Chapter 1
Introduction

Ioannis Anagnostides, Gabriele Farina, and Brian Hu Zhang*

This introductory chapter covers basic background on regret minimization and connections to game-theoretic equilibrium concepts. In particular, we begin by introducing and motivating the notion of Φ-*regret* and its associated solution concept in games—Φ-*equilibrium*. In the second part, we will introduce the canonical algorithmic template for minimizing Φ-*regret* due to Gordon, Greenwald and Marks [GGM08].

1.1 Online learning and regret

We consider a *learner* who makes a sequence of decisions over T rounds. The learner interacts repeatedly with an *environment*. In each round $t \in [T]$, the learner specifies a mixed strategy $\boldsymbol{x}^{(t)} \in \mathcal{X}$, where $\mathcal{X}$ is a convex and compact set. (A canonical case arises when $\mathcal{X}$ is a probability simplex over a finite set of actions, but our focus here is on the general setting.) The environment then selects a *utility vector* $\boldsymbol{u}^{(t)}$, so that the utility obtained by the learner at that round is $\left\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \right\rangle$. In the full-feedback setting, which is the focus of these notes, $\boldsymbol{u}^{(t)}$ is revealed to the learner after the end of the round. An online algorithm produces a sequence of strategies based on the feedback observed up to that point.

What's a sensible way of measuring the performance of the learner in this online environment? There are different notions of *hindsight rationality*. Perhaps the most common performance benchmark is *external regret*, defined as

$$\mathrm{Reg}^{(T)} := \max_{\boldsymbol{x} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} \left\langle \boldsymbol{x}, \boldsymbol{u}^{(t)} \right\rangle \right\} - \sum_{t=1}^{T} \left\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \right\rangle. \tag{1}$$

The second term in the right-hand side of (1) is the cumulative utility obtained by the learner through the T rounds, whereas the first term is the optimal utility that could have been obtained in hindsight *through a fixed strategy*. We will soon introduce more powerful notions of hindsight rationality beyond external regret.

*✉ ianagnos@cs.cmu.edu, {gfarina,zhangbh}@mit.edu. These tutorial notes have not undergone formal peer review. We are grateful for any feedback or reports of typos.

1.2 Games and solution concepts

We will often need to analyze what happens when multiple no-regret players repeatedly interact in a game. To do so, we begin by introducing the canonical normal-form representation of games. While any finite game can be cast in normal form, that representation is often inefficient. This will motivate introducing more compact game representations, as we shall do in the sequel.

Formally, we have a set of n players. In a normal-form game, each player $i \in [n]$ has a finite set of available actions $\mathcal{A}_i$; we will use the shorthand notation $m_i := |\mathcal{A}_i|$ for the number of actions. Every player $i \in [n]$ has a *utility function* u_i mapping a joint action profile $(a_1, ..., a_n) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ to a real value $u_i(a_1, ..., a_n)$. Players can randomize by specifying a probability distribution over their available actions, so that the strategy set of each player is the probability simplex $\mathcal{X}_i = \Delta(\mathcal{A}_i)$. Under a joint strategy $(x_1, ..., x_n) \in \Delta(\mathcal{A}_1) \times \cdots \times \Delta(\mathcal{A}_n)$, the *expected utility* of player $i \in [n]$ reads

$$\begin{aligned} u_i(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) &:= \mathop{\mathbb{E}}_{(a_1,...,a_n)\sim(\boldsymbol{x}_1,...,\boldsymbol{x}_n)}[u_i(a_1, ..., a_n)] \\ &= \sum_{(a_1,...,a_n)\in\mathcal{A}_1\times\cdots\times\mathcal{A}_n} \prod_{i'=1}^{n} \boldsymbol{x}_{i'}[a_{i'}] u_i(a_1, ..., a_n). \end{aligned}$$

Each player is trying to maximize its expected utility.

1.2.1 Correlated and coarse correlated equilibria

A major criticism of the Nash equilibrium is that, even though one always exists—as guaranteed by the famous theorem of John Nash [Nas50]—it is computationally intractable to find one [DGP08]—let alone a welfare-optimal one [GZ89]. As result, we shouldn't expect simple, computationally bounded learning algorithms to converge to Nash equilibria; this raises the question: what are no-regret dynamics converging to in general-sum games?

