CS364A: Algorithmic Game Theory
Lecture #18: From External Regret to Swap Regret and the Minimax Theorem*

Tim Roughgarden†

November 20, 2013

1 Swap Regret and Correlated Equilibria

Last lecture we proved that coarse correlated equilibria (CCE) are tractable, in a satisfying sense: there are simple and computationally efficient learning procedures that converge quickly to the set of CCE. Of course, if anything in our equilibrium hierarchy (Figure 1) was going to be tractable, it was going to be CCE, the biggest set.

The good researcher is never satisfied and always seeks stronger results. What can we say if we zoom in to the next-biggest set, the correlated equilibria? The first part of this lecture shows that correlated equilibria are also tractable. We'll give computationally efficient — if not quite as simple — learning procedures that converge fairly quickly to this set.

Remark 1.1 (Learning vs. Linear Programming) The computational tractability of correlated and coarse correlated equilibria — and mixed Nash equilibria of two-player zero-sum games, see Section 3 — can also be demonstrated by formulating linear programs for them. A bonus of the linear programming approach is that an exact, rather than an approximate, equilibrium can be computed in polynomial time. Another advantage is that linear optimization over the set of equilibria remains computationally tractable, while learning procedures merely guide behavior to somewhere in the set. On the other hand, exact linear programming algorithms seem wholly unrelated to any reasonable model of how agents learn in games.

Recall from Lecture 13 and Exercise 59 that a *correlated equilibrium* of a cost-minimization game is a distribution σ over outcomes such that, for every player i with strategy set S_i and every switching function $\delta : S_i \to S_i$,

$$\mathbf{E}_{\mathbf{s}\sim\sigma}[C_i(\mathbf{s})] \leq \mathbf{E}_{\mathbf{s}\sim\sigma}[C_i(\delta(s_i), \mathbf{s}_{-i})]\,.$$

†Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.

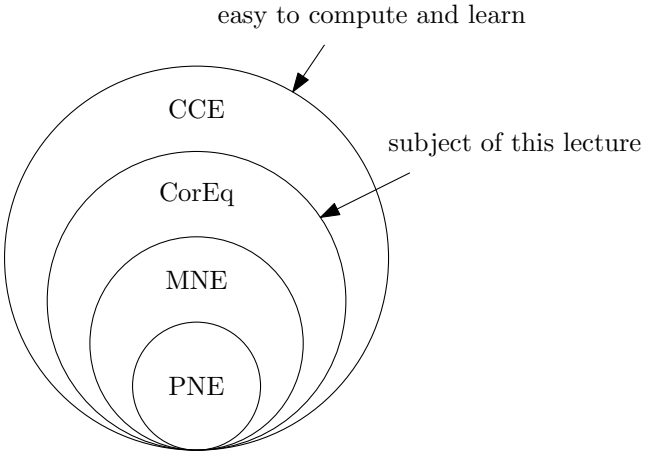

Figure 1: The hierarchy of equilibria from Lecture 13.

For example, in the "traffic intersection game" of Lecture 13, mixing 50/50 between the two pure Nash equilibria gives a (non-Nash) correlated equilibria.

Recall the online decision-making setting from last time: every day $t = 1, 2, \ldots, T$, a decision-maker commits to a distribution p^t over its n actions A, then an adversary chooses a cost function $c^t : A \to [0, 1]$, and finally an action a^t is chosen according to p^t, resulting in cost $c^t(a^t)$ to the decision-maker. Last lecture described an algorithm with time-averaged expected cost as small as that of every fixed action, up to an error term that goes to 0 as the time horizon T grows. When every player of a game uses such a no-external-regret algorithm to choose a strategy at each time step, the time-averaged history of joint play is an approximate CCE. Is there a more stringent regret notion that enjoys an analogous correspondence with correlated equilibria?

Definition 1.2 An online decision-making algorithm has *no swap regret* if for every adversary for it, the expected swap regret

$$\frac{1}{T}\left[\sum_{t=1}^{T} c^t(a^t) - \sum_{i=1}^{T} c^t(\delta(a^t))\right] \tag{1}$$

with respect to every switching function $\delta : A \to A$ is $o(1)$ as $T \to \infty$.

Because fixed actions are the special case of constant switching functions, an algorithm with no swap regret also has no external regret.

In each time step t of *no-swap-regret dynamics*, every player i independently chooses a mixed strategy p_i^t according to a no-swap-regret algorithm. Cost vectors are defined as in no-regret dynamics: $c_i^t(s_i)$ is the expected cost of strategy $s_i \in S_i$, given that every other player j plays its chosen mixed strategy p_j^t. The connection between correlated equilibria and no-swap-regret dynamics is the same as that between CCE and no-(external-)regret dynamics.

