No-Regret Learning in Convex Games

Geoff Gordon, Amy Greenwald, Casey Marks and
Martin Zinkevich

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-07-10
October 2007

No-Regret Learning in Convex Games
Brown University Technical Report CS-07-10

Geoffrey J. Gordon
Department of Machine Learning
Carnegie-Mellon University
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

Amy Greenwald
Department of Computer Science
Brown University
Providence, RI 02912
amy@cs.brown.edu

Casey Marks
Department of Computer Science
Brown University
Providence, RI 02912
casey@cs.brown.edu

Martin Zinkevich
Yahoo!, Inc.
Sunnyvale, CA 94089
maz@yahoo-inc.com

Abstract

Quite a bit is known about minimizing different kinds of regret in experts problems, and how these results translate into statements about different kinds 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 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 modern machine learning algorithms can be expressed as convex programs. In this paper, we work to close this gap: in addition to *external* regret (which is the focus of most previous OCP algorithms, but leads to only a weak form of equilibrium) and *swap* regret (which leads to the stronger notion of correlated equilibrium, but does not yet have efficient algorithms), we define three alternative types of regret for OCPs. We analyze these regret types and their corresponding equilibria, showing relationships among them and to known types of regret and equilibrium. And, we provide new algorithms for minimizing the most important of these forms of regret, leading to the most efficient known learning method that converges to correlated equilibria in convex games.

1 Introduction

We wish to design algorithms that can learn to act effectively in multiagent decision-making problems. We represent these decision problems as general-sum convex games: agent i is given a feasible region $A_i \subset \mathbb{R}^d$ from which to choose an action $a_i \in A_i$. The payoff of agent i depends on the actions $a_{\neg i}$ chosen by other agents, and is assumed to be a linear[1] function of a_i when $a_{\neg i}$ is held fixed.

Without loss of generality, we can take A_i to be convex: if it is not, we may interpret points in the convex hull of A_i as distributions over feasible actions. In fact, A is often polyhedral, with corners representing pure actions and internal points representing mixed actions. In this case, the convex game is a matrix game: if A has n corners, then it is a matrix game with n actions. Still, since often $n \gg d$, for reasons of efficiency it is preferable to work with the convex game representation.

[1] Our algorithms also work if the payoff is a concave function of a_i; see Sec. 5.2 for more details.

Since we are interested in learning (as opposed to planning), we do not assume that the agents know the feasible regions or payoff functions of other agents. So, an agent cannot just compute an equilibrium of the game and play it (even leaving aside the problem of coordinating with other agents on which equilibrium). All an agent can do is *learn* how to play the game by playing it repeatedly and observing its own payoffs.

What, then, is an appropriate goal for a learning agent? Since our game is general-sum, there may be no fixed level of payoff that is reasonable to expect against all opponents. So, researchers have attempted to define other reasonable goals for learning agents. One popular goal is low *external regret* (defined in Sec. 2); a number of previous algorithms have been designed to achieve this goal, including Generalized Gradient Descent [1], GIGA [2], Follow the Perturbed Leader [3], Lagrangian Hedging [4], and algorithms based on Fenchel duality [5].

However, if we want our learners to act rationally in general-sum games, low external regret may not be the right goal to aim for: the agents may all achieve zero external regret and yet still have strictly positive incentives to change their plays. That is, with a no-external-regret learner, an agent may consistently observe that its average payoff per trial would have been at least a constant amount higher if it had chosen action a every time that its learning algorithm recommended a', and yet never switch to playing action a in these situations.[2]

To avoid these kinds of embarrassing departures from rationality, our learning algorithms should provide guarantees stronger than no external regret. For example, in matrix games, learners which achieve no *internal regret* (defined in Sec. 2) can guarantee that the empirical distribution of play approaches the set of *correlated equilibria* (Sec. 5), which rules out exactly the situation described above. For convex games, Stoltz and Lugosi [6] prove the existence of an algorithm that achieves no *swap* regret (Sec. 2) and ensures convergence to the set of correlated equilibria. Or, we can obtain convergence to correlated equilibrium via the naïve algorithm which ignores the convex game structure and learns as if it were playing the corresponding large matrix game. (We call this trick the **corners-as-experts** reduction.) Unfortunately, both of these algorithms are inefficient, as detailed below.

To try to find more efficient learning methods that still have strong guarantees, we define and analyze several forms of regret which lie between external and swap regret in OCPs: in order of increasing strength, they are linear regret, finite-element regret, and σ-regret. Even though finite-element regret is weaker than swap regret, we show that minimizing finite-element regret is sufficient to reach correlated equilibrium if agents always play corners of their feasible regions.

We also present new, efficient algorithms which learn to minimize linear or finite-element regret. The finite-element regret minimizer is particularly interesting since it is faster than either the Stoltz and Lugosi algorithm or the corners-as-experts algorithm, but can still be used to find correlated equilibria. The new algorithm uses space and time linear in the number of corners n for each play, in contrast to $O(n^2)$ space and $O(n^3)$ time for the corners-as-experts algorithm, and unbounded space and time for the Stoltz and Lugosi algorithm. Since the number of corners of a polyhedron in d dimensions with $\mathrm{poly}(d)$ faces can be exponential in d, and since the number of corners of a general set can be even larger, the resulting savings can be substantial.

2 Online convex programming

When playing a repeated convex game, a single agent's learning problem is an **online convex program** (OCP): in each round t, the agent chooses an action $a_t \in A \subset \mathbb{R}^d$. At the same time, forces external to the agent choose a loss vector[3] $l_t \in L \subset \mathbb{R}^d$; the agent observes l_t and pays $l_t \cdot a_t$. The action space A is a convex compact subset of $\mathbb{R}^d$. The loss vector space L is a bounded subset of

[2] Low external regret does guarantee that the empirical distribution of play is near a *coarse correlated equilibrium* (defined in Sec. 5). But, to enforce a coarse correlated equilibrium, the agents must not be able to guess their recommended actions until *after* committing to play them. In our setting the agents compute their own actions (and so can't hide this information from themselves) and do not have any mechanism to commit to an action even if they wanted to.

[3] For consistency with standard notation in game theory and online optimization, we describe games in terms of payoffs and OCPs in terms of losses, with the understanding that a loss is just a negative payoff.

$\mathbb{R}^d$. An **experts** problem is a special case of an OCP in which the feasible region is the probability simplex in d dimensions.