It turns out that no-regret learning is inherently tied to *coarse correlated equilibria* [MV78]. Let's begin by recalling the basic definition and start building some intuition about this solution concept; for now, we restrict our attention to normal-form games.

Definition 1.1 (Coarse correlated equilibrium)**.** A correlateddistribution $\mu \in \Delta(\mathcal{A}_1 \times \cdots \times \mathcal{A}_n)$ is an *ε-coarse correlated equilibrium (CCE)* if for any player $i \in [n]$ and deviation $a_{i'} \in \mathcal{A}_i$,

$$\mathop{\mathbb{E}}_{(a_1,...,a_n)\sim\mu}[u_i(a_1, ..., a_n)] \geq \mathop{\mathbb{E}}_{(a_1,...,a_n)\sim\mu}[u_i(a_{i'}, a_{-i})] - \varepsilon. \tag{2}$$

This definition mirrors Nash equilibria, but with a critical difference: the underlying distribution μ can be *correlated*; by contrast, in a Nash equilibrium μ has to be a *product distribution*, reflecting the fact that players randomize independently. To explain this, let's consider the following two distributions with respect to some 2×2 bimatrix game (meaning that each of the two players has two available actions):

$$\mu = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}, \quad \mu' = \begin{pmatrix} \frac{1}{6} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{3} \end{pmatrix}.$$

Both are distributions over $\mathcal{A}_1 \times \mathcal{A}_2 = \{1,2\} \times \{1,2\}$, but only μ' is a product distribution. Indeed, if Player 1—the row player—plays $\left(\frac{1}{3}, \frac{2}{3}\right)$ and Player 2 plays $\left(\frac{1}{2}, \frac{1}{2}\right)$, the induced distribution over the 4 outcomes matches μ'. In contrast, no pair of strategies gives rise to μ.

A Nash equilibrium is always a CCE; a Nash equilibrium is basically an *uncorrelated* (coarse) correlated equilibrium. But the set of CCEs can unlock new outcomes. Before we examine a concrete example, we also introduce the stronger notion of a *correlated equilibrium*, famously put forward by Aumann [Aum74].

Definition 1.2 (Correlated equilibrium)**.** A correlated distribution $\mu \in \Delta(\mathcal{A}_1 \times \cdots \times \mathcal{A}_n)$ is an *ε-correlated equilibrium (CE)* if for any player $i \in [n]$ and deviation function $\varphi_i : \mathcal{A}_i \to \mathcal{A}_i$, we have

$$\mathop{\mathbb{E}}_{(a_1,\ldots,a_n)\sim\mu}[u_i(a_1,\ldots,a_n)] \geq \mathop{\mathbb{E}}_{(a_1,\ldots,a_n)\sim\mu}[u_i(\varphi_i(a_i), a_{-i})] - \varepsilon. \tag{3}$$

Both correlated and coarse correlated equilibria can be interpreted through the use of a trusted third party—a *mediator* or *correlation device*—who samples a joint action profile $(a_1,\ldots,a_n)$ from the correlated distribution μ and then provides the corresponding action a_i to each player $i \in [n]$ as a *recommendation*. From this point of view, a distribution is a CCE or a CE if no player has an incentive to deviate from the recommendation, but for CEs the set of possible deviations is richer: in a CE, a player can decide whether to deviate *after* observing the recommendation, while in a CCE the decision has to be made in advance. This makes a CCE harder to justify in some applications, as one would need some binding mechanism to ensure the player would not be able to deviate after observing the recommendation.

We now go over a concrete example to further elucidate these concepts.

Example 1.3. We consider the "game of chicken." This is a 2×2 game—played between two drivers who are rapidly approaching an intersection from different streets—whose utilities are tabulated in Table 1. Each player can either play "Stop" or "Go." If both players elect to "Go" a crash ensues—a bad outcome for both players. If a player chooses to "Stop" it gets no utility from the game, whereas if it proceeds while the other player chooses to "Stop" it gets a utility of 1 for safely crossing the intersection.

