15-888 Computational Game Solving (Fall 2021) Thu, September 30th 2021

Lecture 8

Predictive Blackwell approachability

Instructor: Gabriele Farina*

In Lecture 4 we constructed a regret minimizer, called Regret Matching, by solving a suitable Blackwell approachability game. In this lecture, we will do the opposite: we will investigate how regret minimization algorithm can give rise to Blackwell approachability algorithms. From there, we use *predictive* regret minimization algorithms to arrive at *predictive* Blackwell approachability algorithms.

1 Using regret minimization to solve Blackwell approachability games

Recall that a Blackwell approachability game is a tuple $(\mathcal{X}, \mathcal{Y}, \boldsymbol{u}, S)$, where $\mathcal{X}, \mathcal{Y}$ are closed convex sets, $\boldsymbol{u} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ is a biaffine function, and $S \subseteq \mathbb{R}^d$ is a closed and convex *target set*. A Blackwell approachability game represents a vector-valued repeated game between two players. At each time t, the two payers interact in this order:

- first, Player 1 selects an action $\boldsymbol{x}^t \in \mathcal{X}$;
- then, Player 2 selects an action $\boldsymbol{y}^t \in \mathcal{Y}$, which can depend adversarially on all the $\boldsymbol{x}^t$ output so far;
- finally, Player 1 incurs the vector-valued payoff $\boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) \in \mathbb{R}^d$, where $\boldsymbol{u}$ is a biaffine function.

Player 1's objective is to guarantee that the average payoff converges to the target set S. Formally, given target set $S \subseteq \mathbb{R}^d$, Player 1's goal is to pick actions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots \in \mathcal{X}$ such that no matter the actions $\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots \in \mathcal{Y}$ played by Player 2,

$$\min_{\hat{\boldsymbol{s}} \in S} \left\| \hat{\boldsymbol{s}} - \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) \right\|_2 \to 0 \quad \text{as} \quad T \to \infty. \tag{1}$$

As we discussed in Lecture 4, Blackwell's theorem states that goal (1) can be attained if and only if any halfspace $\mathcal{H} \supseteq S$ is *forceable*, where forceability is recalled next.

Definition 1.1 (Forceable halfspace)**.** Let $(\mathcal{X}, \mathcal{Y}, \boldsymbol{u}, S)$ be a Blackwell approachability game and let $\mathcal{H} \subseteq \mathbb{R}^d$ be a halfspace, that is, a set of the form $\mathcal{H} = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{a}^\top \boldsymbol{x} \le b\}$ for some $\boldsymbol{a} \in \mathbb{R}^d, b \in \mathbb{R}$. The halfspace $\mathcal{H}$ is said to be *forceable* if there exists a strategy of Player 1 that guarantees that the payoff is in $\mathcal{H}$ no matter the actions played by Player 2, that is, if there exists $\boldsymbol{x}^* \in \mathcal{X}$ such that

$$\boldsymbol{u}(\boldsymbol{x}^*, \boldsymbol{y}) \in \mathcal{H} \qquad \forall\, \boldsymbol{y} \in \mathcal{Y}.$$

When that is the case, we call action $\boldsymbol{x}^*$ a *forcing action* for $\mathcal{H}$.

Abernethy et al. [2011] showed that it is always possible to convert a regret minimizer into an algorithm for a Blackwell approachability game (that is, an algorithm that chooses actions $\boldsymbol{x}^t$ at all times t in such a way that goal (1) holds no matter the actions $\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots$ played by the opponent).(Gordon's Lagrangian Hedging [Gordon, 2005, 2006] partially overlaps with the construction by Abernethy et al. [2011].)

*Computer Science Department, Carnegie Mellon University. gfarina@cs.cmu.edu.

1.1 A couple preliminaries on convex cones

For simplicity, we will only be interested in Blackwell games whose target sets are (nonempty) closed convex cones $S \subseteq \mathbb{R}^n$.