A **learning algorithm** takes as input a sequence of loss vectors $\{l_t\}$ 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 elements of the loss vector sequence. The learner's objective is to minimize its cumulative loss

$$L_t = \sum_{\tau=1}^{t} l_\tau(a_\tau)$$

The minimum achievable loss depends of course on the sequence $\{l_t\}$. To measure how well our learning algorithm has done, we can calculate its **regret**: the simplest type of regret is called **external** regret, and is defined as

$$\rho_t^{\mathrm{EXT}} = \sup_{a \in A} \sum_{\tau=1}^{t} (l_\tau \cdot a_\tau - l_\tau \cdot a)$$

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 $\mathcal{A}$ is **no external regret** for feasible region A and loss vector set L if we can guarantee that its average external regret per trial goes to zero (or remains negative). In other words, $\mathcal{A}$ is no external regret if there is a function $f(t, A, L)$, sublinear in t, such that:

$$\sum_{\tau=1}^{t} l_\tau \cdot a_\tau \leq l_\tau \cdot a + f(t, A, L) \qquad \forall a \in A, \forall t \geq 1 \tag{1}$$

Sublinear in t means that, for any fixed A and L, $f(t, A, L) \in o(t)$. The function f may depend in a complicated way on A and L, but often depends on properties like the diameters of A and L under particular norms.

More generally, we could allow our agent to consider replacing its sequence $a_1 \dots a_t$ with $\phi(a_1) \dots \phi(a_t)$, where ϕ is some measurable transformation that maps the action space into itself. If A is any measurable action set (for this definition we drop the requirement of convexity and $\Phi \subset (A \mapsto A)$ is a set of allowed transformations, we define our algorithm's Φ**-regret** to be

$$\rho_t^{\Phi} = \sup_{\phi \in \Phi} \sum_{\tau=1}^{t} (l_\tau \cdot a_\tau - l_\tau \cdot \phi(a_t))$$

and say it is **no** Φ**-regret** if it satisfies the analogue of Eq. 1,

$$\sum_{\tau=1}^{t} l_\tau \cdot a_\tau \leq \sum_{\tau=1}^{t} l_\tau \cdot \phi(a_\tau) + g(t, A, L, \Phi) \qquad \forall \phi \in \Phi, \forall t \geq 1 \tag{2}$$

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

External regret is just Φ-regret with Φ equal to the set of constant transformations: $\Phi_{\mathrm{EXT}} = \{\phi_x \mid x \in A\}$, where $\phi_x(a) = x$. By setting Φ to larger, more flexible sets of transformations, we can define new, stronger varieties of regret.

For experts problems, one stronger type of regret is **internal** regret. Here, we define a slight generalization called σ-regret, which makes sense for general OCPs: the σ-transformations are defined as $\Phi_\sigma = \{\phi_{\sigma,x} \mid \sigma \subseteq A, x \in A\}$ where

$$\phi_{\sigma,x}(a) = \begin{cases} x & \text{if } a \in \sigma \\ a & \text{otherwise} \end{cases}$$

Each $\phi_{\sigma,x}$ acts like the identity for $a \notin \sigma$, and replaces any $a \in \sigma$ by x. If we restrict σ to be a singleton set $\{y\}$, we recover the ordinary internal transformation $\phi_{y,x}$.

An apparently even stronger type of regret corresponds to the **swap** transformations: Φ_{SWAP} is defined to be the set of all measurable mappings from A to itself. (Hart and Schmiedler [7] used swap transformations to define correlated equilibria for infinite games.) However, it turns out that

no σ-regret is equivalent to no swap regret in OCPs, just as no internal regret is equivalent to no swap regret in experts problems.

The above types of regret are straightforward generalizations of the corresponding regret types for experts problems. In this paper we study two additional types of regret: **linear regret** and **finite-element regret**. In experts problems, linear, finite-element, and σ-regret are equivalent to one another and to the standard notion of internal regret. In general OCPs, however, these three types of regret are distinct; see Sec. 4.2 for examples that demonstrate the difference.

Linear regret corresponds to the set of all linear transformations that map A into itself:

$$\Phi_{\mathrm{LIN}} = \{\phi \in \mathbb{R}^{d \times d} \mid (\forall a \in A)\, \phi a \in A\}$$

We assume that A is **homogeneous**, that is, satisfies $u \cdot a = 1$ for some $u \in \mathbb{R}^d$ and all $a \in A$. So, Φ_{LIN} effectively includes affine transformations as well. (For example, the probability simplex is homogeneous with $u = (1, 1, \ldots, 1)^{\mathrm{T}}$. We can make any action set homogeneous by adding an extra constant coordinate to each action; we sometimes leave this constant implicit if we can do so without confusion.) The set Φ_{LIN} is convex if A is, since it is the intersection of linear inequalities on ϕ: it is $\cap_{a,m,b}\{m \cdot \phi a \geq b\}$ where $a \in A$ and where $m \cdot a' \geq b$ for all $a' \in A$.

Finite-element regret corresponds to a set of transformations Φ_{FE} defined below in Section 4.1. These transformations make sense only for polyhedral feasible regions A; but, for polyhedral A, they include all linear transformations as special cases.

3 Algorithm

We will discuss the connection between our three new types of regret and various notions of equilibrium in more detail below, in Sec. 5. But first, we present and analyze a new algorithm which learns to minimize Φ-regret for a large class of transformation sets Φ. These sets Φ are flexible enough to include external, linear, and finite-element regret transformations as special cases.

Our algorithm works by representing Φ as the composition of a fixed nonlinear continuous "feature function" with an adjustable linear transformation. That is, we assume $\Phi = \{\phi_C \mid C \in \mathcal{C}\}$, where

$$\phi_C(a) = CB(a), \forall a \in A \tag{3}$$

Here B is our feature function, which maps the feasible region $A \subset \mathbb{R}^d$ to a p-dimensional feature space, while $\mathcal{C}$ is a set of $d \times p$ matrices which map the feature space back down to the d-dimensional feasible region. Often, $p \gg d$. (Sec. 4 gives examples of useful Bs.)

Write $\mathcal{B} = B(A)$ for the image of the feasible set A under the feature mapping B. We assume that both $\mathcal{B}$ and $\mathcal{C}$ are compact. We can assume without loss of generality that $\mathcal{C}$ is convex: if two transformations $C, C' \in \mathcal{C}$ each map $\mathcal{B}$ into A, then so does $\alpha C + (1-\alpha)C'$ for any $\alpha \in [0, 1]$.

