No-Regret Learning in Convex Games

Geoffrey J. Gordon GGORDON@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213

Amy Greenwald AMY@CS.BROWN.EDU
Casey Marks CASEY@CS.BROWN.EDU
Department of Computer Science, Brown University, Providence, RI 02912

Abstract

Quite a bit is known about minimizing different kinds of regret in experts problems, and how these regret types relate to types of equilibria in the multiagent setting of repeated matrix games. Much less is known about the possible kinds of regret in online convex programming problems (OCPs), or about equilibria in the analogous multiagent setting of repeated convex games. This gap is unfortunate, since convex games are much more expressive than matrix games, and since many important machine learning problems can be expressed as OCPs. In this paper, we work to close this gap: we analyze a spectrum of regret types which lie between *external* and *swap* regret, along with their corresponding equilibria, which lie between *coarse correlated* and *correlated* equilibrium. We also analyze algorithms for minimizing these regret types. As examples of our framework, we derive algorithms for learning correlated equilibria in polyhedral convex games and extensive-form correlated equilibria in extensive-form games. The former is exponentially more efficient than previous algorithms, and the latter is the first of its type.

1. Introduction

We wish to build agents that can learn to act effectively in multiagent decision problems. We represent such problems as general-sum games: each agent i is given a feasible region A_i from which to choose an action a_i. The payoff to agent i depends not only on i's choice, but also on the actions $a_{\neg i}$ chosen by other agents. Since we are modeling learning, we assume that each agent knows only its own feasible region and observes only its own payoff structure. So, an agent cannot simply compute an equilibrium of the game and play it (even leaving aside the complexity of such a computation and the problem of coordinating with other agents on an equilibrium). All an agent can do is *learn* a preferred course of action by playing the game repeatedly and observing its own payoffs.

What, then, is an appropriate goal for a learning agent? Unlike zero-sum games, general-sum games do not have a well-defined *value*: even if we had complete knowledge of the game and all players were completely rational, we would not be able to predict how much payoff we should receive. Instead, researchers have defined other goals for learning agents. One popular one is regret minimization. For example, a number of previous algorithms have been designed to minimize *external regret* (defined in Sec. 2) in convex games, including Generalized Gradient Descent (Gordon, 1999b), GIGA (Zinkevich, 2003), Follow the Perturbed Leader (Kalai & Vempala, 2003), Lagrangian Hedging (Gordon, 2006), and algorithms based on Fenchel duality (Shalev-Shwartz & Singer, 2006).

However, no external regret may not be a sufficient goal: a set of agents can all achieve no external regret (which guarantees that the empirical distribution of joint play converges to the set of *coarse correlated equilibria*, defined in Sec. 4) and still have an incentive to change their play. For example, a no-external-regret learner can consistently observe that its average payoff per trial would have been higher if it had chosen action a' every time that it actually played a, and yet never switch to playing action a' in these situations. To avoid such behavior, we seek algorithms that provide guarantees stronger than no external regret. In a seminal paper, Foster and Vohra (1997) present an algorithm that exhibits no *internal regret* (defined in Sec. 2) in matrix games, and further, show that if all players achieve no internal regret, the empirical distribution of joint play converges to the set of *correlated*

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

equilibria (see Sec. 4). This guarantee rules out precisely the anomalous behavior described above.

Stoltz and Lugosi (2007) generalize these results to convex games. Extending the framework of Greenwald and Jafari (2003) for matrix games, they define a continuum of regret measures called Φ-regret, as well as corresponding Φ-equilibria, for convex games. Given a feasible region A, Φ is a collection of *action transformations*; that is, each $\phi \in \Phi$ is a function from A to itself. An agent calculates its Φ-regret by comparing the losses it obtained during its past history of play to the losses it would have obtained had it transformed each action it played according to some $\phi \in \Phi$.

Different choices of Φ lead to different types of regret and corresponding equilibria. In matrix games, the only two regret types known to be of interest are the above-mentioned external and internal regret. No internal regret is equivalent to no *swap* regret, in which Φ is the set of all transformations from A to itself. In convex games, by contrast, there is a much richer variety of regret concepts. We identify and analyze two novel regret types, which we call *extensive-form* and *finite-element* regret. We also analyze *linear* regret. Each of these regret types is distinct from the others and from external and swap regret. In fact, they form a progression: no swap regret (the strongest property) implies no finite element regret, which implies no linear regret, which implies no extensive-form regret, which implies no external regret (the weakest property).

Different regret types require different regret-minimization algorithms. For convex games, until recently, most algorithms minimized only external regret. More recently, Stoltz and Lugosi (2007) proved the existence of a no-swap-regret algorithm, and Hazan and Kale (2007) derived an algorithm that exhibits no Φ-regret for any set Φ which is the convex hull of a finite set of transformations. Simultaneously and independently, we developed an algorithm similar to Hazan and Kale's: our algorithm handled more-general representations of transformation sets, but required exact fixed-point calculations (Gordon et al., 2007).