Definition 1.2. A cone is a set such that for any point $\boldsymbol{s} \in S$, the rescaled point $\lambda \boldsymbol{s}$ belongs to S for any $\lambda \in \mathbb{R}_{\geq 0}$. In particular, $\mathbf{0} \in S$ for any nonempty cone.

Cones have a very regular geometry that will make constructing approachability algorithms simpler. This simplicity actually doesn't come at a generality cost: one of the contributions of Abernethy et al. [2011] is to show that any Blackwell approachability game with non-conic target set can be studied and solved by first transforming the problem into a slightly larger Blackwell approachability game with conic target set.

A standard concept in conic geometry is that of the *polar cone*, which we now define.

Definition 1.3. The *polar* of cone S, denotes S°, is defined as the set of all vectors that form an obtuse angle with the cone S, that is,

$$S^\circ := \{\boldsymbol{w} \in \mathbb{R}^n : \boldsymbol{w}^\top \boldsymbol{s} \leq 0 \quad \forall \boldsymbol{s} \in S\}.$$

The polar S° is itself a closed and convex cone provided that S is a closed and convex cone.

The reason we care about the polar of S is that it gives a characterization of important halfspaces $\mathcal{H} \supseteq S$, which are so crucial to Blackwell's theorem.

Lemma 1.1. Let $\boldsymbol{\theta} \in S^\circ$ and consider the halfspace $\mathcal{H}_{\boldsymbol{\theta}} := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{\theta}^\top \boldsymbol{x} \leq 0\}$. Then, $\mathcal{H}_{\boldsymbol{\theta}} \supseteq S$.

Proof. Take any $\boldsymbol{s}' \in S$; we will show that $\boldsymbol{s}' \in \mathcal{H}_{\boldsymbol{\theta}}$. Since $\boldsymbol{\theta} \in S^\circ$, by definition of polar cone we have that $\boldsymbol{\theta}^\top \boldsymbol{s} \leq 0$ for all $\boldsymbol{s} \in S$, including in particular $\boldsymbol{s} = \boldsymbol{s}'$. So, $\boldsymbol{s}' \in \mathcal{H}_{\boldsymbol{\theta}}$ as we wanted to show. □

1.2 Abernethy et al. [2011]'s idea

Blackwell's algorithm described in Lecture 4 worked by playing, at every time t, a forcing actions for the halfspace tangent to S at the projection point $\boldsymbol{\psi}^t \in S$ of the current average payoff $\bar{\boldsymbol{\phi}}^t := \frac{1}{T}\sum_{\tau=1}^{t-1} \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t)$. Abernethy et al. [2011]'s idea is to generalize this construction by letting a regret minimizer decide which halfspace to force.

Specifically, let $\mathcal{R}_S$ be a regret minimizer that outputs strategies $\boldsymbol{\theta}^t \in S^\circ$ that observes as utilities the Blackwell payoffs $\boldsymbol{\ell}^t := \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t)$. At every time t, we will force the halfspace

$$\mathcal{H}_{\boldsymbol{\theta}^t} := \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{\theta}^t)^\top \boldsymbol{x} \leq 0\},$$

which, as we discussed in Lemma 1.1, is a superset of the target set S (see also Figure 1).

The proof of correctness for Algorithm 1 relies on this lemma that shows that the problem of *minimizing* distance to a cone is equivalent to the problem of *maximizing* the inner product on the polar of the cone.

Lemma 1.2. Let $S \subseteq \mathbb{R}^n$ be a cone and $\boldsymbol{z}$ be a generic point in $\mathbb{R}^n$. Then,

$$\min_{\hat{\boldsymbol{s}} \in S} \|\hat{\boldsymbol{s}} - \boldsymbol{z}\|_2 = \max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} \boldsymbol{z}^\top \hat{\boldsymbol{\theta}},$$

where $\mathbb{B}_2^n := \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 \leq 1\}$ denotes the unit ball in $\mathbb{R}^n$ with respect to the Euclidean norm.