This game has exactly three Nash equilibria: i) (Go, Stop), ii) (Stop, Go), and iii) $\left(\left(\frac{5}{6}, \frac{1}{6}\right), \left(\frac{5}{6}, \frac{1}{6}\right)\right)$, meaning that both players play "Stop" with probability $\frac{5}{6}$. From these three outcomes, the first two are *not* equitable in that they favor one player over the other. The third outcome is even worse: it leads to a crash with some positive probability.

(C)CEs address these issues by unlocking new outcomes. In particular, let's consider the correlated distribution $\frac{1}{2}(\text{Go}, \text{Stop}) + \frac{1}{2}(\text{Stop}, \text{Go})$. It's easy to verify that this is a CE, and thus a CCE. Under that distribution, both players get in expectation a utility of $\frac{1}{2}$. Focusing CEs, there is a natural interpretation of this outcome through a *traffic light*,

which provides a signal to each player. If Player 1 is recommended "Stop," it means that Player 2 will play "Go" with probability 1, so stopping is in Player 1′s interest. On the other hand, if Player 1 is recommended "Go," it means that Player 2 will play "Stop" with probability 1, so crossing is safe for Player 1. That is, in a CE, the signal a player observes updates that player's beliefs concerning the behavior of the other players

	Stop	Go
Stop	0, 0	0, 1
Go	1, 0	−5, −5

Table 1: The game of chicken.

Example 1.4. Our next example clarifies the difference between CCEs and CEs. We consider a 4×4 bimatrix game described with the payoff matrices

$$M_R = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix} \quad \text{and} \quad M_C = \begin{pmatrix} 2 & 0 & 3 & 0 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

for the row and column player, respectively. We label each player's actions as "1," "2," "3," and "4." We claim that the distribution $\mu = \frac{1}{2}(1,1) + \frac{1}{2}(2,2)$ is an exact CCE of this game, whereas the CE gap of μ is large—namely, 1. Specifically, the swap deviation φ that results in a large deviation gain is $1 \mapsto 3$ and $2 \mapsto 4$; the mapping for the rest of the actions is moot, as they are never played under μ. While each player obtains a utility of 2 under μ, deviating per φ gives a utility of 3. At the same time, μ is a CCE as it is robust with respect to constant deviations.

1.2.2 Computational properties

We have seen that CCEs and CEs give rise to new outcomes that are not attainable under independent randomization. Moreover, correlated equilibrium concepts have better computational properties than Nash equilibria. Specifically, a key property is that the set of (C)CEs is convex and can be described through a linear program.

Proposition 1.5. There is a linear program with $\prod_{i=1}^{n} |\mathcal{A}_i|$ variables and $\sum_{i=1}^{n} |\mathcal{A}_i|\ (|\mathcal{A}_i| - 1)$ constraints whose solution is an exact correlated equilibrium of the game.

While the number of swap deviations of each player $i \in [n]$ is $|\mathcal{A}_i|^{|\mathcal{A}_i|}$, it's enough to consider only a certain subset of swap deviations—ones that only change a *single* action; such deviations are called *internal*—with size $|\mathcal{A}_i|\ (|\mathcal{A}_i| - 1)$; the simple proof is left as an exercise. This means that, for normal-form games with a constant number of players, a correlated equilibrium can be computed in polynomial time.

One caveat of this characterization is that the size of the LP grows exponentially with the number of players; the basic reason why this happens is that a correlated distribution in multi-player games is an exponential objective—one needs to specify the value of $\prod_{i=1}^{n}|\mathcal{A}_i| - 1$ coordinates. As we shall see in the sequel, there is an ingenious algorithm for addressing this issue. For now, it is reassuring to know that one can compute a CE in normal-form games with a constant number of players. It is worth noting that one can also incorporate any linear objective function into the linear program, such as the *social welfare*—the sum of the players' utilities.

1.2.3 Connection to no-regret learning

As we have alluded to, (coarse) correlated equilibria are closely tied to the framework of regret minimization in online learning.