Unfortunately, constructing an algorithm according to Stoltz and Lugosi's proof would be prohibitively expensive: both the time and space requirements would grow exponentially with the number of rounds. And, Hazan and Kale's algorithm, which runs in time polynomial in the number of corners of Φ, can also be prohibitively expensive: for example, if A is the unit cube in $\mathbb{R}^d$ and Φ is the set of linear transformations that map A to itself, then Φ, which is the Cartesian product of d copies of the unit L_1 ball, has $(2d)^d$ corners.

In this work, we extend our earlier algorithms and proofs, unifying them with Hazan and Kale's. The result is an algorithm which accommodates more-efficient representations of Φ. In the example above, the natural representation of Φ is as a set of $d \times d$ matrices satisfying certain linear constraints. Using this representation, our algorithm runs in time polynomial in d—an exponential speedup. In general, we can efficiently achieve no linear regret so long as we can efficiently optimize over the set of linear mappings from A to itself.

We also instantiate our algorithm for extensive-form and finite-element regret. These regret types are important in practice: extensive-form regret corresponds to extensive-form correlated equilibrium (Forges & von Stengel, 2002), arguably the most natural notion of equilibrium in extensive-form games. And, our no-finite-element-regret algorithm, with a simple modification described below, guarantees that the empirical distribution of joint play converges to a correlated equilibrium.

For extensive-form regret, our algorithm is polynomial in the dimension of the action set A; we are not aware of any prior no-extensive-form-regret algorithms. For finite-element regret, our algorithm is polynomial in the dimension of the action set and in the size of a finite-element mesh that covers Φ. Although the necessary mesh for some choices of Φ is quite large, our algorithm is still by far the most efficient known that guarantees convergence to correlated equilibrium.

2. The General Algorithm

When playing a repeated convex game, a single agent's learning problem is called an **online convex program** (OCP): in each round t, the agent chooses an action $a_t \in A$. At the same time, forces external to the agent choose a convex loss function $l_t \in L$. (A loss is just a negative payoff.) The agent observes l_t and pays $l_t(a_t)$. The action space A is assumed to be a convex and compact subset of $\mathbb{R}^d$. The set L includes convex loss functions with bounded subgradients. The commonly studied **experts problem** is a special case of an OCP in which the feasible region is the probability simplex in $\mathbb{R}^d$.

A **learning algorithm** takes as input a sequence of loss functions l_t and produces as output a sequence of actions a_t. Action a_t may depend on $l_1 \dots l_{t-1}$, but not on l_t or later loss functions. The learner's objective is to minimize its cumulative loss, $L_t = \sum_{t=1}^{T} l_t(a_t)$.

The minimum achievable loss depends on the specific sequence l_t. To measure how well a learning algorithm

performs against a given sequence, we calculate its **regret**. The simplest type of regret is called **external** regret, and is defined as follows:

$$\rho_t^{\text{EXT}} = \sup_{a \in A} \sum_{t=1}^{T} (l_t(a_t) - l_t(a))$$

That is, the external regret is the difference between the actual loss achieved and the smallest possible loss that could have been achieved on the sequence l_t by playing a fixed $a \in A$.

We say that an algorithm $\mathcal{A}$ exhibits **no external regret** for feasible region A and set L if we can guarantee that its average external regret per trial eventually falls below any $\epsilon > 0$, regardless of the particular sequence l_t. In other words, $\mathcal{A}$ exhibits no external regret if there is a function $f(T, A, L)$ which is $o(T)$ for any fixed A and L, such that for all $a \in A$, $t \geq 1$

$$\sum_{t=1}^{T} l_t(a_t) \leq \sum_{t=1}^{T} l_t(a) + f(T, A, L) \tag{1}$$

The function f can depend on A and L in complicated ways, but usually depends on properties like the diameter of A under some norm, or the length of $\partial l(a)$ under some norm for $a \in A$ and $l \in L$.

More generally, an agent can consider replacing its sequence $a_1 \dots a_t$ with $\phi(a_1) \dots \phi(a_t)$, where ϕ is some **action transformation**, that is, a measurable function that maps A into itself. If Φ is a set of such action transformations, we define an algorithm's **Φ-regret** as

$$\rho_t^{\Phi} = \sup_{\phi \in \Phi} \sum_{t=1}^{T} (l_t(a_t) - l_t(\phi(a_t)))$$

and we say that it exhibits **no Φ-regret** if it satisfies the following analogue of Eq. 1: for all $\phi \in \Phi$, $t \geq 1$

$$\sum_{t=1}^{T} l_t(a_t) \leq \sum_{t=1}^{T} l_t(\phi(a_t)) + g(T, A, L, \Phi) \tag{2}$$

where $g(T, A, L, \Phi)$ is $o(T)$ for any fixed A, L, and Φ.

Note that external regret is just Φ-regret with Φ equal to the set of constant transformations: i.e., $\Phi_{\text{EXT}} = \{\phi_x \mid x \in A\}$, where $\phi_x(a) = x$. By setting Φ to larger, more flexible transformation sets, we can define stronger varieties of regret. However, before studying any specific regret types in detail, we next discuss how to achieve no Φ-regret for general Φ.

