Bounds for Regret-Matching Algorithms

Amy Greenwald AMY@BROWN.EDU
Zheng Li ZHENG@DAM.BROWN.EDU
Casey Marks CASEY@CS.BROWN.EDU
Brown University, Providence, RI 02912

Abstract

We introduce a general class of learning algorithms, regret-matching algorithms, and a regret-based framework for analyzing their performance in online decision problems. Our analytic framework is based on a set Φ of transformations over the set of actions. Specifically, we calculate a Φ-regret vector by comparing the average reward obtained by an agent over some finite sequence of rounds to the average reward that could have been obtained had the agent instead played each transformation $\phi \in \Phi$ of its sequence of actions. The regret matching algorithms analyzed here select the agent's next action based on the vector of Φ-regrets, along with a link function f. Many well-studied learning algorithms are seen to be instances of regret matching. We derive bounds on the regret experienced by (f, Φ)-regret matching algorithms for polynomial and exponential link functions (though we consider polynomial link functions for $p > 1$ rather than $p \geq 2$). Although we do not improve upon the bounds reported in past work (except in special cases), our means of analysis is more general, in part because we do not rely directly on Taylor's theorem. Hence, we can analyze algorithms based on a larger class of link functions, particularly non-differentiable link functions. In ongoing work, we are indeed studying regret matching with alternative link functions, other than polynomial and exponential.

1 Introduction

In this paper, we introduce a general class of learning algorithms, regret-matching algorithms, and a regret-based framework for analyzing their performance in online decision problems (ODPs). In an ODP, an agent repeatedly faces some decision. During each round, the agent plays an action and obtains a reward that depends on its choice of action. The reward function, which governs the relationship between rewards and actions, may change over the course of the ODP, and the particular reward function that applies at any given round is not revealed until after the agent has chosen its action for that round.

Online decision problems encompass a wide variety of machine learning settings. Consider an agent learning in an infinitely repeated one-shot game, for example. If we view this setup as an ODP, the reward dynamics are jointly determined by the payoff matrix and the behavior of the other agents. If all the agents are learning simultaneously, these dynamics need not be stationary. The power of regret-matching algorithms is that bounds on the regret they experience apply to any ODP.

Following Greenwald and Jafari [2003], our analytic framework is based on a set Φ of transformations over the set of actions. Specifically, we calculate a Φ-regret vector by comparing the average reward obtained by an agent over some finite sequence of rounds to the average reward that could have been obtained had the agent instead played each transformation $\phi \in \Phi$ of its sequence of actions. The regret-matching algorithms analyzed here select the agent's next action based on the vector of Φ-regrets, along with a link function f. Many well-studied learning algorithms are seen to be instances of regret matching (e.g., Freund and Schapire [1996], Foster and Vohra [1999]).

Our work is closely related to that of Cesa-Bianchi and Lugosi [2003]. However, our framework is based on sets of transformations, rather than pools of experts. The experts framework applies to some settings, such as "shifting experts" [Freund et al., 1997], where the transformation framework does not, and the transformation framework applies to other settings, such as swap regret [Blum and Mansour, 2005], where the experts framework does not. [1] The two most common forms of regret, internal (or conditional) regret [Foster and Vohra, 1995] and external regret [Hannan, 1957], fit into both frameworks.

Like Cesa-Bianchi and Lugosi [2003], we derive bounds on the regret experienced by (f, Φ)-regret-matching algorithms for polynomial and exponential link functions (though we consider polynomial link functions for $p > 1$ rather than $p \geq 2$). Although we do not improve upon the bounds reported in past work (except in special cases), our means of analysis is more general, in part because we do not rely directly on Taylor's theorem. Hence, we can analyze algorithms based on a larger class of link functions, particularly non-differentiable link functions.[2] In ongoing work, we are studying regret matching with non-standard link functions.