Φ-regret. To formalize this connection in its full generality, we will now introduce the important concept of Φ-*regret*. It is a measure of the learner's performance parameterized by a family of *strategy deviations* Φ. For a set of deviations Φ comprising functions $\varphi : \mathcal{X} \to \mathcal{X}$, Φ-regret is defined as

$$\Phi\mathrm{Reg}^{(T)} := \max_{\varphi\in\Phi}\left\{\sum_{t=1}^{T}\left\langle\varphi\left(\boldsymbol{x}^{(t)}\right), \boldsymbol{u}^{(t)}\right\rangle\right\} - \sum_{t=1}^{T}\left\langle\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}\right\rangle. \tag{5}$$

We covered a moment ago the special case where Φ comprises only *constant deviations*: $\Phi_{\text{const}} = \{\varphi : \exists \boldsymbol{x}' \in \mathcal{X} \text{ such that } \varphi(\boldsymbol{x}) = \boldsymbol{x}'\}$; this is indeed the most standard definition of regret in online learning, referred to as *external* regret. The key point about (5) is that the richer the set of deviations Φ, the stronger the induced notion of hindsight rationality. The other end of the spectrum where Φ consists of all possible deviations $\mathcal{X} \to \mathcal{X}$ gives rise to *swap regret*. The next example shows that an algorithm can experience large swap regret even when its external regret is small.

Example 1.6. Let's say the learner picks a distribution over three actions, "1," "2," and "3." Suppose further that the sequence of utilities and selected actions follow the pattern shown in Table 2, where we can assume $T = 0 \bmod 3$. In this example, the learner obtains overall a utility of $\frac{T}{3}$. In fact, this matches the optimal strategy in hindsight. So, the external regret of the learner is 0 in this example. At the same time, consider the swap deviation

$$\varphi : a \mapsto \begin{cases} 2 \text{ if } a = 1, \\ 1 \text{ if } a = 2, \\ 3 \text{ if } a = 3. \end{cases}$$

Under that deviation, the learner would be able to collect maximal utility, implying that the swap regret of the learner is $\Omega(T)$.

1	0	0	1	0	0	1	0	0	1
2	0	1	0	0	1	0	0	1	0
3	1	0	0	1	0	0	1	0	0

Table 2: An example of a learner with large swap regret but zero external regret.

This example shows that external regret—and the associated solution concept of a CCE—is a weak benchmark. In fact, a CCE can be supported on strictly dominated actions [VZ13]! This motivates considering tighter equilibrium concepts revolving around the notion of Φ-regret.

Now, the techniques we have covered so far for minimizing external regret will not be enough to minimize swap regret. For example, multiplicative weights update (MWU) can have linear swap regret [CL06]. As a result, we will need new algorithmic ideas to go beyond external regret and coarse correlated equilibria.

Before we proceed, we formalize the connection between minimizing Φ-regret and correlated equilibrium concepts. We now extend our scope to general *multilinear games*. Here, each player $i \in [n]$ selects a strategy $\boldsymbol{x}_i \in \mathcal{X}_i$ from a convex and compact set $\mathcal{X}_i$, so that for any joint strategy $(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$, the utility can be expressed as $u_i(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) = \langle \boldsymbol{x}_i, \boldsymbol{u}_i(\boldsymbol{x}_{-i}) \rangle$ for some utility vector $\boldsymbol{u}_i$ that does not depend on $\boldsymbol{x}_i$. This is a useful abstraction for encompassing both normal- and extensive-form games, the latter under the so-called sequence-form representation.