2.1. General Φ

In this section, we develop an algorithm $\mathcal{A}$ that exhibits no Φ-regret for any suitable $\Phi \subset A \mapsto A$. The algorithm itself is fairly simple, and embodies essentially the same idea that was proposed earlier by Gordon et al. (2007) and Hazan and Kale (2007). However, we develop the idea here so that it applies to a more general class of transformation sets Φ than considered previously, and provide a proof that it achieves no Φ-regret under more general conditions. Our extra generality is crucial for developing efficient implementations for important choices of Φ including linear, extensive-form, and finite-element transformations.[1]

Our Φ-regret minimizing algorithm $\mathcal{A}$ is described in Fig. 1. It takes as input a sequence of loss functions $l_t \in L$ and outputs a sequence of actions $a_t \in A$, which, we will show, satisfies Eq. 2.

In designing $\mathcal{A}$, we assume that we have access to subroutines $\mathcal{A}'$ and $\mathcal{A}''$. The subroutine $\mathcal{A}'$ computes approximate fixed points of transformations $\phi \in \Phi$. That is, given any $\phi \in \Phi$ and any $\epsilon > 0$, $\mathcal{A}'$ returns some $a \in A$ such that $\|a - \phi(a)\|_A \leq \epsilon$. Here, $\|\cdot\|_A$ is an arbitrary norm on $\mathbb{R}^d$. The subroutine $\mathcal{A}''$ is an external-regret minimizing algorithm whose feasible region is Φ; we assume that its regret bound is $o(T)$ whenever we can provide a bound (in an appropriate norm) on the subgradients of the loss functions it encounters.

Since algorithm $\mathcal{A}$ accesses the transformation set Φ only through the subroutines $\mathcal{A}'$ and $\mathcal{A}''$, it does not depend on any special properties of Φ beyond the existence of these subroutines. To state our theorem, though, we will embed Φ in a vector space, as follows. Since $A \subset \mathbb{R}^d$, we can write $\phi \in \Phi$ as a d-tuple of "coordinate" functions $(\psi_1, \psi_2, \dots, \psi_d)$, $\psi_i : A \to \mathbb{R}$. For all $\phi \in \Phi$ and $i = 1 \dots d$, we assume ψ_i is a member of some reproducing-kernel Hilbert space (RKHS) $\mathcal{H} \subset A \mapsto \mathbb{R}$.[2] Finally, we assume that Φ is a convex and compact subset of $\mathcal{H}^d$.

To make these assumptions concrete, suppose for example that Φ is the convex hull of a finite set of transformations $\{\phi^1, \dots, \phi^p\}$: i.e.,

$$\Phi = \left\{ \textstyle\sum_{j=1}^{p} \alpha_j \phi^j \mid \alpha_j \geq 0, \sum_{j=1}^{p} \alpha_j = 1 \right\}$$

(This is the case treated by Hazan and Kale.) If we take $\mathcal{H}$ to be the span of all of the coordinate functions ψ_i^j, then Φ is a simplex in $\mathcal{H}^d$ with corners ϕ^j, for $j = 1 \dots p$. (In general, Φ's shape may be much more

[1] Hazan and Kale's algorithm is efficient in the special case of external transformations. Indeed, this section's algorithm specializes to their algorithm in this case.

[2] A Hilbert space is a (possibly infinite-dimensional) vector space that has an inner product. A reproducing-kernel Hilbert space is a Hilbert space of real- or complex-valued functions in which evaluation at the point a is a continuous linear functional for any a.

Given feasible region A, transformation set Φ, initial transformation $\phi_1 \in \Phi$, and subroutines $\mathcal{A}'$ and $\mathcal{A}''$.

For $t = 1, \ldots, T$:

1. Send transformation ϕ_t to the fixed-point algorithm $\mathcal{A}'$, along with accuracy parameter $\epsilon_t = 1/\sqrt{t}$. Receive action a_t satisfying $\|\phi_t(a_t) - a_t\|_A \leq \epsilon_t$.
2. Play a_t; observe loss function l_t and incur loss $l_t(a_t)$.
3. Define $m_t : \Phi \mapsto \mathbb{R}$ by $m_t(\phi) = l_t(\phi(a_t))$.
4. Send m_t to the no-external-regret algorithm $\mathcal{A}''$. Receive transformation $\phi_{t+1} \in \Phi$.

Figure 1. Algorithm $\mathcal{A}$.

complicated than a simplex, as we will see for example in the definition of Φ_{FE} below.)

To bound the Φ-regret of algorithm $\mathcal{A}$, we will need bounds on the actions a and the loss-function subgradients $\partial l(a)$, for all $l \in L$ and $a \in A$. In particular, we will suppose that $\|a\|_A \leq C_1$ and $\|\partial l(a)\|_{A^*} \leq C_2$, for any $a \in A$, any $l \in L$, and some constants $C_1, C_2 > 0$. Here $\|\cdot\|_{A^*}$ is the norm that is dual to $\|\cdot\|_A$.