When Φ is represented as in Eq. 3, there is an interesting connection between Φ-regret on the set A and external regret on the set $\mathcal{C}$. Our algorithm is based on this connection.

We want our algorithm $\mathcal{A}$ to achieve no Φ-regret for the feasible region A and loss vector set L. That is, we want $\mathcal{A}$ to take in a sequence of loss vectors l_t and produce a sequence of predictions $a_t \in A$ satisfying Eq. 2 above. To do so, we construct a new sequence of fictitious loss vectors m_t to pass to a copy of a no external regret algorithm $\mathcal{A}'$, which we run as a subroutine. We tell $\mathcal{A}'$ that its feasible region is $\mathcal{C}$, and so each m_t represents a linear function on $\mathcal{C}$. (Note that $\mathcal{C}$ is a set of $d \times n$ matrices, which we interpret as length-dn vectors by concatenating their columns.)

Given the sequence m_t, $\mathcal{A}'$ responds with a sequence of matrices $C_t \in \mathcal{C}$. We use each C_t to construct an action $a_t \in A$, and we play a_t. We will design these constructions of m_t and a_t so that, as long as $\mathcal{A}'$ achieves no external regret, the overall algorithm $\mathcal{A}$ will achieve no Φ-regret.

In more detail, since ϕ_{C_t} is a continuous operator on a compact set, the Brouwer fixed point theorem guarantees there exists at least one a such that $\phi_{C_t}(a) = a$. We set a_t to be any such a. After we play a_t, we observe l_t and incur loss $l_t \cdot a_t$. We then set $m_t = l_t B(a_t)^{\mathrm{T}}$. (Again, we are identifying elements of $\mathbb{R}^{dn}$ with $d \times n$ matrices for convenience.)

Given these definitions of a_t and m_t, the no-external-regret guarantee for $\mathcal{A}'$ (Eq. 2) tells us that

$$\sum_{t=1}^{T} m_t \cdot C_t \leq \sum_{t=1}^{T} m_t \cdot C + f(T, \mathcal{C}, L\mathcal{B}^{\mathrm{T}}) \qquad \forall C \in \mathcal{C}$$

where $L\mathcal{B}^{\mathrm{T}} = \{lb^{\mathrm{T}} \mid l \in L, b \in \mathcal{B}\}$, and f is sublinear in T.

Since the dot product of two $d \times n$ matrices can be written $X \cdot Y = \mathrm{tr}(X^{\mathrm{T}}Y)$, and since $\mathrm{tr}(XY) = \mathrm{tr}(YX)$, $\forall C$,

$$m_t \cdot C = \mathrm{tr}(B(a_t){l_t}^{\mathrm{T}}C) = \mathrm{tr}({l_t}^{\mathrm{T}}CB(a_t)) = {l_t}^{\mathrm{T}}CB(a_t)$$

and so

$$\sum_{t=1}^{T} {l_t}^{\mathrm{T}}C_tB(a_t) \leq \sum_{t=1}^{T} {l_t}^{\mathrm{T}}CB(a_t) + f(T, \mathcal{C}, L\mathcal{B}^{\mathrm{T}}) \qquad \forall C \in \mathcal{C}$$

But, since $C_tB(a_t) = \phi_{C_t}(a_t) = a_t$, and since each $\phi \in \Phi$ can be represented as $\phi(a) = CB(a)$ with $C \in \mathcal{C}$, this implies

$$\sum_{t=1}^{T} {l_t}^{\mathrm{T}}a_t \leq \sum_{t=1}^{T} {l_t}^{\mathrm{T}}\phi(a_t) + f(T, \mathcal{C}, L\mathcal{B}^{\mathrm{T}}) \qquad \forall \phi \in \Phi$$

which is exactly the required no-Φ-regret guarantee. So, we have proven

Theorem 1 *For any convex compact feasible region A and bounded loss vector space L, and for any set of transformations $\Phi : A \mapsto A$ represented as in Equation 3, $\mathcal{A}$ achieves no Φ-regret whenever its subroutine $\mathcal{A}'$ achieves no external regret.*

As described so far, algorithm $\mathcal{A}$ can play interior points of the feasible region A. To make a tighter connection to game-theoretic equilibria, we can also define a **randomized** variant of $\mathcal{A}$ which plays only boundary points of A. On a trial where the deterministic algorithm would have played $\bar{a}_t$, the randomized algorithm samples its play a_t from a distribution D such that

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

(NB: We still use $\bar{a}_t$, rather than a_t, in the construction of m_t.) If we pick D so that it puts weight only on the boundary of A, then the randomized algorithm will only play boundary points.[4] With this choice of action and loss vector, our Φ-regret on A and external regret on $\mathcal{C}$ differ by a zero-mean random variable; so, we can use standard stochastic convergence results to prove the following corollary. (See Appendix A for details.)

Corollary 2 *Under the conditions of Theorem 1 and the additional assumption (4), if $\mathcal{A}'$ achieves no external regret, then the randomized variant of $\mathcal{A}$ achieves no Φ-regret with probability 1.*

4 Transformations

We presented our algorithm in terms of a generic set of transformations in the form of Eq. 3. To implement the algorithm, we must choose a specific feature function $B(a)$ and a specific set $\mathcal{C}$ of linear transformations; and, we must pick (hopefully efficient) subroutines for achieving no external regret on $\mathcal{C}$ and for finding fixed points of transformations ϕ_C. (All other calculations in our algorithm are $O(dn)$, so these two subroutines will usually be the steps that limit our efficiency.)

To achieve no linear regret, we can take $B(a) = a$. $\mathcal{C}$ will then be a set of square ($d \times d$) matrices. To find a fixed point of ϕ_C, we can pick an appropriate element of the null space of $C - I$, which takes time d^3. The more expensive task is to achieve no external regret on $\mathcal{C}$: depending on the form of A, $\mathcal{C}$ may or may not have a description in terms of a small number of simple constraints.

If A is a probability simplex, then $\mathcal{C}$ is the set of stochastic matrices, which can be expressed with a small number of linear constraints on the entries of C (this setting yields an algorithm very similar to one of Blum and Mansour [8]). If A is a circle, $\mathcal{C}$ consists of the matrices whose largest singular value is ≤ 1; this set can be represented using 1 semidefinite constraint. For general (convex compact) A, the best choice may be to use either GIGA or lazy projection [2]: the difficult step in these algorithms is a Euclidean projection onto $\mathcal{C}$, which we can achieve via the ellipsoid algorithm.