Algorithm 1: From regret minimization to Blackwell approachability

Data: $\mathcal{R}_S$ regret minimizer for S°

```
function NextStrategy()
    θ^t ← R_S.NextStrategy()
    return x^t forcing action for H_{θ^t} := {x : (θ^t)^⊤ x ≤ 0}

function ReceivePayoff(u(x^t, y^t))
    R_S.ObserveLoss(ℓ^t := u(x^t, y^t))
```

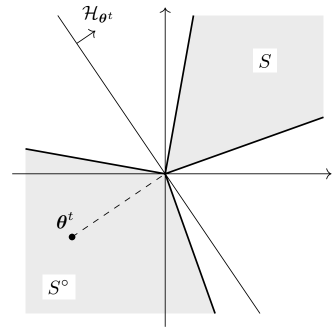

Figure 1: Pictorial depiction of Algorithm 1's inner working: at all times t, the algorithm plays a forcing action for the halfspace H^t induced by the last decision output by $\mathcal{L}$.

Proposition 1.1. Denote the regret of $\mathcal{R}_S$ compared to any $\hat{\boldsymbol{\theta}}$ as

$$R_S^T(\hat{\boldsymbol{\theta}}) \coloneqq \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{\theta}} - \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \boldsymbol{\theta}^t.$$

Then, at all times T, the distance between the average payoff cumulated by Algorithm 1 and the target cone S is upper bounded as

$$\min_{\hat{\boldsymbol{s}} \in S} \left\| \hat{\boldsymbol{s}} - \frac{1}{T}\sum_{t=1}^T \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) \right\|_2 \le \frac{1}{T} \max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} R_S^T(\hat{\boldsymbol{\theta}}),$$

where $\mathbb{B}_2^n$ denotes the unit ball in $\mathbb{R}^n$ with respect to the Euclidean norm, just like in Lemma 1.2.

Proof. Using Lemma 1.2,

$$\min_{\hat{\boldsymbol{s}} \in S} \left\| \hat{\boldsymbol{s}} - \frac{1}{T}\sum_{t=1}^T \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) \right\|_2 = \max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} \left(\frac{1}{T}\sum_{t=1}^T \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) \right)^\top \hat{\boldsymbol{\theta}} = \max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} \left(\frac{1}{T}\sum_{t=1}^T \boldsymbol{\ell}^t \right)^\top \hat{\boldsymbol{\theta}}$$

$$= \frac{1}{T} \max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} \left\{ \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \hat{\boldsymbol{\theta}} \right\} \tag{2}$$

where the second step uses $\boldsymbol{\ell}^t \coloneqq \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t)$. By substituting the definition $R_S^T(\hat{\boldsymbol{\theta}})$ into (2), we then find

$$\min_{\hat{\boldsymbol{s}} \in S} \left\| \hat{\boldsymbol{s}} - \frac{1}{T}\sum_{t=1}^T \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) \right\|_2 = \frac{1}{T} \max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} \left\{ R_S^T(\hat{\boldsymbol{\theta}}) + \sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \boldsymbol{\theta}^t \right\}$$

$$= \frac{1}{T} \max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} \left\{ R_S^T(\hat{\boldsymbol{\theta}}) \right\} + \frac{1}{T}\sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \boldsymbol{\theta}^t.$$

Now, by construction $\boldsymbol{x}^t$ is a forcing action for the halfspace $\mathcal{H}_{\boldsymbol{\theta}^t} = \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{\theta}^t)^\top \boldsymbol{x} \le 0\}$, and so $(\boldsymbol{\theta}^t)^\top \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) = (\boldsymbol{\ell}^t)^\top \boldsymbol{\theta}^t \le 0$. Hence,

$$\frac{1}{T}\sum_{t=1}^T (\boldsymbol{\ell}^t)^\top \boldsymbol{\theta}^t \le 0. \tag{3}$$

Plugging (3) into (2) yields the statement. □

Proposition 1.1 immediately implies that if the regret minimizer $\mathcal{R}_S$ is able to guarantee that the regret on the subset $S^\circ \cap \mathbb{B}_2^n$ of its domain S° grows sublinearly, then goal (1) can be attained.