Definition 1.7 (Φ-equilibrium)**.** A correlated distribution $\mu \in \Delta(\mathcal{X}_1 \times \cdots \times \mathcal{X}_n)$ is an ε-Φ*-equilibrium* if for any player $i \in [n]$ and deviation function $\varphi_i : \mathcal{X}_i \to \mathcal{X}_i$,

$$\mathop{\mathbb{E}}_{(\boldsymbol{x}_1,...,\boldsymbol{x}_n)\sim\mu} u_i(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \geq \mathop{\mathbb{E}}_{(\boldsymbol{x}_1,...,\boldsymbol{x}_n)\sim\mu} u_i(\varphi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - \varepsilon.$$

Theorem 1.8. Suppose that each player $i \in [n]$ incurs Φ_i-regret $\Phi\mathrm{reg}_i^{(T)}$ under the sequence of utilities $\left(\boldsymbol{u}_i\left(\boldsymbol{x}_{-i}^{(t)}\right)\right)_{t=1}^{T}$. Then the average correlated distribution of play $\mu := \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_1^{(t)} \otimes \cdots \otimes \boldsymbol{x}_n^{(t)}$ is an ε-Φ-equilibrium with $\varepsilon = \frac{1}{T}\max_{1\leq i\leq n} \Phi\mathrm{Reg}_i^{(T)}$.

Above, $\boldsymbol{x}_1^{(t)} \otimes \cdots \otimes \boldsymbol{x}_n^{(t)}$ is the product distribution induced by $\left(\boldsymbol{x}_1^{(t)}, ..., \boldsymbol{x}_n^{(t)}\right)$; $\otimes$ denotes the tensor product. The distribution μ produced by Theorem 1.8 is a *mixture* of T product distributions. Correlation arises by playing multiple iterations of the game. As a special case, Theorem 1.8 implies that players minimizing swap regret converge—in terms of the average correlated distribution of play—to correlated equilibria, whereas external regret is associated with *coarse* correlated equilibria.

Proof. For any player $i \in [n]$, we have

$$\Phi\mathrm{Reg}^{(T)} = \max_{\varphi_i \in \Phi_i} \left\{ \sum_{t=1}^{T} \left\langle \varphi\left(\boldsymbol{x}_i^{(t)}\right), \boldsymbol{u}_i^{(t)} \right\rangle \right\} - \sum_{t=1}^{T} \left\langle \boldsymbol{x}_i^{(t)}, \boldsymbol{u}_i^{(t)} \right\rangle \tag{6}$$

$$= \max_{\varphi_i \in \Phi_i} \left\{ \sum_{t=1}^{T} u_i\left(\varphi_i\left(\boldsymbol{x}_i^{(t)}\right), \boldsymbol{x}_{-i}^{(t)}\right) \right\} - \sum_{t=1}^{T} u_i\left(\boldsymbol{x}_1^{(t)}, ..., \boldsymbol{x}_n^{(t)}\right), \tag{7}$$

by multilinearity. Let $\mu = \frac{1}{T}\sum_{t=1}^{T} \bigotimes_{i=1}^{n} \boldsymbol{x}_i^{(t)}$. Continuing from (6),

$$\frac{1}{T}\Phi\mathrm{Reg}^{(T)} = \max_{\varphi_i \in \Phi_i} \mathop{\mathbb{E}}_{(\boldsymbol{x}_1,...,\boldsymbol{x}_n)\sim\mu} u_i(\varphi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - \mathop{\mathbb{E}}_{(\boldsymbol{x}_1,...,\boldsymbol{x}_n)\sim\mu} u_i(\boldsymbol{x}_1, ..., \boldsymbol{x}_n).$$

□

1.3 A framework for minimizing Φ-regret

Having connected Φ-regret with Φ-equilibria, we now introduce the elegant framework of Gordon, Greenwald and Marks [GGM08] to minimize Φ-regret in the online learning setting; by virtue of Theorem 1.8, one can then compute a Φ-equilibrium by having each player employ such algorithms.

Reducing Φ-regret to external regret. The key idea in the construction of Gordon, Greenwald and Marks [GGM08] is that one can reduce minimizing Φ-regret to minimizing external regret. The framework of Gordon, Greenwald and Marks [GGM08] provides a general template based on two basic subroutines.

1. A *fixed-point oracle*: for any deviation $\varphi \in \Phi$, it outputs a fixed point $\boldsymbol{x} = \varphi(\boldsymbol{x})$.
2. An online algorithm R_Φ minimizing *external regret* with respect to *the set* Φ.