[4] For general feature functions B, there may be no distribution which satisfies our requirements. But, a suitable distribution always exists for the feature functions we are interested in.

4.1 Finite-element transformations

To achieve no finite-element regret, we will define B as shown in Fig. 1. (This choice of B also defines Φ_{FE}.) Given a polyhedral feasible set A, $B(a)$ is a point in a high-dimensional space called the **barycentric coordinate space**. To construct B, we first associate each of the n corners of A with one dimension of $\mathbb{R}^n$. We then triangulate A by dividing it into mutually exclusive and exhaustive d-simplexes, so that each corner of A is a corner of one or more simplices.

Now, to calculate $B(a)$, 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 $j(1) \dots j(d+1)$ 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 $b_1 \dots b_{d+1}$ by solving $a = \sum_i b_i c_{j(i)}$, $\sum_i b_i = 1$. We then set entry $[B(a)]_{j(i)} = b_i$ for each corner $j(i)$, and set all other entries of $B(a)$ to 0.

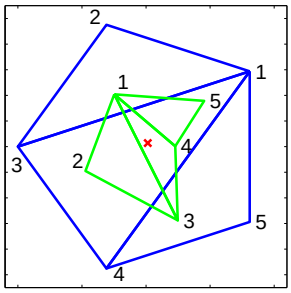

Figure 1: Illustration of barycentric coordinates and Φ_{FE}. A is the outer pentagon, triangulated into three mesh elements. $B(A)$ is a subset of the simplex in $\mathbb{R}^5$ (not shown). $\phi_C(A)$ is the distorted pentagon; the $\times$ shows a fixed point of ϕ_C.

To achieve no external regret in the $\mathcal{C}$ which corresponds to this B, we note that this $\mathcal{C}$ is factored: it is the Cartesian product of n copies of A, since we just need to map each corner of A to a point inside A. So, we can separately run n copies of any no-external-regret algorithm for A. A typical cost for doing so might be $O(nd^3)$.[5]

To find a fixed point of ϕ_C, we just need to check each of its linear pieces separately. Each individual check is the same as for linear regret, and there are n of them, so the total cost is $O(nd^3)$ for this step.

4.2 Comparisons between types of regret

We claimed above that the three new types of regret are equivalent to internal regret in experts problems. To see why, we can start from the standard definition of internal regret for experts problems: internal regret assumes that we play only corners of the probability simplex, and considers transformations which map a single corner to an arbitrary other corner. Since the probability simplex has only d corners in $\mathbb{R}^d$, a linear transformation can place one corner anywhere without affecting the others; this fact shows that linear regret is at least as strong as internal regret. But, since internal regret is the strongest type of Φ-regret for experts problems, linear regret is also at most as strong as internal regret; therefore, the two types of regret are equivalent.

The argument is essentially the same for finite-element and σ-regret. In fact, a stronger statement holds for these latter two regret types: if we happen to play only at corners in an OCP with a polyhedral feasible region A, then our finite-element or σ-regret in the OCP is at least as large as our internal regret in the corners-as-experts problem, since we can duplicate all of the internal transformations on the corners of the simplex using finite-element or σ-transformations on the original convex set A.

We also claimed above that no external regret is weaker than no linear regret, which is in turn weaker than no finite-element regret, which is in turn weaker than no σ-regret. The first statement follows

[5]The cost will depend heavily on the shape of A: if A is particularly simple the cost could be as low as $O(nd)$. For example, if A is a simplex we can use Hedge or external-regret matching—but on the simplex there is a reduced need for our no-Φ-regret algorithm, since Φ-regret on the simplex is already well studied. 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. Taking the LP as an example, each iteration of an interior point method is typically $O(d^3)$, with a constant that depends on the complexity (number of faces) of A. And, we need a number of iterations that depends on the complexity (condition number) of A. Running n copies of the algorithm will therefore take $O(nd^3)$ time, but the notation hides a factor that depends on the two measures of complexity of A.

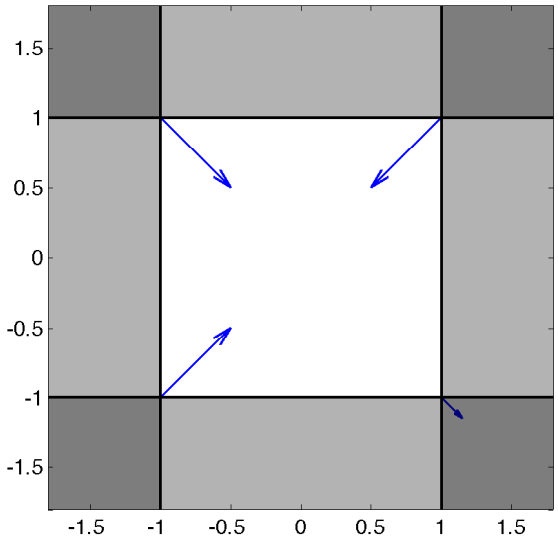

Figure 2: The feasible points and cost vectors for the counterexample. To improve readability, cost vector lengths are not to scale.

from the equivalence between linear and internal regret in experts problems, since external regret is weaker than internal regret for experts problems.

For the second statement, we exhibit an OCP and a sequence of play in which an agent does not experience linear regret but does experience finite-element regret. (It is obvious that finite-element transformations are at least as strong as linear ones, since all linear transformations of the feasible region are included in the set Φ_{FE}.) Suppose our feasible set A is the unit square in the $z = 1$ plane in R^3, with corners $a = (-1, +1, 1)^T$, $b = (-1, -1, 1)^T$, $c = (+1, +1, 1)^T$, and $d = (+1, -1, 1)^T$. (The third coordinate, which is constant for all feasible hypotheses, will be helpful in specifying transformations.) Also suppose we use the triangulation (abc, bcd).