Theorem 1 *Fix a convex and compact feasible region A and a set of loss functions L satisfying the above norm bounds, as well as a set of transformations $\Phi \subset \mathcal{H}^d$, where $\mathcal{H} \subset A \mapsto \mathbb{R}$ is a RKHS. Assume we are given an algorithm $\mathcal{A}''$ which, for any set of possible loss functions M with bounded subgradients, achieves no external regret on Φ. Also assume we are given an algorithm $\mathcal{A}'$ which can compute an approximate fixed point of any $\phi \in \Phi$. Then algorithm $\mathcal{A}$, using subroutines $\mathcal{A}'$ and $\mathcal{A}''$, achieves no Φ-regret.*

PROOF: Define the set of functions $M \subset \Phi \mapsto \mathbb{R}$ as $M = \{l(\phi(a)) \mid l \in L, a \in A\}$. Note that each $m \in M$ is convex because each $l \in L$ is convex and $\phi(a)$ is linear in ϕ. Moreover, the norm of the subgradient of any $m \in M$ at any point $\phi \in \Phi$ is bounded by $C_1 C_2$. (A proof of this fact, as well as a definition of the appropriate norm, is given by Gordon et al. (2008).)

Because $\mathcal{A}''$ exhibits no external regret on Φ with the bounded-subgradient set of potential loss functions M,

$$\sum_{t=1}^{T} m_t(\phi_t) \leq \sum_{t=1}^{T} m_t(\phi) + f(T, \Phi, M) \quad \forall \phi \in \Phi$$

where f is sublinear in T. So, by the definition of m_t,

$$\sum_{t=1}^{T} l_t(\phi_t(a_t)) \leq \sum_{t=1}^{T} l_t(\phi(a_t)) + f(T, \Phi, M) \quad \forall \phi \in \Phi$$

But, since $\|\phi_t(a_t) - a_t\|_A \leq \epsilon_t$ and $\|\partial l_t(a_t)\|_{A^*} \leq C_2$, we have by Hölder's inequality that $l_t(a_t) \leq l_t(\phi_t(a_t)) + \epsilon_t C_2$. So,

$$\sum_{t=1}^{T} l_t(a_t) \leq \sum_{t=1}^{T} (l_t(\phi(a_t)) + \epsilon_t C_2) + f(T, \Phi, M) \quad \forall \phi \in \Phi$$

Since $C_2 \sum_{t=1}^{T} \epsilon_t = O(\sqrt{T})$, this is exactly the desired no-Φ-regret guarantee. □

Clearly, the run-time of $\mathcal{A}$ depends on the run-times of its subroutines. In particular, since $\mathcal{A}$ requires that $\mathcal{A}'$'s accuracy parameter ϵ approach 0 as T increases, it is important that $\mathcal{A}'$ run efficiently even for small ϵ. We will discuss run-times in more detail in the context of specific examples below. For now, we note the following trivial generalization of a result due to Hazan and Kale: if the fixed-point algorithm $\mathcal{A}'$ is a FPTAS, and if the no-external-regret algorithm $\mathcal{A}''$ runs in polynomial time, then $\mathcal{A}$ can process T actions and loss functions in time polynomial in T. Hazan and Kale allow run-times to be polynomial in the number of corners of Φ (among other parameters); this renders their efficiency guarantees meaningless when Φ has many corners. With our more-efficient representations of Φ, we can replace the dependence on the number of corners with parameters like the dimension of Φ and the norm bounds for $a \in A$ and ∂l for $l \in L$; since these latter parameters can be much smaller, the result will be a much faster run-time.

As described so far, the algorithm $\mathcal{A}$ is deterministic if its subroutines $\mathcal{A}'$ and $\mathcal{A}''$ are. Below, we will also define a randomized variant of $\mathcal{A}$, to strengthen the connection to game-theoretic equilibria.

2.2. Finite-dimensional Φ

We defined algorithm $\mathcal{A}$ in terms of a generic set of transformations $\Phi \subset \mathcal{H}^d$, where $\mathcal{H}$ is a RKHS, and each element of $\mathcal{H}$ is a real-valued function on A. (So, each $\phi \in \Phi$ is a d-tuple of real-valued functions on A, which we interpret as a function from A to $\mathbb{R}^d$.)

Because of the reproducing-kernel property, computing component $\psi_i(a)$ of some $\phi \in \mathcal{H}^d$ for $a \in A$ is the same as computing the inner product $\langle \psi_i, K(a) \rangle$. In other words, each ψ_i is the composition of a fixed, possibly-nonlinear function $K(\cdot)$ with a linear mapping $\langle \psi_i, \cdot \rangle$. This is the so-called "kernel trick" (Cortes & Vapnik, 1995): first, K computes a vector of features of the action a. The inner product with ψ_i then combines all of these features to produce the final output $\psi_i(a)$. To evaluate $\phi(a)$ in its entirety, we can compute $K(a)$ once, and then evaluate the d inner products $\langle \psi_1, K(a) \rangle, \ldots, \langle \psi_d, K(a) \rangle$.

In this paper, we are chiefly interested in cases where the dimension of $\mathcal{H}$ is manageable, so that we can di-

rectly write down and work with the transformations $\phi \in \mathcal{H}^d$. So, for the remainder of the paper, we will assume that $\mathcal{H}$ is isomorphic to $\mathbb{R}^p$ for some finite p. We will also restrict our interest to linear loss functions $l_t(a) = a \cdot \partial l_t$. This is without loss of generality, since we can achieve no regret for a sequence of convex loss functions l_t by working with appropriately-chosen linear lower bounds on each l_t (Gordon, 1999a).