Algorithms that are able to guarantee that $\max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} R_S^T(\hat{\boldsymbol{\theta}}) = o(T)$ exist. For example, if $\mathcal{R}_S$ is set to OMD or FTRL with Euclidean regularization, then it can be shown that

$$\max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} R_S^T(\hat{\boldsymbol{\theta}}) \le \sqrt{2\left(\sum_{t=1}^{T} \|\boldsymbol{\ell}^t\|_2^2\right)},$$

which clearly grows at a sublinear rate of $O(\sqrt{T})$.

2 Predictive Blackwell Approachability

Predictive Blackwell approachability is a natural extension of Blackwell approachability [Farina et al., 2021]. Similarly to how we defined *predictive* regret minimization, in predictive Blackwell approachability the environment provides Player 1 with a prediction $\boldsymbol{v}^t$ of the next utility $\boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t)$.

It is immediate to extend the construction of Abernethy et al. [2011] (Algorithm 1) to take into account predictions: since the utility observed by $\mathcal{R}_S$ (Line 5) is exactly $\boldsymbol{u}^t(\boldsymbol{x}^t, \boldsymbol{y}^t)$, we can simply use a predictive regret minimization algorithm $\mathcal{R}_S$ and provide $\boldsymbol{v}^t$ as the prediction of the next utility. The predictive version of Algorithm 1 is given in Algorithm 2.

The analysis in Proposition 1.1 holds verbatim. In fact, it can be shown that when $\mathcal{R}_S$ is set to *predictive* OMD or FTRL with Euclidean regularization, then

$$\max_{\hat{\boldsymbol{\theta}} \in S^\circ \cap \mathbb{B}_2^n} R_S^T(\hat{\boldsymbol{\theta}}) \le \sqrt{2\left(\sum_{t=1}^{T} \|\boldsymbol{\ell}^t - \boldsymbol{v}^t\|_2^2\right)},$$

which clearly grows at a sublinear rate of $O(\sqrt{T})$ and can be very small if the predictions $\boldsymbol{v}^t$ are accurate.

Algorithm 2: Predictive Blackwell approachability algorithm

Data: $\mathcal{R}_S$ *predictive* regret minimizer for S°

1 **function** NextStrategy($\boldsymbol{v}^t$)
  [▷ $\boldsymbol{v}^t$ is the prediction of the next Blackwell payoff $\boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t) \in \mathbb{R}^n$]
2   $\boldsymbol{\theta}^t \leftarrow \mathcal{R}_S$.NextStrategy($\boldsymbol{v}^t$)
3   **return** $\boldsymbol{x}^t$ *forcing action for* $\mathcal{H}_{\boldsymbol{\theta}^t} := \{\boldsymbol{x} : (\boldsymbol{\theta}^t)^\top \boldsymbol{x} \le 0\}$

4 **function** ReceivePayoff($\boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t)$)
5   $\mathcal{R}_S$.ObserveLoss($\boldsymbol{\ell}^t := \boldsymbol{u}(\boldsymbol{x}^t, \boldsymbol{y}^t)$)

References

Jacob Abernethy, Peter L Bartlett, and Elad Hazan. Blackwell approachability and no-regret learning are equivalent. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 27–46, 2011.

Geoffrey J. Gordon. No-regret algorithms for structured prediction problems. Technical Report CMU-CALD-05-112, Carnegie Mellon University, 2005.

Geoffrey J Gordon. No-regret algorithms for online convex programs. *Advances in Neural Information Processing Systems*, 19:489, 2006.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive Blackwell approachability: Connecting regret matching and mirror descent. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2021.