Suppose that the agent plays the sequence a, b, c, d and observes the following sequence of loss vectors on time steps 1 through 4:

$$l_1 = (1, -1, 0)^T,\ l_2 = (1, 1, 0)^T,\ l_3 = (-1, -1, 0)^T,\ l_4 = (0.1, -0.1, 0)^T$$

The sequence of plays is illustrated in Figure 2: the cost vector for each time step is plotted with its base at the agent's play for that time step. On the first three rounds, the agent played the best possible corner, but on the last round, the agent played the worst possible corner. Because l_4 is a much smaller vector than $l_1 \ldots l_3$, the agent feels no linear regret; but, the agent feels regret with respect to the finite-element transformation which leaves a, b, and c in place but maps d to a. (For more detail, see Appendix D.)

For the last statement (finite-element vs. σ), consider the feasible set $A = [0, 1]$. On this set, the linear and finite-element transformations are the same. Suppose the agent plays $a_1 = 0.3$, $a_2 = 0.7$ and observes $l_1 = 1$, $l_2 = -1$. The agent experiences no linear regret, but has regret with respect to the σ-transformation that maps the singleton set $\{0.3\}$ to 0. In the other direction, the finite-element transformations are all measurable; so, assuming that our conjecture above holds (i.e., that σ-regret and swap regret are equivalent), σ-regret is strictly stronger than finite-element regret.

5 Equilibria and Regret

The algorithm of Sec. 3 guarantees no Φ-regret in an online convex program, for any Φ represented as in Eq. 3. In this section, we relate this guarantee back to equilibria in convex games.

Most generally, a game consists of a set of players N, a set of actions A_i for each player $i \in N$, and a reward function $r_i : \otimes_i A_i \to \mathbb{R}$ for each player. For technical reasons each action set will also need a σ-algebra. To simplify notation we denote the joint action set as $\mathcal{A} = \otimes_i A_i$. Given a joint action $a \in \mathcal{A}$, we indicate player i's action by a_i, and the actions of all players except for i by $a_{\neg i}$.

The two types of games we are interested in are **matrix games**, in which each action set is finite, and **convex games**, in which each action set is a convex subset of Euclidean space and each reward function is multi-linear (linear in each of its arguments). (In the former case the σ-algebras are the power sets; in the latter case we use the Borel σ-algebras.)

Following Greenwald and Jafari [9], equilibria in games can be defined in terms of transformations (as defined in Section 2) of the action sets. In this section, we will write the action set explicitly, as in $\Phi_{\mathrm{FE}}(A_i)$ or $\Phi_{\mathrm{SWAP}}(A_i)$.

Definition 3 *Given a game and a collection of sets of action transformations, $\{\Phi_i\}_{i \in N}$, $\Phi_i \subseteq \Phi_{\mathrm{SWAP}}(A_i)$, a probability distribution q over $\mathcal{A}$ is a $\{\Phi_i\}$-**equilibrium** if*

$$\mathbb{E}\left[r_i(\phi(a_i), a_{\neg i}) - r_i(a)\right] \leq 0 \quad \forall i \in N,\ \forall \phi \in \Phi_i. \tag{5}$$

Intuitively, an equilbrium is a distribution under which no player would prefer to deviate according to any transformation in its set. In the case of matrix games, taking each Φ_i to be the set of internal transformations defines **correlated equilibria** and taking each Φ_i to be the set of constant transformations defines **coarse correlated equilibria**.

The definition of coarse correlated equilibria applies also to convex games—coarse correlated equilibria in convex games are $\{\Phi_{\mathrm{EXT}}(A_i)\}$-equilibria. However, if we try to extend the above definition of correlated equilibria to convex games, the set of internal transformations will not give us a satisfactory result: if no single action has non-zero probability under q, then Equation 5 is trivially satisfied. Instead, we define a **correlated equilibrium for a convex game** using σ-transformations, i.e., it is a $\{\Phi_\sigma(A_i)\}$-equilibrium.[6]

Given a convex game with compact polyhedral action sets, we can generate a corresponding **corner game** by restricting each player's actions to the corners of its action sets. The corner game is relevant if we use an algorithm (such as the randomized algorithm of Sec. 3) that only plays corner points. The following Proposition is proved in Appendix B:

Proposition 4 *A correlated equilibrium of the corner game generated from a convex game is also a correlated equilibrium of the convex game.*

5.1 Repeated Games

As described above, our agents play a game repeatedly and try to learn by observing actions and payoffs. In the case of a convex game, the task facing each agent is equivalent to an online convex program: the joint action of the other agents, together with the reward function, determines the loss vector at each time step.

We define the **empirical distribution of play** at step t to be the following distribution over the joint action set $\mathcal{A}$, where $a^t \in \mathcal{A}$ is the joint action played at time step t:

$$z^t(\alpha) = \left|\{\tau \mid a_\tau = \alpha\}\right|/t$$

For matrix games, it is well known (e.g., [10]) that, if each agent plays according to a no-internal-regret algorithm, then the empirical distribution of play will converge to the set of correlated equilibria with probability 1. We have the following analogous result for $\{\Phi_i\}$-equilibrium in convex games (proven in Appendix C):

Theorem 5 *If each agent i plays a repeated convex game according to a no-Φ_i-regret algorithm, the empirical distribution of play converges to the set of $\{\Phi_i\}$-equilibria of the game with probability 1.*

[6] A reviewer pointed out that Hart and Schmiedler [7] defined correlated equilibria as $\{\Phi_{\mathrm{SWAP}}(A_i)\}$-equilibria. The two definitions are equivalent; a proof will be included in a future version of this paper.

A consequence of Theorem 5 is that, if each agent plays the corner game of the convex game according to an NIR algorithm, the joint empirical distribution of play will converge to the set of correlated equilibria of the convex game. In particular, if we use Φ_{FE} with the randomized algorithm of Sec. 3, the equivalence of internal regret on the corner game and finite-element regret on the OCP means that we achieve no internal regret; so, we have the following corollary:

Corollary 6 *If, in a repeated 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 algorithm of Sec. 3 with $\Phi = \Phi_{\mathrm{FE}}$), then the empirical distribution of 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.

Another consequence of Theorem 5 is that the deterministic version of the algorithm of Sec. 3 causes play to converge to a Φ_{FE}-equilibrium. While this type of equilibrium may not be interesting in its own right, it is worth noting that we can "flatten" such an equilibrium to find a correlated equilibrium: we can move all of the mass to corner points of $\mathcal{A}$ without changing any conditional expectations $\mathbb{E}\left[r_i(a) \mid a_i = \alpha\right]$ or introducing any Φ_{FE}-regret, and the resulting flattened distribution must therefore be near a correlated equilibrium of the convex game. So, if we just want to compute an approximate correlated equilibrium, running a number of copies of the deterministic algorithm against one another and then flattening may be a fast way to do so.