With these additional assumptions, the steps of $\mathcal{A}$ can be simplified: each derived loss function m_t is linear, and can be described by its subgradient as follows:

$$\partial m_t(\phi) = \partial(l_t(\phi(a_t))) = \partial(\phi(a_t) \cdot \partial l_t) = \partial l_t K(a_t)^{\mathrm{T}}$$

The subgradient ∂m_t is a $d \times p$ matrix, since ∂l_t is a d-vector and $K(a_t)$ is a p-vector. Each transformation ϕ also corresponds to a $d \times p$ matrix (a d-tuple of p-vectors). To evaluate the loss function m_t on a transformation ϕ, we take the dot product $\partial m_t \cdot \phi$, which is defined to be $\mathrm{tr}(\partial {m_t}^{\mathrm{T}} \phi) = \mathrm{tr}(K(a_t) \partial {l_t}^{\mathrm{T}} \phi) = \mathrm{tr}(\partial {l_t}^{\mathrm{T}} \phi K(a_t)) = \partial {l_t}^{\mathrm{T}} \phi K(a_t)$.

As we will see in the next section, a number of interesting transformation sets can be represented as $d \times p$ matrices. Representing transformations and subgradients in this way means we can manipulate them efficiently, and, in turn, design efficient no-regret algorithms.

3. Specific Algorithms

We now instantiate our algorithm with various transformation sets Φ. We define each Φ as a set of $d \times p$ matrices ϕ, together with a kernel function $K : A \mapsto \mathbb{R}^p$, with the guarantee that $\phi K(a) \in A$ for all $a \in A$ and $\phi \in \Phi$. To minimize each ensuing regret type, we go on to identify efficient subroutines $\mathcal{A}'$ and $\mathcal{A}''$ for finding fixed points and achieving no external regret. (All other calculations in our algorithm are $O(pd)$, so these subroutines will usually be what limits our efficiency.)

For completeness, we also mention Φ_{EXT}, the set of constant transformations on A, and Φ_{SWAP}, the set of all measurable transformations on A. Φ_{EXT} is the weakest form of regret of interest here, and Φ_{SWAP} the strongest. These are the only two regret types known to be of interest in matrix games (no swap regret and no internal regret are equivalent in this setting).

In convex games, however, there is a much richer variety of interesting regret concepts. Below, we analyze linear, finite-element, and extensive-form regret, corresponding to transformation sets Φ_{LIN}, Φ_{FE}, and Φ_{EF}. As we will see, in general, $\Phi_{\mathrm{EXT}} \subset \Phi_{\mathrm{EF}} \subset \Phi_{\mathrm{LIN}} \subset \Phi_{\mathrm{FE}} \subset \Phi_{\mathrm{SWAP}}$. So, no swap regret implies no finite-element regret, which implies no linear regret, which implies no extensive-form regret, which implies no external regret. We show in the long version of this paper (Gordon et al., 2008) that these five regret varieties are in fact distinct: it is possible to have, e.g., no Φ_{LIN}-regret while still having positive Φ_{FE}-regret.

Linear Regret The set Φ_{LIN} includes all linear transformations that map A into itself. To achieve no linear regret, we can take K to be the identity. Φ will then be a set of square $d \times d$ matrices. To find a fixed point of $\phi \in \Phi$, we choose an appropriate element of the null space of $\phi - I$, which takes time polynomial in d. The more expensive task is to achieve no external regret on Φ: depending on the form of A, Φ may or may not lend itself to a description in terms of a small number of simple constraints.

If A is a probability simplex, then Φ is the set of stochastic matrices, which can be expressed with $O(d^2)$ linear constraints on the entries of ϕ (this setting yields an algorithm very similar to that of Blum and Mansour (2005)). If A is a unit Euclidean ball, then Φ consists of those matrices whose largest singular value is ≤ 1; this set can be represented using a single semidefinite constraint. For general (convex compact) A, the best choice may be to use either GIGA or lazy projection (Zinkevich, 2003): the difficult step in these algorithms is a Euclidean projection onto Φ, which can be achieved via the ellipsoid algorithm.

Finite-Element Regret The finite-element transformations only apply to polyhedral feasible regions A. For finite-element regret, we will define K as a mapping from a polyhedral feasible set A to a high-dimensional space $K(A)$ called the **barycentric coordinate space**. To construct $K(a)$, we first associate each of the p corners of A with one dimension of $\mathbb{R}^p$. We then triangulate A by dividing it into mutually exclusive and exhaustive d-simplices, so that each corner of A is a corner of one or more simplices.

Now, to calculate $K(a)$, we first determine the simplex in which a lies (or choose one arbitrarily if it is on a boundary) and calculate the weights of a with respect to the $d + 1$ corners of that simplex. That is, if $j(1) \ldots j(d+1)$ are the indices of the corners of the simplex containing a, and if $c_{j(1)} \cdots c_{j(d+1)}$ are their coordinates, we find the weights $b_1 \ldots b_{d+1}$ by solving $a = \sum_i b_i c_{j(i)}$, $\sum_i b_i = 1$. We then set entry $[K(a)]_{j(i)} = b_i$ for each corner $j(i)$, and set all other entries of $K(a)$ to 0.