Proposition 1.3 *Suppose after T iterations of no-swap-regret dynamics, every player of a cost-minimization game has swap regret at most ϵ for each of its switching functions. Let $\sigma^t = \prod_{i=1}^k p_i^t$ denote the outcome distribution at time t and $\sigma = \frac{1}{T}\sum_{t=1}^T \sigma^t$ the time-averaged history of these distributions. Then σ is an ϵ-approximate correlated equilibrium, in the sense that*

$$\mathbf{E}_{\mathbf{s}\sim\sigma}[C_i(\mathbf{s})] \leq \mathbf{E}_{\mathbf{s}\sim\sigma}[C_i(\delta(s_i), \mathbf{s}_{-i})] + \epsilon$$

for every player i and switching function $\delta : S_i \to S_i$.

2 A Black-Box Reduction From Swap Regret to External Regret

This section gives a "black-box reduction" from the problem of designing a no-swap-regret algorithm to that of designing a no-external-regret algorithm — a problem that we already solved in the previous lecture.

Theorem 2.1 ([1]) *If there is a no-external-regret algorithm, then there is a no-swap-regret algorithm.*

As we'll see, the reduction in Theorem 2.1 also preserves computational efficiency. For example, plugging the multiplicative weights algorithm into this reduction yields a polynomial-time no-swap-regret algorithm. We conclude that correlated equilibria are tractable in the same strong sense as coarse correlated equilibria.

Proof of Theorem 2.1: The reduction is very natural, one that you'd hope would work. It requires one clever trick, as we'll see at the end of the proof.

Let n denote the number of actions. Let $M_1, \ldots, M_n$ denote n different no-(external-)regret algorithms, for example n instantiations of the multiplicative weights algorithm. Each of these algorithms is poised to produce probability distributions over actions and receive cost vectors as feedback. Very roughly, we can think of algorithm M_j as responsible for protecting against profitable deviations from action j to other actions.

The "master algorithm" M is as follows; see also Figure 2.

1. At time $t = 1, 2, \ldots, T$:
   (a) Receive distributions $q_1^t, \ldots, q_n^t$ over actions from the algorithms $M_1, \ldots, M_n$.
   (b) Compute and output a consensus distribution p^t.
   (c) Receive a cost vector c^t from the adversary.
   (d) Give algorithm M_j the cost vector $p^t(j) \cdot c^t$.

We discuss how to compute the consensus distribution p^t from the distributions $q_1^t, \ldots, q_n^t$ at the end of the proof; this is the clever trick in the reduction. The fourth step parcels out

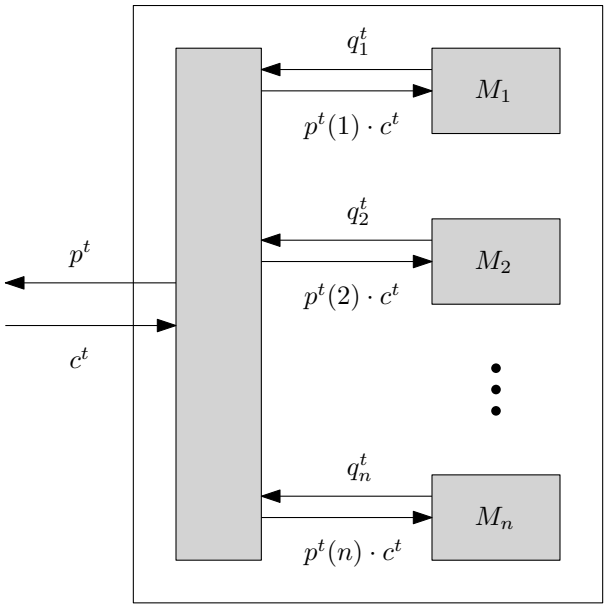

Figure 2: Blackbox reduction from swap regret to external regret.

the true cost vector c^t to the no-external-regret algorithms, scaled according to the current relevance (i.e., $p^t(j)$) of the algorithm.

Our hope is that we can piggyback on the no-external-regret guarantee provided by each algorithm M_j and conclude a no-swap-regret guarantee for the master algorithm M. Let's take stock of what we've got and what we want, parameterized by our computed consensus distributions $p^1, \ldots, p^T$.