5.2 Convex games with concave payoffs

So far, we have assumed that our payoff function is multilinear. We could instead assume that it is concave in each argument separately, that is, that

$$r_i(pa_i + (1-p)a_i', a_{\neg i}) \geq p\, r_i(a_i, a_{\neg i}) + (1-p)\, r_i(a_i', a_{\neg i})$$

for all i, all $a_i, a_i' \in A_i$, all $p \in [0,1]$, and all $a_{\neg i} \in A_{\neg i}$. We will call such a payoff function **multi-concave**.

For multi-concave payoffs, the agents can still run no-Φ-regret algorithms using a simple modification based on Taylor expansions, and most of our guarantees will still hold. In more detail, suppose that we are given a no-Φ-regret algorithm for agent i. Write $\partial r_i(a)$ for a supergradient of r_i with respect to a_i, so that we have

$$r_i(a_i', a_{\neg i}) \leq r_i(a_i, a_{\neg i}) + (a_i' - a_i) \cdot \partial r_i(a_i, a_{\neg i})$$

for all i, all $a_i, a_i' \in A_i$, and all $a_{\neg i} \in A_{\neg i}$. And, suppose that on time step t, the observed joint play is a^t.

Then, we can tell agent i that its loss vector at step t is $l_t = -\partial r_i(a^t)$. In this case, the no-Φ-regret algorithm will guarantee us that, for any time T and any $\phi \in \Phi$,

$$\sum_{t=1}^{T} l_t \cdot a_i^t - \sum_{t=1}^{T} l_t \cdot \phi(a_i^t) \leq g(T, A_i, L_i, \Phi) \tag{6}$$

where $g \in o(T)$ and L_i is the set of possible values of l_i.

Now, for any $\phi \in \Phi$, we have

$$\begin{aligned} \sum_{t=1}^{T} r_i(\phi(a_i^t), a_{\neg i}^t) &\leq \sum_{t=1}^{T} r_i(a_i^t, a_{\neg i}^t) - \sum_{t=1}^{T} (\phi(a_i^t) - a_i^t) \cdot l_t & (7) \\ &\leq \sum_{t=1}^{T} r_i(a_i^t, a_{\neg i}^t) + g(T, A_i, L_i, \Phi) & (8) \end{aligned}$$

The first line uses the definition of a supergradient, and the second uses Eq. 6.

Eq. 8 is the analogue of Eq. 2 for multi-concave payoffs. In other words, by combining an ordinary no-Φ-regret algorithm with the supergradient of the reward function, we get a no-Φ-regret algorithm for a game with multi-concave payoffs.

However, in games with multi-concave payoffs, Corollaries 2 and 6 are no longer relevant: in a game with multi-linear payoffs,

$$\mathbb{E}\left[r_i(a_i, a_{\neg i})\right] = r_i(\mathbb{E}\left[a_i\right], a_{\neg i})$$

where the expectation is taken over player i's randomization. But, in a game with multi-concave payoffs, we can have

$$\mathbb{E}\left[r_i(a_i, a_{\neg i})\right] < r_i(\mathbb{E}\left[a_i\right], a_{\neg i})$$

So, in order to bound their Φ-regret, the agents may have to play interior points instead of corner points, and may not be able to use the randomized variant of algorithm $\mathcal{A}$. And, we will no longer be able to guarantee convergence to correlated equilibrium using our efficiently-representable finite-element transformations.

6 Discussion

We have presented several new forms of regret for online convex programs, and given the first algorithms which 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. 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.

Stoltz and Lugosi [6] proved the existence of an algorithm that minimizes swap regret and ensures convergence to correlated equilibria. However, they do not explicitly construct such an algorithm. Constructing an algorithm according to their proof of existence would require, at each time step t, computation of a convex combination of a set of transformations (departure functions) that grows unboundedly as t increases (due to their use of the doubling trick). Additionally, such a construction would need to find a fixed point of this convex combination, a non-trivial task given that the transformations may be arbitrary (continuous) functions.[7]

Still-stronger goals than correlated equilibrium of the stage game may be possible (and desirable). The most interesting of these goals may be convergence to some kind of equilibrium of the *repeated* game itself: according to the folk theorem, these equilibria include all stage-game equilibria, and may include many more. (For example, in an equilibrium of the repeated game, otherwise-unachievable payoffs may be sustained through the use of threats and promises.) While we do not yet know practical algorithms which effectively explore this larger set of equilibria, we point to the ideas of efficient learning equilibrium [11] and rational learning [12] as steps in the right direction.

References

[1] Geoffrey J. Gordon. Regret bounds for prediction problems. In *Proceedings of the ACM Conference on Computational Learning Theory*, 1999.

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

[3] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. In *Proceedings of the 16th Annual Conference on Learning Theory*, 2003.

[4] G. J. Gordon. No-regret algorithms for online convex programs. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, 2006.

[5] S. Shalev-Shwartz and Y. Singer. Convex repeated games and fenchel duality. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, 2006.

[6] G. Stoltz and G. Lugosi. Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, 59:187–208, 2007.

[7] Sergiu Hart and David Schmeidler. Existence of correlated equilibria. *Mathematics of Operations Research*, 14(1):18–25, 1989.

[8] Avrim Blum and Yishay Mansour. From external to internal regret. In *Proceedings of the Conference on Computational Learning Theory (COLT)*, 2005.

[7] In Footnote 4, Stoltz and Lugosi suggest that an approximation of their algorithm could in fact be constructed by means of partitioning, but no details (or complexity analyses) are provided.

[9] A. Greenwald and A. Jafari. A general class of no-regret algorithms and game-theoretic equilibria. In *Proceedings of the 16th Annual Conference on Learning Theory*, 2003.

[10] S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26–54, 2001.

[11] Ronen I. Brafman and Moshe Tennenholtz. Efficient learning equilibrium. *Artificial Intelligence*, 159(1–2), 2004.

[12] Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61(5):1019–1045, 1993.