2 Regret Analysis

2.1 Online Decision Problem

Formally, an online decision problem is parameterized by a *reward system* – a pair $(A, \mathcal{R})$, where A is a set of actions and $\mathcal{R}$ is a set of rewards. In this work we consider only ODPs with finite action sets and real-valued rewards (i.e., $|A| \in \mathbb{N}$ and $\mathcal{R} \subset \mathbb{R}$). Further, we restrict our attention to bounded rewards. WLOG, we let $\mathcal{R} = [0, 1]$.

A particular instance of an ODP is described by a *reward schedule* – a sequence of functions $\{r_t\}_{t=1}^{\infty}$, where each $r_t : A \to \mathcal{R}$. For $a \in A$, $r_t(a)$ corresponds to the reward the agent receives for playing action a in round t. For the remainder of this paper, an ODP will be assumed to be defined with respect to the reward system $(A,\ [0, 1])$ for some finite set A.

We denote by $\Delta(A)$ the set of probability distributions over the set A, and we allow the agent to play *mixed strategies*, which means that rather than selecting an action $a \in A$ to play at each round, the agent chooses a mixed strategy $q \in \Delta(A)$. Hence, round t proceeds like so:

1. the agent selects a mixed strategy $q_t \in \Delta(A)$,
2. an action $a_t \in A$ is sampled from q_t,
3. the agent receives reward $r_t(a_t)$,
4. the agent is informed of r_t.

The last step, in which the agent learns what rewards it would have obtained for actions that were not played, characterizes an *informed* ODP, which is the subject of the work. Omitting this step, that is, informing the agent only of $r_t(a_t)$, yields a *naïve* ODP. See Auer et al. [1995], for example, for consideration of the naïve setting.

The *set of histories of length* t is denoted by H^t and is given by $A^t \times \{r : A \to \mathcal{R}\}^t$. An *online learning algorithm* is a sequence of functions $\mathcal{L} = \{\mathcal{L}_t\}_{t=1}^{\infty}$, where $\mathcal{L}_t : H^{t-1} \to \Delta(A)$ so that $\mathcal{L}_t(h) \in \Delta(A)$ corresponds to the mixed strategy that is played at time t, for $h \in H^{t-1}$. We define H^0 to be a singleton.

2.2 Transformations

Given an action set A, an *action transformation* is a function $\phi : A \to \Delta(A)$. We let Φ_{ALL} denote the set of all action transformations over the set A. Following Blum and Mansour [2005], we let $\Phi_{\text{SWAP}}(A)$ denote the set of action transformations that map actions to pure strategies (i.e., distributions with all their weight on a single action). Let $\delta_a \in \Delta(A)$ denote the distribution with all its weight on a.

[1] Lehrer's (2003) setup, which combines history-dependent "replacing schemes" with activation functions, is the most general of the three, but rather than derive bounds on the regret that is accrued by regret-matching algorithms after finite time t, he shows that certain such algorithms exhibit no-regret as $t \to \infty$.

[2] There appears to be a problem with Cesa-Bianchi and Lugosi [2003]'s analysis in that they apply Taylor's theorem to the polynomial link function even though it is not differentiable at the origin.

There are two well-studied subsets of Φ_{SWAP}: external and internal action transformations. An external transformation is simply a constant transformation, so for $a \in A$,

$$\phi^{(a)}_{\text{EXT}} : x \mapsto \delta_a, \quad \text{for all } x \in A \tag{1}$$

Internal transformations behave like the identity, except on one particular input, so for $a, b \in A$

$$\phi^{(a,b)}_{\text{INT}} : x \mapsto \begin{cases} \delta_b & \text{if } x = a \\ \delta_x & \text{otherwise} \end{cases} \tag{2}$$

Let $\Phi_{\text{EXT}}(A)$ denote the set of external transformations and let $\Phi_{\text{INT}}(A)$ denote the set of internal transformations. Observe that $|\Phi_{\text{SWAP}}(A)| = |A|^{|A|}$, $|\Phi_{\text{INT}}(A)| = |A|^2$, and $|\Phi_{\text{EXT}}(A)| = |A|$.

