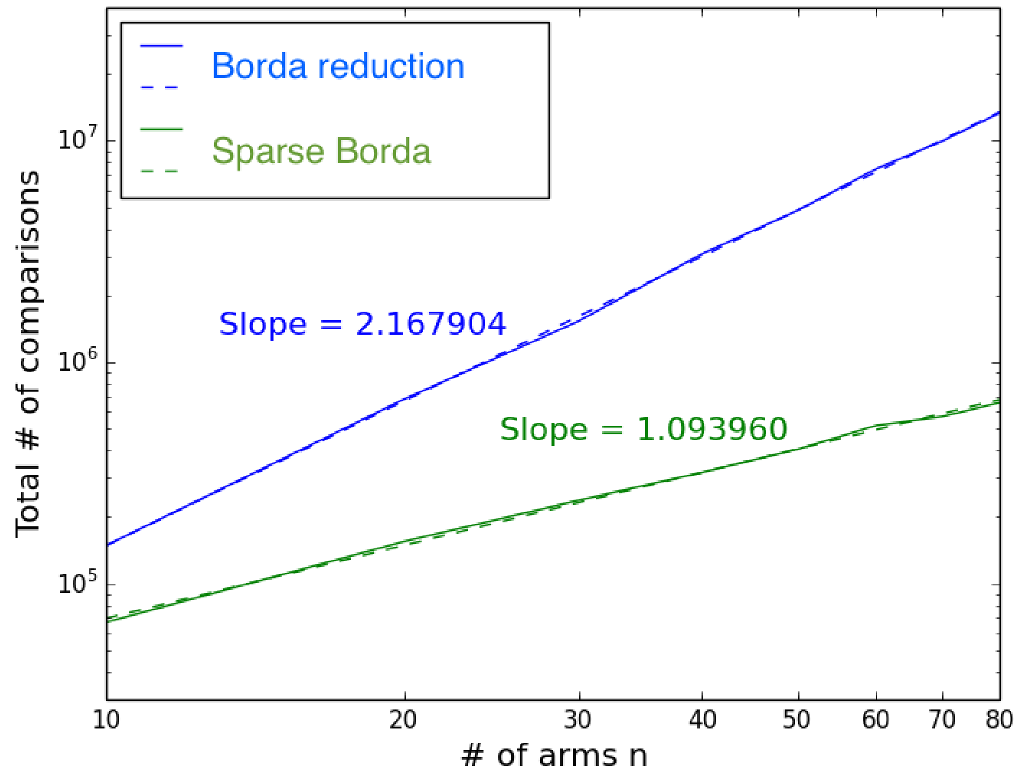
$$\text{Sparse Borda:} T_{sb} = \mathcal{O}\left(\sum_{i=2}^{n} \frac{k^2}{n} \Delta_i^{-2} \log\left(\log \frac{\Delta_i^{-2}}{\delta}\right)\right)^*$$

$$\boxed{\text{For small } k,\ T_{sb} = \mathcal{O}\left(\frac{T_{br}}{n}\right)}$$

*Actual expression more complicated, see paper.

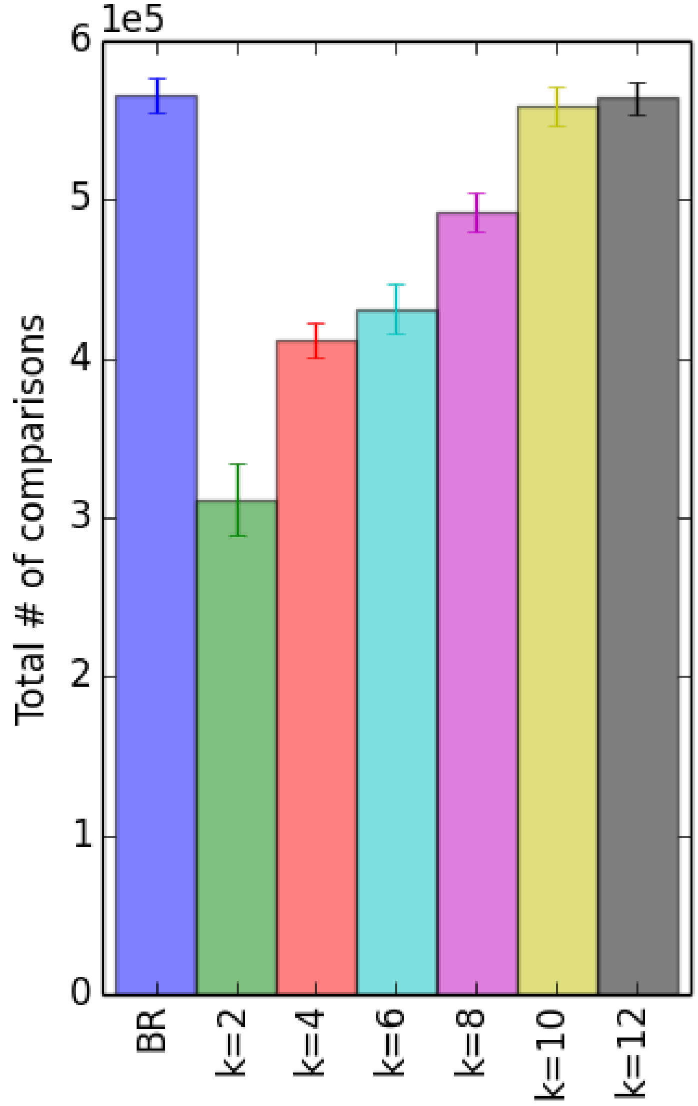
Sparse Borda Algorithm (simulated data)