The time-averaged expected cost of the master algorithm is

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{n} p^t(i) \cdot c^t(i). \tag{2}$$

The time-averaged expected cost under a switching function $\delta : A \to A$ is

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{n} p^t(i) \cdot c^t(\delta(i)). \tag{3}$$

Remember that our goal is to prove that (2) is at most (3), plus a term that goes to 0 as $T \to \infty$, for every switching function δ.

Now adopt the perspective of an algorithm M_j. This algorithm believes that actions are being chosen according to its recommended distributions $q_j^1, \ldots, q_j^T$ and that the true cost vectors are $p^1(j) \cdot c^1, \ldots, p^T(j) \cdot c^T$. Thus, the algorithm perceives its time-averaged expected cost as

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{n} q_j^t(i) \left(p^t(j) c^t(i) \right). \tag{4}$$

Since M_j is a no-regret algorithm, its perceived cost (4) is, up to the regret term, at most that of every fixed action $k \in A$:

$$\frac{1}{T}\sum_{t=1}^{T} p^t(j)c^t(k) + R_j, \tag{5}$$

where $R_j \to 0$ as $T \to \infty$.

Now fix a switching function δ. Summing the inequality between (4) and (5) over all $j = 1, 2, \ldots, n$, with k instantiated as $\delta(j)$ in (5), yields

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{n} q_j^t(i)p^t(j)c^t(i) \leq \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{n} p^t(j)c^t(\delta(j)) + \sum_{j=1}^{n} R_j. \tag{6}$$

Observe that the right-hand side of (6) is exactly (3), up to a term $\sum_{j=1}^{n} R_j$ that goes to 0 as $T \to \infty$. (Recall that we think of n as fixed as $T \to \infty$.) Indeed, we chose the splitting of the cost vector c^t amongst the no-external-regret algorithms $M_1, \ldots, M_n$ to guarantee this property.

If we can choose the consensus distributions $p^1, \ldots, p^T$ so that (2) and the left-hand side of (6) coincide, then the reduction will be complete. We show how to choose each p^t so that, for each $i \in A$ and $t = 1, 2, \ldots, T$,

$$p^t(i) = \sum_{j=1}^{n} q_j^t(i)p^t(j). \tag{7}$$

The left- and right-hand sides of (7) are the coefficients of $c^t(i)$ in (2) and in the left-hand side of (6), respectively.

The equations (7) might be familiar as those defining the stationary distribution of a Markov chain. This is the key trick in the reduction: given distributions $q_1^t, \ldots, q_n^t$ from algorithms $M_1, \ldots, M_n$ at time t, form the following Markov chain (Figure 3): the set of states is $A = \{1, 2, \ldots, n\}$, and for every $i, j \in A$, the transition probability from j to i is $q_j^t(i)$. That is, the distribution q_j^t specifies the transition probabilities out of state j. A probability distribution p^t satisfies (7) if and only if it is the stationary distribution of this Markov chain. At least one such distribution exists, and one can be computed in polynomial time via an eigenvector computation (see e.g. [3]). This completes the reduction. ■

Our choice of the consensus distribution p^t from the no-external-regret algorithms' suggestions $q_1^t, \ldots, q_n^t$ is uniquely defined by the proof approach, but it also has a natural interpretation as a limit of the following decision-making process. Suppose you first ask an arbitrary algorithm M_{j_1} for a recommended strategy. It gives you a recommendation j_2 drawn from its distribution $q_{j_1}^t$. You then ask algorithm M_{j_2} for a recommendation, which it draws from its distribution $q_{j_2}^t$, and so on. This random process is effectively trying to converge to a stationary distribution p^t of the Markov chain defined above, and will successfully do so when the chain is ergodic.

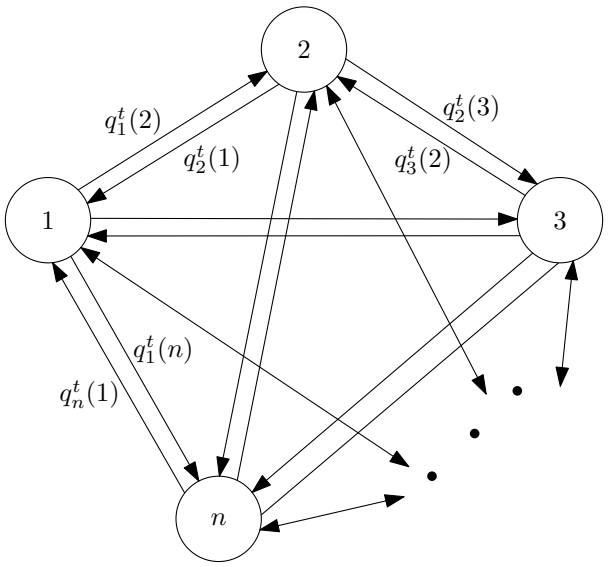

Figure 3: Markov chain.

