Completing the classification of adversarial partial monitoring games

Tor Lattimore & Csaba Szepesvári
DeepMind & University of Alberta
Outline

• Why and what: partial monitoring
• Spam-game
• The classification result
• Low regret for the easy case:
  Smooth NeighborhoodWatch
• Conclusions and outlook
Why and what

**Full information**  Observe all losses

**Bandits**  Loss for chosen action only

**Partial monitoring**  Loss not directly observed
Two decades of history

- Rustichini
  - ’99
  - $R^*_n = o(n)$

- Piccelboni and Schindelhauer
  - ’01
  - $R^*_n = O(n^{3/4})$

- Cesa-Bianchi, Lugosi, Stoltz
  - ’06
  - $R^*_n = \Theta(n^{2/3})$

- Alberta group (Antos, Bartók, Pál, Sz.) ‘10–’14
  - classification

- Foster and Rakhlin
  - ’12
  - easy, adv.
Game played over $n$ rounds

Adversary secretly chooses outcomes

\[ i_1, i_2, ..., i_n \in \{1, 2, ..., E\} =: [E] \]

In round $t$ learner chooses action $A_t = \{1, 2, ..., K\} = [K]$

Observes feedback $\Phi_t = \Phi_{A_t i_t}$ where $\Phi \in \Sigma^{K \times E}$

$\Sigma$ is a finite set of feedbacks with $|\Sigma| = F$

Suffers loss $y_{tA_t}$ where $y_{ta} = \mathcal{L}_{ai_t}$ where $\mathcal{L} \in [0,1]^{K \times E}$

$\Phi$ and $\mathcal{L}$ are known

Regret of policy $\pi$ in partial monitoring game $G = (\mathcal{L}, \Phi)$ is

\[
R_n(G, \pi, i_{1:n}) = \max_a \sum_{t=1}^n (y_{tA_t} - y_{ta})
\]

Minimax regret $R^*_n(G) = \min_{\pi} \max_{i_{1:n}} \mathbb{E}[R_n(G, \pi, i_{1:n})]$
• Email classification
• Only receive feedback by paying human expert (let’s pretend)

How to play this game?
What is the regret?
General games?
Theorem (Bartók et al. 2014)

\[ R^*_n(G) = \begin{cases} 
0 & \text{if } c = 0 \\
\tilde{\Theta}(\sqrt{n}) & \text{if } c \in (0, 1/2) \\
? & \text{if } c = 1/2 \\
\Theta(n^{2/3}) & \text{if } c \in (1/2, \infty) \\
\Omega(n) & \text{if } c = \infty.
\end{cases} \]

Minimax regret depends on \textit{price of information}
THE CLASSIFICATION PROBLEM
“Hard” games: $c > 1/2$ (why?)

\[ 1/2 < c < \infty \]
\[ R_n^* = \Omega(n^{2/3}) \]
\[ \mathcal{L} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ c & c \end{bmatrix} \]
\[ \Phi = \begin{bmatrix} \bot & \bot \\ \bot & \bot \\ s & \text{ns} \end{bmatrix} \]

\[ R_n^* = c \cdot \left( n \varepsilon \land \frac{c - 1/2 + \varepsilon}{\varepsilon^2} \right) \]

Set $\varepsilon = n^{-1/3}$. 

Loss of actions

\[ \mathbb{P}(\text{spam}) \]

\[ \frac{1}{2} - \varepsilon \]

\[ \frac{1}{2} + \varepsilon \]

\[ c - \frac{1}{2} + \varepsilon \geq c - 1/2 \]
From “hard” to “easy” games

\[ \ell_m \geq \ell_t \geq \ell_m - \varepsilon \]

Loss of actions

\[ \mathbb{P}(\text{spam}) \]

\[ c > 1/2 \]

\[ c < 1/2 \]
Why easy when \( c < \frac{1}{2} \)?

- Loss of actions

\[
\begin{align*}
\text{spam}=s & \quad \rightarrow \quad a=\neg \text{spam} \quad \text{is optimal} & \quad C_{ns} \\
\neg \text{spam}=\text{ns} & \quad \rightarrow \quad a=\text{tm} \quad \text{is optimal} & \quad C_{tm} \\
& \quad \rightarrow \quad a=\text{spam} \quad \text{is optimal} & \quad C_s
\end{align*}
\]

- \( c > \frac{1}{2} \): \( C_{tm} = \emptyset \)
- \( c = \frac{1}{2} \): \( C_{tm} = \{\frac{1}{2}\} \)
- \( c < \frac{1}{2} \): \( C_{tm} = [c, 1-c] \)
Some definitions