Regarding the fixed-point oracle, we will assume that Φ consists of continuous functions mapping $\mathcal{X}$ to $\mathcal{X}$, so that the *existence* of a fixed point is guaranteed by Brouwer's fixed-point theorem; whether such a fixed point can be computed efficiently is a different story. (As we shall see in the sequel, computing approximate fixed points of general functions is known to be equivalent to finding Nash equilibria [DGP08], which defeats our purpose.) For now, we can assume that Φ is structured enough so that it admits an efficient fixed-point oracle; for example, this is the case when Φ contains only linear deviations. A final point is that it will be enough if one has instead an *approximate* fixed-point oracle, in that $\|\boldsymbol{x} - \varphi(\boldsymbol{x})\| \leq \varepsilon$.

Assuming access to a fixed-point oracle, the reduction of Gordon, Greenwald and Marks [GGM08] reduces Φ-regret to external regret, but with an important catch: the algorithm minimizing external regret needs to *operate over the set of deviations* Φ. This is a significantly more complex set than the one we started with, and will be the critical step in establishing efficient Φ-regret minimizers.

Assuming access to these oracles, the algorithm of Gordon, Greenwald and Marks [GGM08] produces a Φ-regret minimizer R as follows.

- In every time $t \in [T]$, it obtains the next strategy $\varphi^{(t)}$ of R_Φ. R then produces as the next strategy $\boldsymbol{x}^{(t)} \in \mathcal{X}$ any fixed point of $\varphi^{(t)}$ through the fixed-point oracle.
- Next, upon observing $\boldsymbol{u}^{(t)}$, R feeds to R_Φ the utility function $u_\Phi^{(t)} : \varphi \mapsto \left\langle \varphi\left(\boldsymbol{x}^{(t)}\right), \boldsymbol{u}^{(t)} \right\rangle$.

Algorithm [GGM08]: Φ-regret minimizer

Input: An external regret minimizer R_Φ for the set Φ

function `NextStrategy`():

  Set $\varphi^{(t)} := R_\Phi$.`NextStrategy`()

  return a fixed point $\boldsymbol{x}^{(t)} = \varphi^{(t)}\big(\boldsymbol{x}^{(t)}\big)$

function `ObserveUtility`($\boldsymbol{u}^{(t)}$):

  Set $u_\Phi^{(t)} : \varphi \mapsto \big\langle \varphi\big(\boldsymbol{x}^{(t)}\big), \boldsymbol{u}^{(t)} \big\rangle$

  R_Φ.`ObserveUtility`$\Big(u_\Phi^{(t)}\Big)$

Theorem 1.9 ([GGM08])**.** If $\mathrm{Reg}^{(T)}$ is the external regret of R_Φ and $\Phi\mathrm{Reg}^{(T)}$ is the Φ-regret of R, then $\mathrm{Reg}^{(T)} = \Phi\mathrm{Reg}^{(T)}$.

Proof. We have

$$\Phi\mathrm{Reg}^{(T)} = \max_{\varphi\in\Phi}\left\{\sum_{t=1}^{T}\big\langle\varphi\big(\boldsymbol{x}^{(t)}\big), \boldsymbol{u}^{(t)}\big\rangle\right\} - \sum_{t=1}^{T}\big\langle\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}\big\rangle \tag{8}$$

$$= \max_{\varphi\in\Phi}\left\{\sum_{t=1}^{T}\big\langle\varphi\big(\boldsymbol{x}^{(t)}\big), \boldsymbol{u}^{(t)}\big\rangle\right\} - \sum_{t=1}^{T}\big\langle\varphi^{(t)}\big(\boldsymbol{x}^{(t)}\big), \boldsymbol{u}^{(t)}\big\rangle, \tag{9}$$

since $\boldsymbol{x}^{(t)} = \varphi^{(t)}\big(\boldsymbol{x}^{(t)}\big)$. Continuing from (8),

$$\Phi\mathrm{Reg}^{(T)} = \max_{\varphi\in\Phi}\left\{\sum_{t=1}^{T} u_\Phi^{(t)}(\varphi)\right\} - \sum_{t=1}^{T} u_\Phi^{(t)}\big(\varphi^{(t)}\big) = \mathrm{Reg}^{(T)}.$$