3 The Minimax Theorem for Two-Player, Zero-Sum Games

Having resolved the complexity of correlated equilibria in satisfactory fashion, we now zoom in further to the set of mixed Nash equilibria (Figure 1). We'll see next week that, while the set of mixed Nash equilibria is guaranteed to be non-empty, computing one is a computationally intractable problem. Today we'll focus on a special case with a happier answer: two-player zero-sum games.

In a two-player zero-sum game, the payoff of each player is the negative of the other — one player can win only at the other's expense. Such a game can be specified by a single matrix A, with the two strategy sets corresponding to the rows and columns. The entry a_{ij} specifies the payoff of the row player in the outcome (i, j) and the negative payoff of the column player in this outcome. Thus, the row and column players prefer bigger and smaller numbers, respectively. The matrix below describes the payoffs in the Rock-Paper-Scissors game (Lecture 1) in our current language.

	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Pure Nash equilibria generally don't exist in two-player zero-sum games, so the focus is squarely on mixed Nash equilibria. We use $\mathbf{x}$ and $\mathbf{y}$ to denote mixed strategies (probability distributions) over the rows and columns, respectively.

With mixed strategies, we think of each player as randomizing independently. Thus, the expected payoff of the row player when payoffs are given by A, the row strategy is $\mathbf{x}$, and the column strategy is $\mathbf{y}$, is

$$\sum_{i,j} \mathbf{Pr}_{\mathbf{x}}[i] \cdot \mathbf{Pr}_{\mathbf{y}}[j] \cdot a_{ij} = \mathbf{x}^T A \mathbf{y};$$

the column player's expected payoff is the negative of this. Thus, a *mixed Nash equilibrium* is a pair $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ such that

$$\hat{\mathbf{x}}^T A \hat{\mathbf{y}} \geq \mathbf{x}^T A \hat{\mathbf{y}} \qquad \text{for all distributions } \mathbf{x} \text{ over rows}$$

and

$$\hat{\mathbf{x}}^T A \hat{\mathbf{y}} \leq \hat{\mathbf{x}}^T A \mathbf{y} \qquad \text{for all distributions } \mathbf{y} \text{ over columns.}$$

Suppose you're due to play a zero-sum game with someone else. Would you rather move — meaning commit to a mixed strategy — first or second? Intuitively, there is only a first-mover disadvantage, since the second player can adapt to the first player's strategy. The Minimax Theorem is the amazing statement that *it doesn't matter*.

Theorem 3.1 (Minimax Theorem) *For every two-player zero-sum game A,*

$$\max_{\mathbf{x}} \left(\min_{\mathbf{y}} \mathbf{x}^T A \mathbf{y} \right) = \min_{\mathbf{y}} \left(\max_{\mathbf{x}} \mathbf{x}^T A \mathbf{y} \right). \tag{8}$$

On the left-hand side of (8), the row player moves first and the column player second. The column plays optimally given the strategy chosen by the row player, and the row player plays optimally in light of the column player's behavior. On the right-hand side of (8), the roles of the two players are reversed.

The Minimax Theorem is equivalent to the statement that every two-player zero-sum game has at least one mixed Nash equilibrium (see the Exercises). Borel, who you might know from his work developing measure-theoretic probability, was interested in the latter problem. He was discouraged after he noticed the equivalence with the Minimax Theorem, which seemed intuitively false [2, Chapter 15]. In the 1920's, von Neumann proved the Minimax Theorem using Brouwer's fixed-point theorem. Many equilibrium existence results require fixed-point theorems — more on this soon — but the Minimax Theorem can also be proved with less heavy machinery. In the 1940's, von Neumann proved the Minimax Theorem again, using arguments equivalent to strong linear programming duality.[1] This is why, when a very nervous George Dantzig first explained his new simplex algorithm to von Neumann, the latter was able to respond with an impromptu lecture outlining the corresponding duality theory [4]. These days, we don't even need linear programming per se to prove the Minimax

[1] This implies that minimax pairs and, equivalently, Nash equilibria, can be computed in polynomial time in two-player zero-sum games. See the Problems for details.

Theorem — all we need is the existence of a no-(external-)regret algorithm, such as the multiplicative weights algorithm![2]

Proof of Theorem 3.1: Since it's only worse to go first, the left-hand side of (8) is at most the right-hand side: if $\hat{\mathbf{x}}$ is optimal for the row player when it plays first, it always has the option of playing $\hat{\mathbf{x}}$ when it plays second. We turn our attention to the reverse inequality.