- $\ell_a$: $a$th row of $\mathcal{L}$ (losses under action $a$)
- $C_a = \{ u \in \Delta_{E-1} : \langle \ell_a, u \rangle \leq \min_b \langle \ell_b, u \rangle \}$
- $\dim(a) = \dim(\text{aff}(C_a))^1$
- $\dim(a) = E - 1$: $a$ Pareto optimal
- $\dim(a) < E - 1$: $a$ degenerate
- $C_a = \emptyset$: $a$ dominated
- $a, b$ duplicates if $\ell_a = \ell_b$
- $a, b$ Pareto optimal are neighbors if $\dim(\text{aff}(C_a \cap C_b)) = E - 2$

$u_3 = 1 - u_1 - u_2$
$E = 3$

$u_0 = \frac{1}{2}$
$u_0^* = \frac{1}{2}$
$C_0, C_*$
$C_2$
$C_{1,2}$
$C_5$
$C_{1,7}$
$C_1, C_{3,4}$
$C_{1,3}$
$C_{2,4}$
$C_{1,5}$
$C_{2,5}$
$C_{3,6}$
$C_{4,6}$
$C_{5,6}$
$C_7$

\[ -1 \text{ when } C_a = \emptyset \]
Neighborhoods

- \( a \) Pareto; 
  \( \mathcal{N}_a: a, \) its dups and neighbors
- \( a, b \) (Pareto) neighbors: 
  \( \mathcal{N}_{ab} = \{c: C_a \cap C_b \subset C_c\} \)

\( \mathcal{N}_{13} = \mathcal{N}_{14} = \mathcal{N}_{31} = \mathcal{N}_{41} = \{1,3,4\} \)
\( \mathcal{N}_{23} = \{2,3,4,5\} \)
\( \mathcal{N}_{12} = \mathcal{N}_{21} = \emptyset \)

Cheap actions to compare \( a, b \)

Neighborhood graph:

- \( \mathcal{N}_1 = \{1,3,4\}, \mathcal{N}_2 = \{2,3,4\} \)
- \( \mathcal{N}_3 = \{1,2,3,4\}, \mathcal{N}_4 = \{1,2,3,4\} \)
Neighborhood lemma

**Lemma:** \( c \in \mathcal{N}_{ab} \Rightarrow \exists! \alpha \in [0,1] \) s.t.
\[
\ell_c = \alpha \ell_a + (1 - \alpha)\ell_b
\]

**Corollary:** \( c \in \mathcal{N}_{ab} \) is not a duplicate of \( a, b \Leftrightarrow \alpha \in (0,1) \)
Further, \( C_c = C_a \cap C_b \)
Learning strategy

• General strategy (e.g., Exp3):
  – Step 1: Estimate losses
  – Step 2: Feed estimated losses to a full-information algorithm (e.g., EWA/Mirror Descent)

• Trouble with this strategy:
  – Estimating losses can be impossible even though sublinear regret may be possible

• Idea: Estimate loss differences
  – Only for neighboring (hence, Pareto optimal) actions
  – Sufficient and necessary for figuring out the optimal action in hindsight

\[ \mathcal{L} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \end{bmatrix} \]
\[ \Phi = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 \end{bmatrix} \]
How to estimate loss differences?

• Given \( P \in \mathcal{P}_{K-1} \), \( A \sim P \), \( \Phi_{A,i} \) observed, \( i \) unobserved.

Need unbiased estimate of \( \mathcal{L}_{a,i} - \mathcal{L}_{b,i} \)

• Necessary and sufficient: \( \exists \nu: [K] \times [F] \to \mathbb{R} \) such that

\[
\mathbb{E}[\nu(A, \Phi_{A,i})/P_A] = \mathcal{L}_{a,i} - \mathcal{L}_{b,i}
\]

regardless of \( i \in [E] \) (which we cannot control)

• Condition: \( \exists \nu = \nu^{ab} \) such that:

\[
(E) \quad \sum_c \nu(c, \Phi_{c,i}) = \mathcal{L}_{a,i} - \mathcal{L}_{b,i} \text{ holds } \forall i \in [E]
\]
Hopeless, hard, easy

• If there are neighbors $a, b$ such that no function $v^{ab}$ satisfies (E), then $G = (\mathcal{L}, \Phi)$ is **hopeless**, $R_n^* = \Omega(n)$

• If this is not the case and if $G$ is not **trivial**, $G$ will be “**easy**” or “**hard**” to learn depending on whether we can find $v^{ab}$ for all $a, b$ neighbors such that for any feedback $f \in [F]$, $v^{ab}(\cdot, f)$ is **supported on** $\mathcal{N}_{ab}$
• $a, b$ neighbors are **globally observable** if $\exists v^{ab}$ that satisfies (E)

• $a, b$ neighbors are **locally observable** if $\exists v^{ab}$ that satisfies (E) and $v^{ab}(c, f) = 0$ for all $f \in [F], c \notin N_{ab}$