A Proof of Corollary 2

From the proof of Theorem 1 and our new assumptions about the distribution of a_t, we have

$$\sum_{t=1}^{T} {l_t}^{\mathrm{T}} a_t \leq \sum_{t=1}^{T} {l_t}^{\mathrm{T}} \phi(a_t) + \sum_{t=1}^{T} {l_t}^{\mathrm{T}} R_t + f(T, C, L\mathcal{B}^{\mathrm{T}}) \qquad \forall \phi \in \Phi$$

where

$$R_t = \phi(\bar{a}_t) - \phi(a_t) + a_t - \bar{a}_t$$

Since R_t and l_t are both bounded, and since $E(R_t \mid l_{1:t-1}, a_{1:t-1}) = 0$, we can use standard martingale convergence results (encapsulated in Lemma 7 below) to show that $\sum_{t=1}^{T} l_t \cdot R_t < T^{0.5+\epsilon}$ with probability 1 as $T \to \infty$. □

Lemma 7 *Suppose the sequence X_t satisfies $E(X_t \mid \mathcal{F}_{t-1}) = 0$ and $E(X_t^2 \mid \mathcal{F}_{t-1}) \leq K$ for some K, where $\mathcal{F}_t$ is a sequence of σ-algebras such that $X_{1:t}$ are measurable with respect to $\mathcal{F}_t$. Then*

$$P\left(\sum_{t=1}^{T} X_t \geq T^{0.5+\epsilon}\right) \leq KT^{-2\epsilon}$$

for all $\epsilon > 0$ and all $T \geq 1$.

PROOF: Consider the sequences

$$Y_t = \sum_{i=1}^{t} X_i \qquad Z_t = Y_t^2$$

Clearly $Z_t \geq 0$. We also have

$$Z_t = (Y_{t-1} + X_t)^2 = Y_{t-1}^2 + 2X_t Y_{t-1} + X_t^2$$

So,

$$E(Z_t \mid \mathcal{F}_{t-1}) = Z_{t-1} + E(X_t^2 \mid \mathcal{F}_{t-1}) \leq Z_{t-1} + K \leq Kt$$

Now, by Doob's inequality (Lemma 8),

$$P(Z_t \geq t^{1+2\epsilon}) \leq Kt/t^{1+2\epsilon}$$

So,

$$P(Y_t \geq t^{0.5+\epsilon}) \leq Kt^{-2\epsilon}$$

which is the desired result. □

Lemma 8 (Doob) *Suppose X_t is a nonnegative submartingale, that is, $X_t \geq 0$ and*

$$E(X_t \mid \mathcal{F}_{t-1}) \geq X_{t-1}$$

where $\mathcal{F}_t$ is a sequence of σ-algebras such that $X_{1:t}$ are measurable with respect to $\mathcal{F}_t$. Then

$$P(\sup\{X_{1:T}\} \geq C) \leq E(X_T)/C$$

for all $C > 0$ and all $T \geq 1$.

B Proof of Proposition 4

Let $\kappa(S)$ denote the set of corners of a convex set S. Let q be a correlated equilibrium of the corner game. Choose a player i, a set $S \subseteq A_i$, and an action $y \in A_i$. We can rewrite left-hand side of Equation 5 as

$$\mathbb{E}\left[r_i\left(\phi^{(S,y)}(a_i), a_{-i}\right) - r_i(a)\right] = \mathbb{E}\left[r_i(y, a_{-i}) - r_i(a) \mid a_i \in S\right] \tag{9}$$

Because A_i is convex we can write y as the convex combination of a finite number of corners, $\{c_j\} \subseteq \kappa(A_i)$: $y = \sum_j \beta_j c_j$, and by multi-linearity of r_i we have $r_i(y, a_{-i}) = \sum_j \beta_j r_i(c_j, a_{-i})$. By linearity of expectation:

$$\mathbb{E}\left[r_i(y, a_{-i}) - r_i(a) \mid a_i \in S\right] = \sum_j \beta_j \mathbb{E}\left[r_i(c_j, a_{-i}) - r_i(a) \mid a_i \in S\right] \tag{10}$$

Because q only puts weight on corners, we can take S to be a set of corners and rewrite the expectations:

$$\mathbb{E}\left[r_i(c_j, a_{-i}) - r_i(a) \mid a_i \in S\right] = \mathbb{E}\left[r_i\left(\phi^{(S,c_j)}(a_i), a_{-i}\right) - r_i(a)\right]. \tag{11}$$

Because q is a correlated equilibrium of the corner game, all such expectations are non-positive, so their convex combination must be as well.

C Proof of Theorem 5

First we present a technical lemma necessary for the proof:

Lemma 9 *Let (X, d_X) be a compact metric space and let (Y, d_Y) be a metric space. Let $\{x_t\}$ be an X-valued sequence, and let S be a nonempty, closed subset of Y. If $f : X \to Y$ is continuous and if $f^{-1}(S)$ is nonempty, then $d_X(x_t, f^{-1}(S)) \to 0$ as $t \to \infty$ if and only if $d_Y(f(x_t), S) \to 0$ as $t \to \infty$.*

PROOF: We write $d = d_X$ and $d = d_Y$, since the appropriate choice of distance metric is always clear from the context. To prove the forward implication, assume $d(x_t, f^{-1}(S))) \to 0$ as $t \to \infty$. Choose t_0 *s.t.* for all $t \geq t_0$, $d(x_t, f^{-1}(S)) < \frac{\delta}{2}$. Observe that for all x_t and for all $\gamma > 0$, there exists $q_t^{(\gamma)} \in f^{-1}(S)$ *s.t.* $d(x_t, q_t^{(\gamma)}) < d(x_t, f^{-1}(S)) + \gamma$. Now, since $d(x_t, q_t^{(\frac{\delta}{2})}) < \frac{\delta}{2} + \frac{\delta}{2} = \delta$, by the continuity of f, $d(f(x_t), f(q_t^{(\frac{\delta}{2})})) < \epsilon$, for all $\epsilon > 0$. Therefore, $d(f(x_t), S) < \epsilon$, since $f(q_t^{(\frac{\delta}{2})}) \in S$.