Given a two-player zero-sum game A, suppose both players play the game using their favorite no regret algorithms, for a long enough time T so that both have expected regret at most ϵ with respect to every fixed strategy. For example, if both players use the MW algorithm from last lecture, then $T = \Theta((\ln n)/\epsilon^2)$ is long enough.[3]

Formally, let $\mathbf{p}^1, \ldots, \mathbf{p}^T$ and $\mathbf{q}^1, \ldots, \mathbf{q}^T$ be the mixed strategies played by the row and column players, respectively, as advised by their no-regret algorithms. The inputs to the no-regret algorithms at time t are $A\mathbf{q}^t$ for the row player and $(\mathbf{p}^t)^T A$ for the column player — the expected payoff of each strategy on day t, given the mixed strategy played by the other player on day t. Set

$$\hat{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{p}^t$$

to be the time-averaged mixed-strategy of the row player,

$$\hat{\mathbf{y}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{q}^t$$

to be the time-averaged mixed-strategy of the column player, and

$$v = \frac{1}{T}\sum_{t=1}^{T} (\mathbf{p}^t)^T A \mathbf{q}^t$$

the time-averaged expected payoff of the row player.

Adopt the row player's perspective. Since its expected regret is at most ϵ with respect to every row i and corresponding pure strategy e_i, we have

$$(e_i)^T A\hat{\mathbf{y}} = \frac{1}{T}\sum_{t=1}^{T} (e_i)^T A\mathbf{q}^t \leq \frac{1}{T}\sum_{t=1}^{T} (\mathbf{p}^t)^T A\mathbf{q}^t + \epsilon = v + \epsilon. \tag{9}$$

Since an arbitrary row mixed strategy $\mathbf{x}$ is just a distribution over the e_i's, by linearity (9) implies that

$$\mathbf{x}^T A\hat{\mathbf{y}} \leq v + \epsilon \tag{10}$$

[2]It is not hard to prove that the Minimax Theorem and strong linear programming duality are equivalent, so this argument establishes the latter as well!

[3]Last lecture we defined online decision-making problems and regret in terms of cost vectors. It is straightforward to adjust the definitions for payoff vectors. It is also straightforward to adapt the MW algorithm to payoff-maximization while preserving its optimal regret bound of $O(\sqrt{(\ln n)/T})$; see the Exercises for details.

for every mixed row strategy $\mathbf{x}$.

A symmetric argument from the column player's perspective, using that its expected regret is also at most ϵ for every fixed strategy, shows that

$$\hat{\mathbf{x}}^T A\mathbf{y} \geq v - \epsilon \tag{11}$$

for every mixed column strategy $\mathbf{y}$. Thus

$$\begin{aligned}
\max_{\mathbf{x}} \left(\min_{\mathbf{y}} \mathbf{x}^T A\mathbf{y} \right) &\geq \min_{\mathbf{y}} \hat{\mathbf{x}}^T A\mathbf{y} \\
&\geq v - \epsilon && (12) \\
&\geq \max_{\mathbf{x}} \mathbf{x}^T A\hat{\mathbf{y}} - 2\epsilon && (13) \\
&\geq \min_{\mathbf{y}} \left(\max_{\mathbf{x}} \mathbf{x}^T A\mathbf{y} \right) - 2\epsilon,
\end{aligned}$$

where (12) and (13) follow from (11) and (10), respectively. Taking the limit as $\epsilon \downarrow 0$ (and $T \to \infty$) completes the proof. ■

There are a number of easy but useful corollaries of the Minimax Theorem and its proof. First, in the limit, the mixed strategies $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are a Nash equilibrium of the game A. This establishes the existence of Nash equilibria in all two-player zero-sum games. This is remarkable because most equilibrium existence results require the use of a fixed-point theorem. Second, the equivalence between Nash equilibria and minimax pairs — row and column mixed strategies $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ that optimize the left- and right-hand sides of (8), respectively — implies a "mix and match" property: if $(\mathbf{x}^1, \mathbf{y}^1)$ and $(\mathbf{x}^2, \mathbf{y}^2)$ are Nash equilibria of the same two-player zero-sum game, then so are $(\mathbf{x}^1, \mathbf{y}^2)$ and $(\mathbf{x}^2, \mathbf{y}^1)$.

References

[1] A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007.

[2] V. Chvátal. *Linear Programming*. Freeman, 1983.

[3] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 2012. Fourth edition.

[4] J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver, editors. *History of Mathematical Programming: A Collection of Personal Reminiscences*. CWI, 1991.