MergeDTS for Large Scale Condorcet Dueling Bandits

Chang Li, Ilya Markov, Maarten de Rijke and Masrour Zoghi

What are dueling bandits?

- The *K*-armed dueling bandits (Yue et al, COLT 2009):
  - *K* arms (aka actions)
  - Each time-step:
    - ➡ the algorithm chooses ***two*** arms, *l* and *r* (for "left" and "right");
    - ➡ the dueling happens between *l* and *r* with one returned as the winner.
  - **Goal**: converge to the ***optimal play*** for both *l* and *r.*

What is the optimal play?

- **Notation**: $\mathbf{P} := [P_{ij}]$ is the preference matrix with

$$P_{ij} = Pr(\text{arm } i \text{ beats arm } j)$$

- **Assumption**: there exists one arm that on average beats all the other arms: called the ***Condorcet*** winner.

$$P_{1j} > 0.5 \text{ for all } j \neq 1$$

- **Regret**: the loss of comparing non-Condorcet winner.

$$r_t = 0.5 * (P_{1l} - 0.5) + 0.5 * (P_{1r} - 0.5)$$

- **Optimal play**: only play the Condorcet winner, i.e. choose the Condorcet winner as l and r.

Related works

- **DTS** (Wu et al. NIPS 2016), etc.
  Limited to ***small scale*** set up, i.e. *K* is small

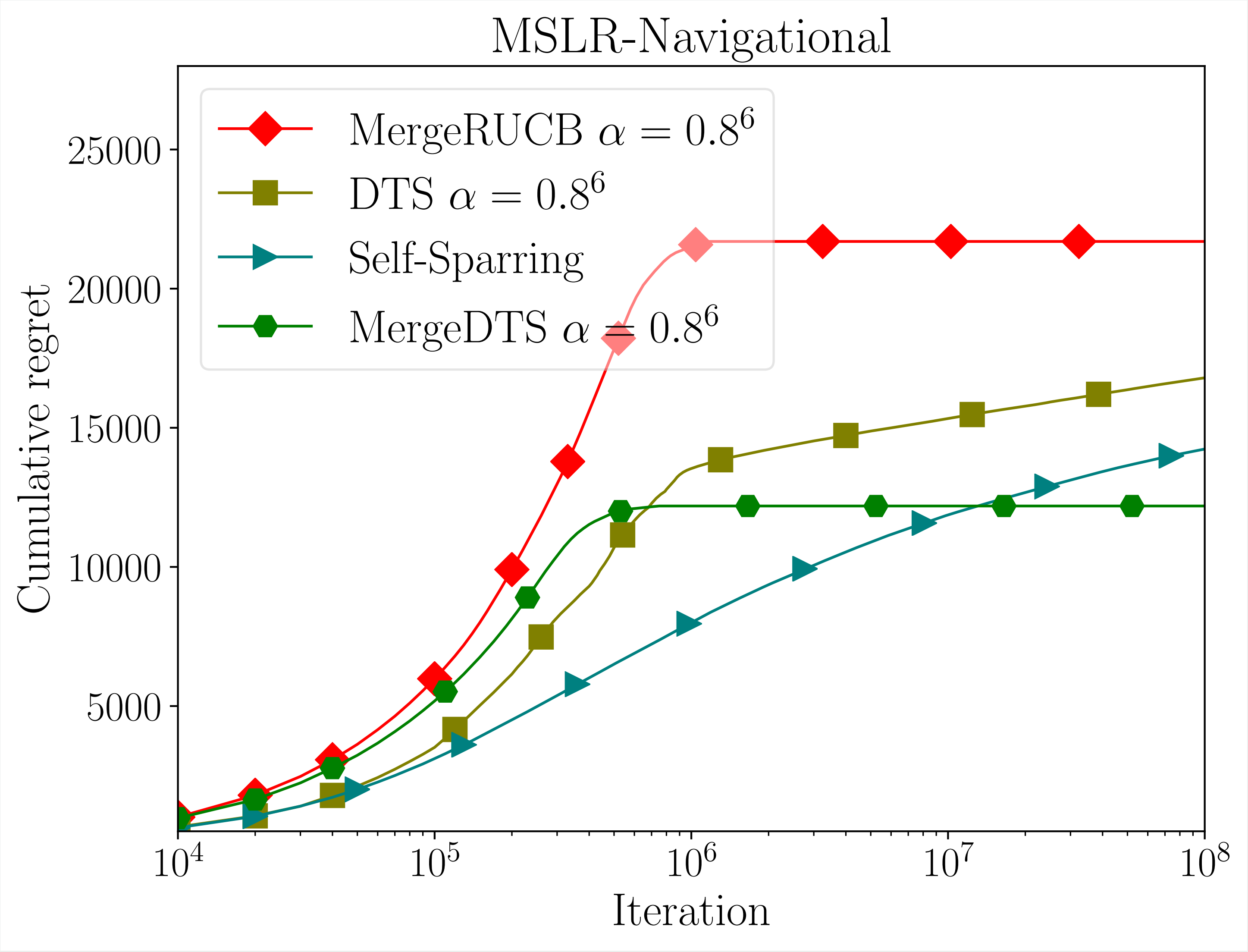
- **Self-Sparring** (Sui et al. UAI 2017) , etc.
  Designed under strict assumptions, i.e. ***not cyclic relationship***

- **MergeRUCB** (Zoghi, WSDM 2014)
  Designed for large scale dueling bandits yet with ***high cumulative regret***

Merge Double Thompson Sampling

- Randomly partition arms into small groups.
- Each time step:
  1. Sample a tournament inside a small group;
  2. Choose the winner and loser of the tournament as *l* and *r*, respectively;
  3. Compare *l* and *r* online, and update statistic;
  4. Eliminate an arm if it is dominated by any other arm with high confidence.
  5. If half arms are eliminated, re-partition rankers.
- Stop if only one arm left.

Experiment: online ranker evaluation


MSLR-Navigational
MergeRUCB $\alpha = 0.8^6$
DTS $\alpha = 0.8^6$
Self-Sparring
MergeDTS $\alpha = 0.8^6$
Cumulative regret
10^4
10^5
10^6
10^7
10^8
Iteration