We can extend an action transformation to a strategy transformation. Given an action transformation $\phi : A \to \Delta(A)$, let $[\phi] : \Delta(A) \to \Delta(A)$ be the linear transformation defined by

$$[\phi](q) = \sum_a q \cdot \phi(a) \tag{3}$$

2.3 Regret

Given a reward function $r : A \to \mathcal{R}$, an action $a \in A$, and an action transformation $\phi \in \Phi_{\text{ALL}}$, the *$\phi$-regret* is given by $\rho^\phi(r, a) = \mathbb{E}_{a' \sim \phi(a)} r(a') - r(a)$. This quantity is the difference between the rewards that the agent obtains by playing action a and the rewards that the agent would have expected to obtain by playing the transformed strategy $\phi(a)$. Given a set of action transformations $\Phi \subseteq \Phi_{\text{ALL}}(A)$, the *$\Phi$-regret* vector is given by $\rho^\Phi(r, a) = (\rho^\phi(r, a))_{\phi \in \Phi}$.

Given an ODP over A with reward schedule $\{r_t\}$, a sequence of actions $\{a_t\}$, and a set $\Phi \subset \Phi_{\text{ALL}}(A)$, *cumulative* Φ-regret at time T is

$$R^\Phi_T(\{r_t\}, \{a_t\}) = \sum_{t=1}^{T} \rho^\Phi(r_t, a_t). \tag{4}$$

The ϕ entry in the cumulative regret vector compares the cumulative rewards obtained by an agent at time T and the rewards that it would have obtained by consistently transforming each of its actions by ϕ. In general we are interested in minimizing the maximal element of finite cumulative regret vectors.

We sometimes treat cumulative regret as a function from histories to regret vectors and write $R^\Phi_t(h)$, where $h \in H^T$ for $t \leq T$. When considering a particular ODP and learning algorithm we treat cumulative regret as a random vector over the probability space defined by the reward schedule and the behavior of the algorithm. In this case we write R^Φ_t.

As shorthand we write R^{ALL}_T to denote $R^{\Phi_{\text{ALL}}}_T$, and similarly R^{SWAP}_T, R^{EXT}_T, and R^{INT}_T.

Since $\Phi_{\text{SWAP}}(A) \supseteq \Phi_{\text{EXT}}(A), \Phi_{\text{INT}}(A)$, it follows that $\max_\Phi R^{\text{EXT}}_t \leq \max_\Phi R^{\text{SWAP}}_t$ and $\max_\Phi R^{\text{INT}}_t \leq \max_\Phi R^{\text{SWAP}}_t$. Additionally, from Marks et al. [2004], we recall that $\max_\phi R^{\text{EXT}}_t \leq (|A| - 1) \max_\phi R^{\text{INT}}_t$ and $\max_\phi R^{\text{SWAP}}_t \leq |A| \max_\phi R^{\text{INT}}_t$.

A commonly studied property is "no-regret:"

Definition 1 *Given an action set A and a set of action transformations $\Phi \subseteq \Phi_{ALL}(A)$, an online learning algorithm is said to exhibit* no-Φ-regret *if there exists $\epsilon > 0$ such that for any ODP (reward schedule $\{r_t\}_{t=1}^\infty$), $P(\frac{1}{t} \sup_{\phi \in \Phi} R^\phi_t > \epsilon) < \epsilon$.*

3 Regret Matching

In this section, we define a general class of online learning algorithms, which we call regret-matching algorithms,[3] that are parameterized by a set of action transformations Φ and a link function[4] $f : \mathbb{R}^\Phi \to \mathbb{R}^\Phi_+$.

[3] We co-opt this terminology from Hart and Mas-Colell [2001], whose regret matching algorithm based on Φ_{EXT} and the $p = 2$ polynomial link function is an instance of this class.