□

1.3.1 The algorithm of Blum and Mansour

Finally, we will see how to make use of the previous framework to minimize swap regret when the underlying strategy set is the probability simplex—corresponding to normal-form games. The resulting algorithm was first developed by Blum and Mansour [BM07] (we also refer to a closely related algorithm due to Stoltz and Lugosi [SL05]). As alluded to above, the key to applying the framework of Gordon, Greenwald and Marks [GGM08] is to understand the structure of the set of deviations Φ. In this special case, it is enough to consider only *linear functions* mapping $\Delta(\mathcal{A})$ to $\Delta(\mathcal{A})$, for which there is a simple combinatorial characterization in terms of *(column)-stochastic* matrices.

Lemma 1.10. Any linear function $\varphi : \Delta(\mathcal{A}) \to \Delta(\mathcal{A})$ can be equivalently expressed as $\boldsymbol{x} \mapsto \mathbf{M}\boldsymbol{x}$ for some stochastic matrix $\mathbf{M}$.

In proof, since φ is linear it can be expressed as $\boldsymbol{x} \mapsto \mathbf{M}\boldsymbol{x}$ for some matrix $\mathbf{M}$. Now, every column of $\mathbf{M}$ is equal to the output of φ for the probability distribution that places all the

probability in the corresponding action profile. Since φ maps $\Delta(\mathcal{A})$ to $\Delta(\mathcal{A})$, it follows that every column of $\mathbf{M}$ is a probability distribution.

Armed with the characterization of Lemma 1.10, we now see how to implement the two oracles required in the framework of Gordon, Greenwald and Marks [GGM08]. First, a stochastic matrix induces a Markov chain over $\mathcal{A}$. Any *stationary distribution* of that Markov chain is a fixed point, and can be computed efficiently as it boils down to solving a linear system. (More broadly, when the underlying set $\mathcal{X}$ is a polytope and Φ comprises linear deviations, computing a fixed point amounts to solving a linear program, which can be done in polynomial time.)

The next step is to minimize regret with respect to the set of stochastic matrices. By definition, the set of stochastic matrices is a product of simplices—one probability distribution for each column:

$$\left\{\left[\left(\boldsymbol{x}_a\right)_{a\in\mathcal{A}}\right] : \boldsymbol{x}_a \in \Delta(\mathcal{A}) \quad \forall a \in \mathcal{A}\right\},$$

where, for $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{\mathcal{A}}$, $[(\boldsymbol{x}, \boldsymbol{x}')]$ denotes the matrix with columns x and x'. Minimizing (external) regret over such a set can be accomplished by simply having an independent regret minimizer for each column.

Lemma 1.11. There is an efficient no-regret algorithm for minimizing external regret over the set of stochastic matrices.

The overall construction is given below.

Algorithm: Swap regret minimizer

Input: A regret minimizer R_a for each action $a \in \mathcal{A}$

`NextStrategy`():

  for each action $a \in \mathcal{A}$ **do**

    $\boldsymbol{x}_a^{(t)} \coloneqq R_a$.`NextStrategy`()

  Set $\mathbf{M}^{(t)} \coloneqq \left[\left(\boldsymbol{x}_a^{(t)}\right)_{a\in\mathcal{A}}\right]$

  return a fixed point $\boldsymbol{x}^{(t)} = \mathbf{M}^{(t)}\boldsymbol{x}^{(t)}$

`ObserveUtility`($\boldsymbol{u}^{(t)} \in \mathbb{R}^{\mathcal{A}}$):

  for each action $a \in \mathcal{A}$ **do**

    Set $\boldsymbol{u}_a^{(t)} \coloneqq \boldsymbol{x}^{(t)}[a]\boldsymbol{u}^{(t)}$