$$P_1 = \begin{array}{cc} \begin{pmatrix} - & \frac{1}{2} & \frac{3}{4}{\color{red}+\epsilon} & \cdots & \frac{3}{4} \\ \frac{1}{2} & - & \frac{3}{4} & \cdots & \frac{3}{4} \\ \frac{1}{4}-\epsilon & \frac{1}{4} & - & \cdots & \frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & - \end{pmatrix} & \begin{array}{c} \frac{3}{4}+\frac{\epsilon}{n} \\ \frac{3}{4} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{array} \end{array}$$

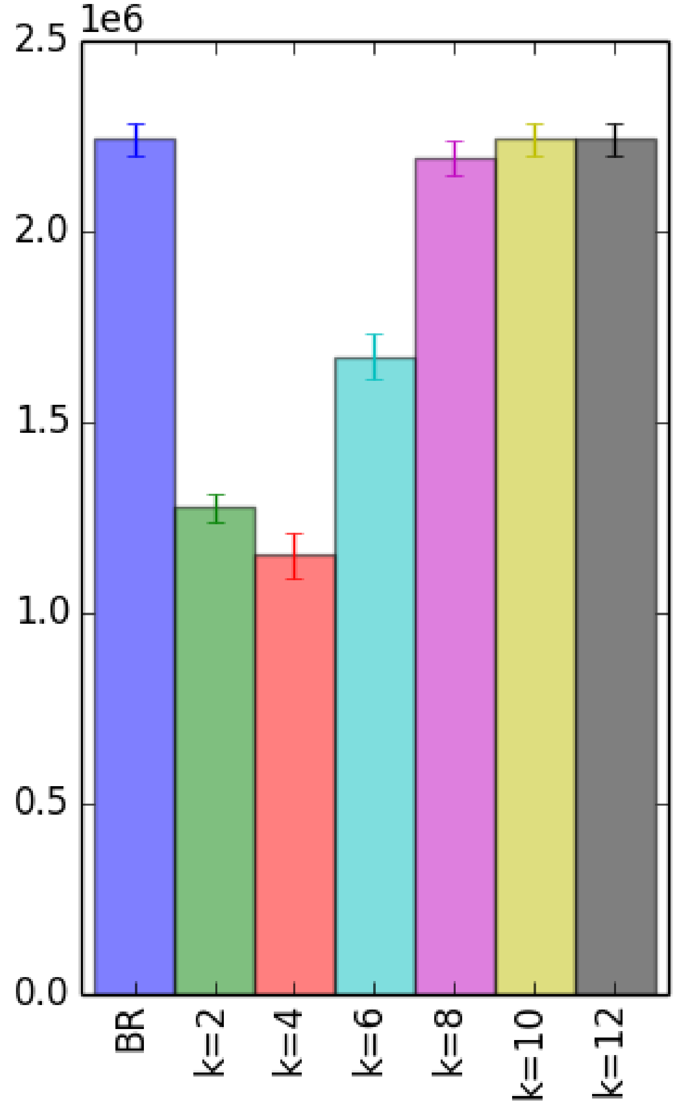


$$T_{br} = \tilde{\mathcal{O}}(n^2)$$
$$T_{sb} = \tilde{\mathcal{O}}(n)$$

Sparse Borda Algorithm (Microsoft LETOR datasets)



(d) MSLR-WEB10k



(e) MQ2008

Thank You

Statistics > Machine Learning

Sparse Dueling Bandits

Kevin Jamieson, Sumeet Katariya, Atul Deshpande, Robert Nowak

(Submitted on 31 Jan 2015)

The dueling bandit problem is a variation of the classical multi-armed bandit in which the allowable actions are noisy comparisons between pairs of arms. This paper focuses on a new approach for finding the "best" arm according to the Borda criterion using noisy comparisons. We prove that in the absence of structural assumptions, the sample complexity of this problem is proportional to the sum of the inverse squared gaps between the Borda scores of each suboptimal arm and the best arm. We explore this dependence further and consider structural constraints on the pairwise comparison matrix (a particular form of sparsity natural to this problem) that can significantly reduce the sample complexity. This motivates a new algorithm called Successive Elimination with Comparison Sparsity (SECS) that exploits sparsity to find the Borda winner using fewer samples than standard algorithms. We also evaluate the new algorithm experimentally with synthetic and real data. The results show that the sparsity model and the new algorithm can provide significant improvements over standard approaches.

Subjects: **Machine Learning (stat.ML)**; Learning (cs.LG)
Cite as: arXiv:1502.00133 **[stat.ML]**
(or arXiv:1502.00133v1 **[stat.ML]** for this version)

Submission history

From: Sumeet Katariya [view email]
[v1] Sat, 31 Jan 2015 16:18:14 GMT (540kb,D)