[4] Some authors, such as Cesa-Bianchi and Lugosi [2003], base their analyses on potential functions, while others, such as Gordon [1999] use link functions. These approaches are equivalent if a link function is considered to be a (sub-)gradient of a

Regret matching algorithms satisfy a property closely related to Blackwell's condition for approachability. This property is parameterized by a set of transformations Φ and a link function f.

In this paper we consider two well-known classes of link functions: the polynomial link function, $f_i(x) = (x_i^+)^{p-1}$ for some $p > 1$, and the exponential link function, $f_i(x) = e^{\eta x_i}$ for some $\eta > 0$. (Here $x_i^+ = \max\{x_i, 0\}$.)

3.1 Preliminaries

Given two sets X and Y, we denote by X^Y the set of functions $\{f : X \to Y\}$. In particular, if Y is a finite set then $\mathbb{R}^Y$ is isomorphic to $\mathbb{R}^{|Y|}$. We denote the positive orthant of R^n by $R^n_+ \equiv \{\vec{x} \in R^n : x_i \geq 0 \ \forall i\}$.

3.2 Regret Matching Property

We define the regret matching property, which is inspired by Blackwell's approachability condition (1956) and related to Hart and Mas-Colell's Λ-strategy property (2001).

Definition 2 (Regret Matching) *Given a finite set of action transformations $\Phi \subset \Phi_{ALL}(A)$, and a function $f : \mathbb{R}^\Phi \to \mathbb{R}^\Phi_+$, a learning algorithm $\mathcal{L}$ is called an* (f, Φ)-regret-matching algorithm *if for all reward functions r, for all times T, for all histories $h \in H^{T-1}$,*

$$f(R^\Phi_{t-1}(h)) \cdot \mathbb{E}_{a \sim \mathcal{L}_t(h)}[\rho^\Phi_t(a, r)] \leq 0. \tag{5}$$

Given a finite set $\Phi \subseteq \Phi_{\text{ALL}}(A)$ of action transformations, Blackwell's condition (1956) for a regret vector to approach the negative orthant is equivalent to the existence of an (f, Φ)-regret-matching algorithm with the $p = 2$ polynomial link function. An immediate consequence of this observation is that such an algorithm exhibits no-Φ-regret [Greenwald and Jafari, 2003].

The bounds derived in Section 5 suggest that the polynomial Φ-regret-matching algorithms exhibit no-Φ-regret for all $p > 1$ and for any set of action transformations Φ. Such an analysis is the subject of future work.

We rely on the following observation in Section 5.

Observation 3 *Let f, f' be link functions, mapping $\mathbb{R}^\Phi$ to $\mathbb{R}^\Phi_+$. If there exists a function $\psi : \mathbb{R}^\Phi \to \mathbb{R}_+$ such that $\psi(x)f(x) = f'(x)$ and $\|f(x)\| > 0 \Rightarrow \psi(x) > 0$ for all $x \in \mathbb{R}^\Phi$, then a learning algorithm is an (f, Φ)-regret-matching algorithm if and only if it is an (f', Φ)-regret matching algorithm.*

3.3 Regret Matching Algorithms

Given a set of action transformations Φ, a link function $f : \mathbb{R}^\Phi \to \mathbb{R}^\Phi_+$, and a history h of length $t - 1$, apply the link function to the cumulative regret vector $R^\Phi_{t-1}(h)$, yielding a vector $Y_t \equiv f(R^\Phi_{t-1}(h)) \in \mathbb{R}^\Phi_+$. Then define a strategy transformation M_t by taking a convex combination of strategy transformations weighted by Y_t as follows:

$$M_t(h) = \frac{\sum_{\phi \in \Phi}(Y_t)_\phi[\phi]}{\sum_{\phi \in \Phi}(Y_t)_\phi}. \tag{6}$$