For example, if $A = [0,1]^2$, we can divide A into two triangles, one with corners $(0,0)$, $(0,1)$, and $(1,1)$, and the other with corners $(0,0)$, $(1,0)$, and $(1,1)$. To calculate $K(\frac{1}{3}, \frac{2}{3})$, note that $(\frac{1}{3}, \frac{2}{3})$ is in the first

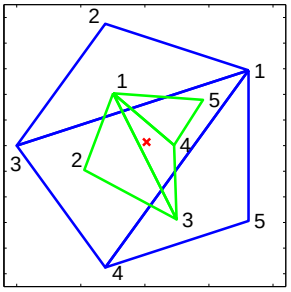


Figure 2. Illustration of barycentric coordinates and Φ_{FE}. A is the outer pentagon, triangulated into three simplices. $K(A)$ is a subset of the simplex in $\mathbb{R}^5$ (not shown). $\phi(A)$ is the distorted pentagon. The $\times$ marks a fixed point of ϕ.

triangle. If we associate corners of A with dimensions of $K(A)$ in the order $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$, then $K(\frac{1}{3},\frac{2}{3}) = (\frac{1}{3},\frac{1}{3},0,\frac{1}{3})$, since these weights express $(\frac{1}{3},\frac{2}{3})$ as a convex combination of corners 1, 2, and 4.

Given this definition of K, Φ_{FE} is the set of matrices ϕ that map $K(A)$ into A. If A is a simplex, then K will be a linear mapping and $\Phi_{\text{FE}} = \Phi_{\text{LIN}}$. (In general, $\Phi_{\text{FE}} \supset \Phi_{\text{LIN}}$.) For another example, see Fig. 2.

We note that Φ_{FE} can be factored: it is the Cartesian product of p copies of A, since it just needs to map each corner of A to a point inside A. So, to achieve no external regret in Φ, we can separately run p copies of any no-external-regret algorithm for A. A typical cost for doing so might be $O(pd^3)$.[3] To find a fixed point of ϕ, we just need to check each of its linear pieces separately. Each individual check costs $O(d^3)$, and there is one for each simplex in our mesh.

Extensive-Form Regret Let T be a player's **sequence tree**, describing all possible sequences of choices and observations in an extensive-form game (e.g., Fig. 3 (left)). Suppose that each element of the feasible region A is a **sequence weight vector** on T (Forges & von Stengel, 2002), specifying a behavior strategy for the game. Define an **extensive form** transformation as follows: fix a set D of choice nodes in T, along with pure-strategy sequence weight vectors w_b for each $b \in D$. If the original strategy is ever about to play $b \in D$, the transformed strategy deviates, and instead follows w_b. We assume that there are no $b, b' \in D$ with b' an ancestor of b (so that all $b \in D$ are reachable), and that each $b \in D$ has a sibling a with $w_b(a) = 1$. Extensive-form transformations are interesting since they correspond to the incentive constraints in extensive-form correlated equilibrium (Forges & von Stengel, 2002).

We show (Gordon et al., 2008) that each extensive form transformation can be represented by a matrix ϕ, whose rows and columns are indexed by choices, so that any action $w \in A$ is transformed into another action $\phi w \in A$. The entries of ϕ are as follows:

$$\phi_{ab} = \begin{cases} w_b(a) & \text{if } b \preceq a \text{ and } b \in D \\ 1 & \text{if } b = a \text{ and } \forall b' \in D,\ b \notin T_{b'} \\ 0 & \text{otherwise} \end{cases}$$

($T_{b'}$ is the subtree of T rooted at b', so that $b \notin T_{b'}$ means b is not a descendent of b'; $b \preceq a$ means b is an ancestor or a sibling of an ancestor of a in T.) This equation says that column b of ϕ is either: a copy of w_b with entries $w_b(a)$ replaced by 0s for $b \not\preceq a$ (if $b \in D$, cases 1, 3); a single 1 on the diagonal (if neither b nor any of its ancestors is in D, cases 2, 3); or all 0s (if $b \notin D$, but one of b's (strict) ancestors is in D, case 3).

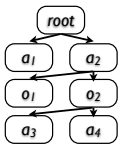
Now, if we take Φ_{EF} to be the convex hull of all such ϕs, then $\Phi_{\text{EF}} \subset \Phi_{\text{LIN}}$, and no Φ_{EF}-regret immediately implies no regret vs. any extensive form transformation. (So, no Φ_{EF}-regret is related to extensive-form correlated equilibrium; see Sec. 4).

For example, if T is as shown in Fig. 3 (left), elements of A are vectors of 4 sequence weights, one each for $a_1 \ldots a_4$. The weight for, e.g., a_3 is $P(a_2 \mid \text{root})P(a_3 \mid o_2)$, the product of the conditional probabilities of all choice nodes along the path from the root to a_3. So, strategy a_1, a_3 yields weights $w = (1,0,0,0)^{\text{T}}$, while a_2, a_3 yields $w' = (0,1,1,0)^{\text{T}}$.

