Contextual Dueling Bandits

Miro Dudík (MSR)
Katja Hofmann (MSR)
Rob Schapire (MSR)
Alex Slivkins (MSR)
Masrour Zoghi (University of Amsterdam)

@katjahofmann
Contextual Dueling Bandits

Learn from partial relative feedback in the form of pairwise comparisons, given context.

In each round:

select two actions $a, b \in A = \{1 \ldots K\}$

observe outcome $o$ of a stochastic duel:

$$o = \begin{cases} 
+1 & \text{if } a \text{ beats } b \\
-1 & \text{if } b \text{ beats } a
\end{cases}$$

Goal: learn to select the best action.

Applications

Pairwise feedback is often more natural than absolute feedback when learning from user interactions, e.g., for web search, online news recommendation, ...
Contextual Dueling Bandits

Learn from partial relative feedback in the form of pairwise comparisons, given context.

In each round:
- environment chooses context $x \in X$
- select two actions $a, b \in A = \{1 \ldots K\}$
- observe outcome $o$ of a stochastic duel:
  $$o = \begin{cases} +1 \text{ if } a \text{ beats } b \\ -1 \text{ if } b \text{ beats } a \end{cases}$$

Goal: learn to select the best action, given context.

Applications
Pairwise feedback is often more natural than absolute feedback when learning from user interactions, e.g., for web search, online news recommendation, ...
Challenges & Contributions

How to define the “best” action?

Introduce new solution concept: the von Neumann winner (simple, natural, guaranteed to exist).

How to use context effectively?

Extend the von Neumann winner to the contextual dueling bandit setting.

Present learning algorithms for computing, approximating, or performing as well as the “best” solution.
Solution Concept

Previous solutions

Make assumptions about preference structure.

Example: **transitivity**
\[ P(a, b) > 0 \land P(b, c) > 0 \iff P(a, c) > 0 \]

Example: **Condorcet assumption**
\[ \exists a \in A \ \forall b \in A \ P(a, b) \geq 0 \]

All are frequently violated in practice.

**Define** \[ P(a, b) = E \text{ [outcome of duel between } a \text{ and } b] \]
“\( a \) beats \( b \)” when \( P(a, b) > 0 \); “\( a \) ties \( b \)” when \( P(a, b) = 0 \)
Solution Concept

Previous solutions

Make assumptions about preference structure.

Example: transitivity
\[ P(a, b) > 0 \land P(b, c) > 0 \iff P(a, c) > 0 \]

Example: Condorcet assumption
\[ \exists a \in A \quad \forall b \in A \quad P(a, b) \geq 0 \]

All are frequently violated in practice.

Our solution

Idea: Sample actions from a distribution \( w \).

Goal: find \( w \) such that beats or ties all other actions:
\[ \forall b : E [\text{outcome}] = \sum_a w(a)P(a, b) \geq 0 \]

When this holds, \( w \) is called the von Neumann winner (guaranteed to exist).

Define \( P(a, b) = E [\text{outcome of duel between } a \text{ and } b] \)

“\( a \) beats \( b \)” when \( P(a, b) > 0 \); “\( a \) ties \( b \)” when \( P(a, b) = 0 \)
Contextual Setting

Goal: find the “best” policy in $\Pi$

Extend von Neumann winner to contextual dueling bandits:
Randomized policy (distribution over $\Pi$) that beats or ties every other policy $\rho$ - guaranteed to exist
Contextual Setting

Goal: find the “best” policy in $\Pi$

Extend von Neumann winner to contextual dueling bandits:
Randomized policy (distribution over $\Pi$) that beats or ties every other policy $\rho$ - guaranteed to exist

Challenges

$\Pi$ typically huge! Need to solve a $|\Pi| \times |\Pi|$ game matrix $M$.
Even representing von Neumann winner seems to require $O(|\Pi|)$ space

Wanted: time, space, data in $\text{poly}(\log|\Pi|)$
Contextual Setting

Goal: find the “best” policy in $\Pi$

Extend von Neumann winner to contextual dueling bandits:
Randomized policy (distribution over $\Pi$) that beats or ties every other policy $\rho$ - guaranteed to exist

Challenges

$\Pi$ typically huge! Need to solve a $|\Pi| \times |\Pi|$ game matrix $M$.
Even representing von Neumann winner seems to require $O(|\Pi|)$ space

Wanted: time, space, data in $\text{poly}(\log|\Pi|)$

Algorithms

Full-explore-exploit setting
Optimal regret but $O(|\Pi|)$ time and space.

Explore-first setting
Suboptimal regret/approximation bound, $\text{poly}(\log|\Pi|)$ time and space.
Contextual Setting

Goal: find the “best” policy in $\Pi$

Extend von Neumann winner to contextual dueling bandits:
Randomized policy (distribution over $\Pi$) that beats or ties every other policy $\rho$ - guaranteed to exist

Challenges

$\Pi$ typically huge! Need to solve a $|\Pi| \times |\Pi|$ game matrix $M$.
Even representing von Neumann winner seems to require $O(|\Pi|)$ space

Wanted: time, space, data in $\text{poly}(\log|\Pi|)$

Algorithms

**Full-explore-exploit** setting
Optimal regret but $O(|\Pi|)$ time and space.

**Explore-first** setting
Suboptimal regret/approximation bound, $\text{poly}(\log|\Pi|)$ time and space.

Open problem: optimal-regret or optimal-approximation algorithm with $\text{poly}(\log|\Pi|)$ time and space!
Visit our poster to learn more about …

Contextual Dueling Bandits

Miro Dudík, Kaija Hofmann, Rob Schapire, Alex Slivkins (Microsoft Research), Masrour Zoghi (University of Amsterdam)

Motivation
Key motivation: learning to act (e.g., select recommendations, rank from user interactions in online interactive systems, e.g., search engines). Difficult to define absolute reward — interpretations as relative comparisons often easier and natural. Provide rich context (e.g., query, time of day).

Main Contributions
Introduce novel solution concept, the von Neumann winner (simple, natural, guaranteed to work). Introduce and study general version of contextual dueling bandits.

The von Neumann Winner
Define a distribution w from which actions are sampled at random.

Goal find w such that

\[ \frac{1}{W} \sum w(a|u, b) \leq \frac{1}{W} \sum w(a|u, c) \leq \frac{1}{W} \sum w(a|u, b) \]

When this holds, w is called the von Neumann winner. With context, find randomized policy distribution over policies π and extract the von Neumann winner.

Algorithms
Algorithm: approx Point Policy (Bakshy, 2011):

- dom 2 copies of DPP [P] to address context
- optimal regret
- O(n) time and space

Open Problem
Algorithms with optimal regret or optimal-approximation using \( poly(\log|I|) \) time and space.

Applications
von Neumann winners

… and more!