We prove that a learning algorithm that plays a fixed point of M_t at every round t such that M_t is well-defined (i.e., whenever Y_t is not the zero vector) is a regret-matching algorithm. Formally,

Theorem 4 (Regret Matching Theorem) *Given a finite set of action transformations $\Phi \subset \Phi_{ALL}(A)$, and a function $f : \mathbb{R}^\Phi \to \mathbb{R}^\Phi_+$, a learning algorithm $\mathcal{L}$ that for all times t, for all histories $h \in H^{t-1}$ such that M_t is well-defined, satisfies $\mathcal{L}_t(h) = M_t(\mathcal{L}_t(h))$ (where $M_t : \Delta(A) \to \Delta(A)$ is defined in Equation 6) is an (f, Φ)-regret-matching algorithm.*

potential function.

Proof Define $q \equiv \mathcal{L}_t(h)$, $Y_t \equiv f\left(R^{\Phi}_{t-1}\right)$, and M_t as in Equation 6. We need to show that

$$Y_t \cdot \mathbb{E}_{a\sim q}[\rho^{\Phi}_t(a,r)] \leq 0. \tag{7}$$

If Y_t is the zero vector, the conclusion follows immediately. Otherwise,

$$\begin{aligned}
Y_t \cdot \mathbb{E}_{a\sim q}[\rho^{\Phi}_t(a,r)] &= \sum_{\phi\in\Phi}(Y_t)_\phi \; \mathbb{E}_{a\sim q}[\rho^{\phi}_t(a,r)] && (8)\\
&= \sum_{\phi\in\Phi}(Y_t)_\phi \; r\cdot([\phi](q)-q) && (9)\\
&= r\cdot\sum_{\phi\in\Phi}(Y_t)_\phi\,([\phi](q)-q) && (10)\\
&= r\cdot\left(\sum_{\phi\in\Phi}(Y_t)_\phi\;[\phi](q) - \sum_{\phi\in\Phi}(Y_t)_\phi\; q\right) && (11)\\
&= \left(\sum_{\phi\in\Phi}(Y_t)_\phi\right) r\cdot(M_t(q)-q) && (12)\\
&= \left(\sum_{\phi\in\Phi}(Y_t)_\phi\right) r\cdot(q-q) && (13)\\
&= 0 && (14)
\end{aligned}$$

Line (12) follows because $\mathcal{L}_t(h)$ is a fixed point of $M_t(h)$. ■

Because M_t is a linear transformation on a finite-dimensional simplex, by Brouwer's fixed point theorem it has a fixed point. In an informed setting, the agent is able to compute R^{Φ}_{t-1} and therefore construct M_t before choosing its mixed strategy q_t. Thus a learning algorithm which computes and plays the fixed point of M_t at every iteration (such that M_t exists) is a regret-matching algorithm. The pseudocode for this algorithm is shown in Algorithm 1.

3.4 Complexity

Maintaining a cumulative regret vector and computing Y_t has time cost $O(|\Phi|)$ at each iteration (assuming the complexity of f is linear in $|\Phi|$, as it is for the link functions we consider here). If we represent the strategy transformations $[\phi]$ as $|A|\times|A|$ matrices, then the matrix for M_t can computed one entry at a time from Y_t in $O(|A|^2)$ time. Finding the fixed point of that matrix can be accomplished via Gaussian elimination in $O(|A|^3)$ time. Thus we have an (f,Φ)-regret-matching algorithm which has complexity $O(\max\{|A|^3,|\Phi|\})$. For both internal and external regret matching this simplifies to $O(|A|^3)$ complexity.

However, for external regret matching we present an $O(|A|)$ regret-matching algorithm which does not require matrix manipulation. Observe that for $\Phi = \Phi^{\text{EXT}}(A)$, for any $q \in \Delta(A)$,

$$M_t(h)(q) : a \mapsto \frac{f_a(R^{\text{EXT}}_t(h))}{\sum_{a'\in A} f_{a'}(R^{\text{EXT}}_t(h))}. \tag{15}$$

Thus the (unique) fixed-point of the linear transformation can be computed as an $O(|A|)$ operation.