The set Φ_{EF} for this game is shown in Fig. 3 (right). The parameters a, d, e, and f determine the probability that each choice node is included in D: $a \geq 0$ is $P(a_1 \in D)$, $d \geq 0$ is $P(a_2 \in D)$, $e \geq 0$ is $P(a_3 \in D)$, and $f \geq 0$ is $P(a_4 \in D)$. If $a_1 \in D$, parameters b and c specify a strategy for the subtree rooted at a_2. (If $a_1 \notin D$, the game ends right after we reach D, and so we need not specify further choices.) The inequalities listed in Fig. 3 are consistency constraints: e.g., the events $a_2 \in D$ and $a_3 \in D$ are mutually exclusive, so we must have $d + e \leq 1$.

To represent the transformation "play a_2, a_3 instead of a_1," we construct a matrix ϕ by setting $a, b = 1$ and $c, d, e, f = 0$. It is easy to verify that $\phi w = w'$ as expected. On the other hand, the transformation "play a_1 instead of a_2" corresponds to ψ with $d = 1$ and $a, b, c, e, f = 0$; again, it is easy to check $\psi w' = w$.

[3] The precise cost will depend heavily on the shape of A. For general A, most no-external-regret algorithms have a step like solving an LP with feasible region A or projecting onto A by minimum Euclidean distance. These computations cost $O(d^3)$ if we assume that an appropriate measure of the complexity of A is held constant.



$$\begin{bmatrix} 1-a & d & 0 & 0 \\ a & 1-d & 0 & 0 \\ b & 0 & 1-d-e & f \\ c & 0 & e & 1-d-f \end{bmatrix}$$

Figure 3. Φ_{EF} example. $b + c = a$, $d + e \leq 1$, $d + f \leq 1$, and $0 \leq a, b, c, d, e, f \leq 1$.

4. Regret and Equilibria

Algorithm $\mathcal{A}$ achieves no Φ-regret in an online convex program, for any suitable Φ. In this section, we relate this guarantee back to equilibria in convex games.

A game consists of a set of players N, a set of actions A_i for each player $i \in N$, and a payoff function $r_i : \otimes_i A_i \to \mathbb{R}$ for each player $i \in N$. A **matrix** game is one in which each action set is finite. A variant on a matrix game is an **experts** game in which each action set is a probability simplex. Generalizing experts games, a **convex** game is one in which each action set is a convex and compact subset of Euclidean space and each payoff function is multi-linear. In experts games and convex games, players can play interior points; but, assuming polyhedral action sets (PAS), we can generate a corresponding **corner game** by restricting each player's actions to the corners of its action sets.

Following Stoltz and Lugosi (2007), who generalize the definition for matrix games given in Greenwald and Jafari (2003), we define equilibria in convex games in terms of transformation sets.

Definition 2 *Given a game and a collection of transformation sets, $\langle \Phi_i \rangle_{i \in N}$, with each $\Phi_i \subseteq \Phi_{SWAP}$, a probability distribution q over $\otimes_i A_i$ is a $\langle \Phi_i \rangle_{i \in N}$-*
equilibrium *iff the expectation over $a \sim q$ satisfies*

$$\mathbb{E}\left[r_i(\phi(a_i), a_{\neg i}) - r_i(a)\right] \leq 0 \quad \forall i \in N,\ \phi \in \Phi_i \qquad (3)$$

Intuitively, an equilibrium is a distribution from which no player prefers to deviate using any transformation in its set. Taking each Φ_i to be the set of swap transformations defines **correlated equilibria**; taking each Φ_i to be the set of external (i.e., constant) transformations defines **coarse correlated equilibria**. These definitions lead to the following propositions, proved by Marks (2008) and Gordon et al. (2007).

Proposition 3 *A correlated equilibrium of the corner game generated from a PAS convex game is also a correlated equilibrium of the convex game itself.*

Proposition 4 *For every correlated equlibrium in a PAS convex game, the corresponding corner game has a payoff-equivalent correlated equilibrium.*

4.1. Repeated Games

As described above, we assume the agents play some game repeatedly and learn by observing the relationship between their actions and their payoffs. In repeated matrix games, Greenwald and Jafari (2003) have shown that if each agent plays according to a no Φ_i-regret algorithm, then the empirical distribution of joint play converges to the set of $\langle \Phi_i \rangle_{i \in N}$-equilibria with probability 1. The **empirical distribution of joint play** at step t is the following distribution over the joint action set, where $a^t \in \otimes_i A_i$ is the joint action played at time step t: $z^t(\alpha) = \left|\{\tau \mid a^\tau = \alpha\}\right|/t$. The analogous result holds for $\langle \Phi_i \rangle_{i \in N}$-equilibrium in repeated convex games (e.g., Stoltz and Lugosi (2007)).

Because extensive-form games are one class of convex games (Forges & von Stengel, 2002), this result implies that, if the agents all play extensive-form regret-minimization algorithms, their play will converge to the set of extensive-form correlated equilibria. (Marks (2008) also provides algorithms with this property, using the less-efficient normal-form representation of extensive-form games.)