To prove the reverse implication, assume $d(f(x_t), S) \to 0$ as $t \to \infty$. We must show that for all $\epsilon > 0$, there exists a t_0 *s.t.* for all $t \geq t_0$, $d(x_t, f^{-1}(S)) < \epsilon$. Define $T = \{x \in X \mid d(x, f^{-1}(S)) \geq \epsilon\}$. If $T = \emptyset$, the claim holds. Otherwise, observe that T can be expressed as the complement of the union of open balls, so that T is closed and thus compact. Define $g : X \to \mathbb{R}$ as $g(x) = d(f(x), S)$. By assumption S is closed; hence, $g(x) > 0$, for all x. Because T is compact, g achieves some minimum value, say $L > 0$, on T. Choose t_0 *s.t.* $d(f(x_t), S) < L$ for all $t \geq t_0$. Thus, for all $t \geq t_0$, $g(x_t) < L \Rightarrow x_t \notin T \Rightarrow d(x_t, f^{-1}(S)) < \epsilon$. □

We now show that each agent's Φ_i regret converges to 0 if and only if the empirical distribution of play converges to the set of $\{\Phi_i\}$-equilibria.

For each player i and transformation ϕ, let $f_{i,\phi}$ be the function that maps a distribution over $\mathcal{A}$ to the expected ϕ-regret for player i under that distribution. Let Z be set set of all distributions such that all players have non-positive regret for all of their transformations, i.e.,

$$Z = \cap_i \cap_{\phi \in \Phi_i} f_{i,\phi}^{-1}(\mathbb{R}_-)$$

Applying Lemma 9 for each $f_{i,\phi}$ implies that given a sequence of distributions, each player's Φ_i regret with respect to those distributions converges to 0 as $t \to \infty$ if and only if that sequence of distributions converges to the set of $\{\Phi_i\}$-equilibria.

Taking the sequence of distributions to be the sequence of empirical distributions of play and noting that an agent's time averaged ϕ regret at time t is equal to the ϕ regret under the empirical distribution of play at time t, we have the desired result.

D Difference between linear and finite-element regret

To calculate linear regret we consider all linear transformations from A to itself.

A general linear transformation mapping the $z = 1$ plane into itself has the form

$$\phi = \begin{pmatrix} x & y & z \\ u & v & w \\ 0 & 0 & 1 \end{pmatrix}$$

If we in addition wish to have ϕ map A into itself, the entries x, y, z, u, v, w must satisfy the additional constraints

$$\phi(a) \in A \quad \phi(b) \in A \quad \phi(c) \in A \quad \phi(d) \in A$$

By calculation, $\phi(a) = (-x + y + z, -u + v + w)^T$. To check that $\phi(a)$ is in A, we just need to check that each of its coordinates is in $[-1, 1]$:

$$-1 \leq -x + y + z \leq 1$$
$$-1 \leq -u + v + w \leq 1$$

Similarly, for $\phi(b)$:

$$-1 \leq -x - y + z \leq 1$$
$$-1 \leq -u - v + w \leq 1$$

And, for $\phi(c)$:

$$-1 \leq x + y + z \leq 1$$
$$-1 \leq u + v + w \leq 1$$

And, for $\phi(d)$:

$$-1 \leq x - y + z \leq 1$$
$$-1 \leq u - v + w \leq 1$$

The above constraints describe the set Φ of linear transformations from H into itself. We now have to find the transformation $\phi \in \Phi$ against which we have the most regret. This is the transformation which maximizes

$$\sum_{t=1}^{4} (l_t \cdot a_t - l_t \cdot \phi(a_t))$$

First, $l_1 \cdot a_1 = -2$; $l_2 \cdot a_2 = -2$; $l_3 \cdot a_3 = -2$; and $l_4 \cdot a_4 = 0.2$. Hence,

$$\sum_{t=1}^{4} l_t \cdot a_t = -5.8$$

Second, we have

$$l_t^T \phi(a_t) = \text{TR}(l_t a_t^T \phi)$$

So, we need to find $\sum_{t=1}^{4} l_t a_t^T$. The contribution at step 1 is:

$$a_1 l_1^T = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \end{pmatrix}$$

And at step 2,

$$a_2 l_2^T = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & -1 & 0 \\ -1 & -1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

For step 3,

$$a_3 l_3^T = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & -1 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & 0 \end{pmatrix}$$

Finally, for step 4,

$$a_4 l_4^T = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} \frac{1}{10} & -\frac{1}{10} & 0 \end{pmatrix} = \frac{1}{10} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix}$$

So, the total is:

$$\sum_{t=1}^{4} a_t l_t^T = \begin{pmatrix} -2.9 & -1.1 & 0 \\ -1.1 & -2.9 & 0 \\ 1.1 & -1.1 & 0 \end{pmatrix}$$

The regret versus ϕ is therefore

$$\text{TR}\left(-\left(\sum_{t=1}^{4} l_t a_t^T \right) \phi \right) - 5.8 = 2.9x + 1.1y + 1.1u + 2.9v - 5.8$$

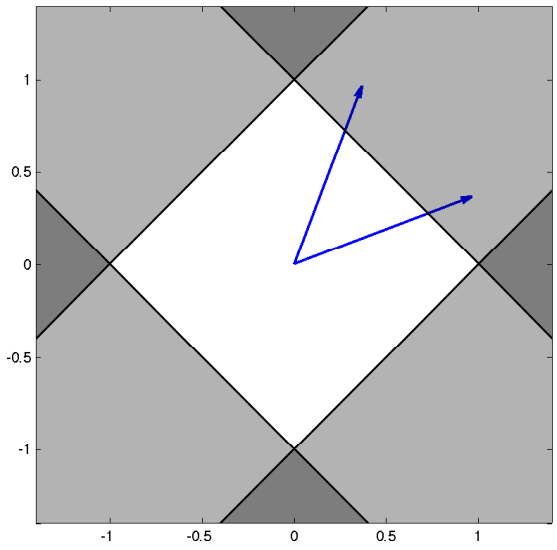

Figure 3: Constraints and objective for x and y in the problem of finding the maximum-regret transformation ϕ.

And, the ϕ which maximizes regret can be found by maximizing $2.9x + 1.1y + 1.1u + 2.9v - 5.8$ subject to the eight constraints described above.

It turns out that these constraints factor into separate constraints on x, y and u, v. (The feasible values of z and w depend on x, y, u, and v, but z and w do not affect the value of the objective.) The constraints for x and y are shown in Figure 3; the lower arrow points in the direction of increasing objective $2.9x + 1.1y$. The constraints for u and v are similar, but the objective points instead in the direction $1.1u + 2.9v$ (upper arrow). We can see from the figure that the maximum value occurs at $x = 1$, $y = 0$, $u = 0$, and $v = 1$; by plugging these values into the regret formula, we find that this ϕ achieves a regret of $2.9 + 2.9 - 5.8 = 0$. So, as claimed, we have no linear regret.