    R_a.`ObserveUtility`$\left(\boldsymbol{u}_a^{(t)}\right)$

It consists of $|\mathcal{A}|$ separate regret minimizers, $(R_a)_{a\in\mathcal{A}}$, each of which operates over $\Delta(\mathcal{A})$. To obtain the next strategy, we create the stochastic matrix $\mathbf{M}^{(t)}$ in which each column is given by the strategy of the corresponding regret minimizer, and then output any fixed point of $\mathbf{M}^{(t)}$. To explain the second part of the algorithm, let's first note that the utility observed by R_Φ, per the construction in Theorem 1.9, can be cast as

$$u_\Phi(\varphi) = \left\langle \varphi\left(\boldsymbol{x}^{(t)}\right), \boldsymbol{u}^{(t)} \right\rangle = \left\langle \mathbf{M}\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \right\rangle = \left\langle \mathbf{M}, \boldsymbol{u}^{(t)} \otimes \boldsymbol{x}^{(t)} \right\rangle,$$

where we used that $\varphi\big(\boldsymbol{x}^{(t)}\big) = \mathbf{M}\boldsymbol{x}^{(t)}$. In other words, R_Φ observes the utility vector $\boldsymbol{u}^{(t)} \otimes \boldsymbol{x}^{(t)}$. Moreover, one should forward to each R_a its corresponding component, which is $\boldsymbol{x}^{(t)}[a]\boldsymbol{u}^{(t)}$. If we instantiate each R_a with MWU and invoke Theorem 1.9, we arrive at the following result.

Theorem 1.12 ([BM07])**.** There is an online algorithm whose swap regret is bounded by $O\Big(\sqrt{T\ |\mathcal{A}|\log|\mathcal{A}|}\Big)$.

The naive argument here would only yield a swap regret bound of $O\Big(|\mathcal{A}|\ \sqrt{T\log|\mathcal{A}|}\Big)$ since each MWU algorithm incurs an external regret bounded by $O\Big(\sqrt{T\log|\mathcal{A}|}\Big)$. However, one can make use of the structure of the utilities to obtain the improved bound claimed in Theorem 1.12. In particular, we observe that for any $t \in [T]$,

$$\sum_{a\in\mathcal{A}} \big\|\boldsymbol{u}_a^{(t)}\big\|_\infty^2 = \big\|\boldsymbol{u}^{(t)}\big\|_\infty^2 \sum_{a\in\mathcal{A}} \big(\boldsymbol{x}^{(t)}[a]\big)^2 \le \big\|\boldsymbol{u}^{(t)}\big\|_\infty^2.$$

So, using the regret bound of MWU together with Theorem 1.9,

$$\Phi\mathrm{Reg}^{(T)} \le \frac{|\mathcal{A}|\log|\mathcal{A}|}{\eta} + \eta\sum_{t=1}^{T}\big\|\boldsymbol{u}^{(t)}\big\|_\infty^2 \le \frac{|\mathcal{A}|\log|\mathcal{A}|}{\eta} + \eta T.$$

Optimizing the learning rate η gives the claim.

Bibliography for this chapter

[GGM08] G. J. Gordon, A. Greenwald and C. Marks. (2008). No-regret learning in convex games. *International Conference on Machine Learning*.

[Nas50] J. Nash. (1950). Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences*, 36, 48–49.

[DGP08] C. Daskalakis, P. Goldberg and C. Papadimitriou. (2008). The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing*.

[GZ89] I. Gilboa and E. Zemel. (1989). Nash and Correlated Equilibria: Some Complexity Considerations. *Games and Economic Behavior*, 1, 80–93.

[MV78] H. Moulin and J.-P. Vial. (1978). Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7, 201–221.

[Aum74] R. Aumann. (1974). Subjectivity and Correlation in Randomized Strategies. *Journal of Mathematical Economics*, 1, 67–96.

[VZ13] Y. Viossat and A. Zapechelnyuk. (2013). No-regret dynamics and fictitious play. *Journal of Economic Theory*, 148, 825–842.

[CL06] N. Cesa-Bianchi and G. Lugosi. (2006). Prediction, learning, and games. Cambridge university press.

[BM07] A. Blum and Y. Mansour. (2007). From External to Internal Regret. *Journal of Machine Learning Research*, 8, 1307–1324.

[SL05] G. Stoltz and G. Lugosi. (2005). Internal regret in on-line portfolio selection. *Machine Learning*, 59, 125–159.