We can also say something about convergence to full-fledged correlated equilibria in repeated convex games: define a **randomized** variant of $\mathcal{A}$ as follows. On a trial where the deterministic algorithm would have played $\bar{a}_t$, the randomized algorithm samples its play a_t from any distribution D such that

$$E_D(a_t) = \bar{a}_t \qquad E_D(K(a_t)) = K(\bar{a}_t) \qquad (4)$$

(We still use $\bar{a}_t$, rather than a_t, in constructing m_t.) With such a D, if loss functions are linear, our Φ-regret on A and external regret on Φ differ by a zero-mean random variable; so, we can use standard stochastic convergence results to prove:

Corollary 5 *Under the conditions of Thm. 1, the additional assumption (4), and restricting L to include only linear loss functions, the randomized variant of $\mathcal{A}$ achieves no Φ-regret with probability 1.*

For Φ_{FE}-regret, we can always find a D that satisfies Equation (4); so (Gordon et al., 2007):

Corollary 6 *If, in a repeated PAS convex game, each agent plays only corner points and uses an algorithm that achieves no internal regret for the corner game (such as the randomized version of $\mathcal{A}$ with $\Phi = \Phi_{\text{FE}}$), then the empirical distribution of joint play converges to the set of correlated equilibria of the convex game with probability 1.*

To our knowledge, ours is the most efficient algorithm which can make this claim, by a factor which is exponential in the dimension d.

5. Discussion

We have presented several new forms of regret for online convex programs, analyzed their relationships to one another and to known regret types, and given the first efficient algorithms that directly minimize some of these forms of regret. These algorithms are by far the most efficient known for several purposes, including guaranteeing convergence to a correlated equilibrium in a repeated convex game, and to an extensive-form correlated equilibrium in an extensive-form game. By contrast, most previous OCP algorithms only guarantee convergence to coarse correlated equilibrium, an outcome which may yield much lower payoffs and may leave incentives for rational agents to deviate.

In the process of designing our algorithms, we derived efficient representations of the transformation sets for each of our regret types except Φ_{SWAP}: we wrote each as a nonlinear kernel mapping followed by a linear transformation chosen from an appropriate set of matrices. These representations may be of separate interest for designing future algorithms. In this paper, we were chiefly interested in cases where the dimension of the kernel mapping was manageable, so that we could directly work with the transformation matrices. However, it would be very interesting to try to design "kernelized" no-Φ-regret algorithms. In such algorithms we would never explicitly write down a transformation ϕ, but instead represent it in terms of observed actions and loss functions, thereby making it feasible to use very high-dimensional sets of transformations.

Important application areas for OCPs and convex games include multi-agent planning (in which the feasible region for each player is a set of plans, and interactions include contending for resources) and learning in extensive-form games such as poker. We are particularly interested in extensive-form games; this application requires further developments such as learning efficiently from bandit feedback and abstracting large games into smaller representations which we can work with in real time.

Acknowledgments

The authors would like to thank Martin Zinkevich for very helpful discussions during an early phase of this work. This work was supported in part by a grant from DARPA's Computer Science Study Panel program and in part by a grant from the Sloan Foundation.

References

Blum, A., & Mansour, Y. (2005). From external to internal regret. *Proceedings of the Conference on Computational Learning Theory (COLT)*.

Cortes, C., & Vapnik, V. N. (1995). Support-vector networks. *Machine Learning Journal*, *20*, 273–297.

Forges, F., & von Stengel, B. (2002). *Computationally efficient coordination in game trees* (Technical Report LSE-CDAM-2002-02). London School of Economics and Political Science, Centre for Discrete and Applicable Mathematics.

Foster, D., & Vohra, R. (1997). Regret in the on-line decision problem. *Games and Economic Behavior*, *21*, 40–55.

Gordon, G., Greenwald, A., & Marks, C. (2008). *No-regret learning in convex games* (Technical Report CS-08-03). Brown University, Department of Computer Science.

Gordon, G., Greenwald, A., Marks, C., & Zinkevich, M. (2007). *No-regret learning in convex games* (Technical Report CS-07-10). Brown University, Department of Computer Science.

Gordon, G. J. (1999a). *Approximate solutions to Markov decision processes*. Doctoral dissertation, Carnegie Mellon University.

Gordon, G. J. (1999b). Regret bounds for prediction problems. *Proceedings of the ACM Conference on Computational Learning Theory*.

Gordon, G. J. (2006). No-regret algorithms for online convex programs. *Advances in Neural Information Processing Systems (NIPS), 19*.

Greenwald, A., & Jafari, A. (2003). A general class of no-regret algorithms and game-theoretic equilibria. *Proceedings of the 2003 Computational Learning Theory Conference* (pp. 1–11).

Hazan, E., & Kale, S. (2007). Computational equivalence of fixed points and no regret algorithms, and convergence to equilibria. *Advances in Neural Information Processing Systems (NIPS), 20*.

Kalai, A., & Vempala, S. (2003). Efficient algorithms for online decision problems. *Proceedings of the 16th Annual Conference on Learning Theory*.

Marks, C. (2008). No-regret learning and game-theoretic equilibria. Ph.D. Dissertation, Department of Computer Science, Brown University, Providence, RI.

Shalev-Shwartz, S., & Singer, Y. (2006). Convex repeated games and Fenchel duality. *Advances in Neural Information Processing Systems (NIPS), 19*.

Stoltz, G., & Lugosi, G. (2007). Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, *59*, 187–208.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the 20th International Conference on Machine Learning*.