• **$G$ is globally observable** if all neighbors are globally observable

• **$G$ is locally observable** if all neighbors are locally observable
Back to the spam-game

\[ \sum_{c} \nu^{ab}(c', \Phi_{c,i}) = \mathcal{L}_{a,i} - \mathcal{L}_{b,i} \]

\[ \nu^{s,ns}(tm, s) = 0 - 1 = -1 \]
\[ \nu^{s,ns}(tm, ns) = 1 - 0 = +1 \]
\[ \nu^{s,ns}(c', f) = 0, \text{ ow.} \]

Globally, but not locally observable

Locally observable

\[ \nu^{ns,tm}(tm, ns) = 0 - c = -c \]
\[ \nu^{ns,tm}(tm, s) = 1 - c \]
\[ \nu^{ns,tm}(c', f) = 0, \text{ ow} \]
And $c = 1/2$?

$\mathcal{N}_{ns,s} = \{s, ns, tm\}$

$v^{ns,s}(tm, ns) = 0 - c$
$v^{ns,s}(tm, s) = 1 - c$
$v^{ns,s}(c, f) = 0$, ow.

$\sum_{c, \nu^{ab}(c', \Phi_{c,i}) = \mathcal{L}_{a,i} - \mathcal{L}_{b,i}}$

Loss of actions

Locally observable still!
Classification

Theorem (Lattimore-Sz’18):

All games can be classified as either: trivial, easy, hard or hopeless and

\[
R_n^*(G) = \begin{cases} 
0 & \text{if } G \text{ has no neighboring (nb) actions} \\
\Theta(\sqrt{n}) & \text{if } G \text{ is locally observable, has nb actions} \\
\Theta(n^{2/3}) & \text{if } G \text{ is globally, but not locally observable} \\
\Omega(n) & \text{otherwise}
\end{cases}
\]

The hidden constants depend on $L$ and $\Phi$, including, but not limited to dependence on $K, E$ and $F$

What’s new? We allow for degenerate actions.
“Easy” case upper bound was not known beforehand
EASY GAMES AND DEGENERATE ACTIONS
NeighborhoodWatch2 (NW2)

Let $G = (\mathcal{L}, \Phi)$ be a partial monitoring game that is locally observable and has neighboring actions with losses in $[0,1]$

**Theorem (Lattimore-Sz’ 18):**
There exists a game dependent constant $c_G > 0$ such that for any $n > 0$, $\delta \in (0,1)$ and sequence $i_{1:n}$ of outcomes, with probability at least $1 - \delta$

$$\hat{R}_n(G, NW2(\delta), i_{1:n}) \leq c_G \sqrt{n \log(e/\delta)}$$

NeighborhoodWatch2: refined, improved, amended version of the NeighborhoodWatch algorithm by B-F-P-Sz-R, which is based on Foster and Rakhlin’s work
Prepare: Split actions into “for play” and “info” actions (duplicates, degenerates become “info” actions)

Repeat:

1. Use Exp3 based on local relative loss estimates to come up with local distributions over “for play” actions

2. Merge these local probability distributions into a single global distribution (Markov chain); supported on set of “for play” actions

3. **NEW:** Let “for play” actions distribute just enough of their probability mass to neighboring “info” actions while not changing expected loss

4. Add exploration (optional for expected regret)

5. Play from resulting distribution

6. Estimate relative losses with estimator functions

We also eliminate the “switching mechanism” from NeighborhoodWatch --- all loss estimates are updated in all time steps.
Summary

• The new algorithm closed the gap in the classification problem of finite partial monitoring problems

• We extended NeighborhoodWatch to degenerate games
  – NeighborhoodWatch updates losses only in some neighborhood
  – Our method updates all losses
  – Simplifies proof, improves effectiveness

• Ideas that don’t work:
  – Leave out degenerate actions
  – Make them of the same rank as “for play” actions (include them in EWA)

• Key: Probability redistribution. Works because of the “Neighborhood lemma”

• Markov “merging” is magical!
Open problems

• Dependence on constants
• How big is this $V$? When no degenerate actions $V \leq 1 + F$, dependence on $F$ is "real" then.
• Game-dependent lower bounds (not just asymptotics)
• Localization
• Easy sub-games and adaptivity
• Simpler algorithms (one to rule them all)
• Understanding regime changes, even in stochastic case
• Continuous feedback/outcomes spaces
• Continuous action spaces
• Contextual version
For further “exploration”

• Blog/lecture notes/upcoming book: [http://banditalgs.com](http://banditalgs.com)
• Joint effort with Tor Lattimore

• **Contents:**
  – Stochastic and adversarial bandits
  – Linear and contextual bandits
  – Combinatorial bandits
  – Ranking
  – Partial monitoring
  – MDPs (RL)
  – and more!

• **Software:**
  [https://github.com/tor/libbandit](https://github.com/tor/libbandit)
Questions?