4 General Bounding Theorem

The theorem presented in this section is inspired by unpublished work due to Geoffrey Gordon.

Algorithm 1 (f, Φ)-RegretMatchingAlgorithm

```
initialize $R_0 = 0$
initialize $q_1 \in \Delta(A)$ arbitrarily
for $t = 1, \ldots, \infty$ do
    sample pure action $a_t$ from $q_t$
    observe reward function $r_t$
    for all $\phi \in \Phi$ do
        compute instantaneous regret $\rho^\Phi(a_t, r_t)$
        update cumulative regret vector $R_T^\Phi = R_{T-1}^\Phi + \rho^\Phi(a_t, r_t)$
    end for
    let $Y = f(R_T^\Phi)$
    if $Y = 0$ then
        choose $q_{t+1} \in \Delta(A)$ arbitrarily
    else
        let $M = \sum_{\phi \in \Phi} Y_\phi[\phi] / \sum_{\phi \in \Phi} Y_\phi$
        let $q_{t+1}$ be a fixed point of $M$
    end if
end for
```

Definition 5 (Gordon Triple) *For a positive integer d, Let $G : \mathbb{R}^d \to \mathbb{R}$, $g : \mathbb{R}^d \to \mathbb{R}_+^d$, $\gamma : \mathbb{R}^d \to \mathbb{R}$ satisfy*

$$G(x + y) \leq G(x) + g(x) \cdot y + \gamma(y) \tag{16}$$

for all $x, y \in R^d$. We call the triple $\langle G, g, \gamma \rangle$ a Gordon triple.

The function G is a potential function and the function g will be the gradient or sub-gradient of G. When applied to regret-matching algorithms, g will also be a link function. The following theorem bounds the growth of the potential function:

Theorem 6 *For a positive integer d, let $x_1, x_2, \ldots$ be a sequence of random vectors taking values in $\mathbb{R}^d$. Let $X_t = \sum_{\tau=1}^t x_\tau$. Let $\langle G, g, \gamma \rangle$ be a Gordon triple. Additionally let $C : \mathbb{N} \to \mathbb{R}$ satisfy*

$$\mathbb{E}_{t-1}\left[g(X_{t-1}) \cdot x_t\right] + \mathbb{E}_{t-1}\left[\gamma(x_t)\right] \leq C(t) \ \textit{a.s.} \tag{17}$$

Then

$$\mathbb{E}\left[G(X_t)\right] \leq G(0) + \sum_{\tau=1}^t C(\tau) \tag{18}$$

Proof Proof by induction. For $t = 0$,

$$\mathbb{E}\left[G(0)\right] = G(0) \tag{19}$$

Assume (18) holds for a particular $t \geq 0$.

$$\begin{aligned} G(X_{t+1}) &= G(X_t + x_{t+1}) & (20)\\ &\leq G(X_t) + g(X_t) \cdot x_{t+1} + \gamma(x_{t+1}) & (21) \end{aligned}$$

Take conditional expectations on both sides:

$$\mathbb{E}_t\left[G(X_{t+1})\right] \leq G(X_t) + C(t+1) \text{ a.s.} \tag{22}$$

Now take expectations on both sides, and apply the law of iterated expectations on the left-hand side:

$$\begin{aligned} \mathbb{E}\left[G(X_{t+1})\right] &\leq \mathbb{E}\left[G(X_t)\right] + C(t+1) && (23) \\ &\leq G(0) + \sum_{\tau=1}^{t} C(\tau) + C(t+1) && (24) \\ &= G(0) + \sum_{\tau=1}^{t+1} C(\tau) && (25) \end{aligned}$$

thus completing the induction. ■

Corollary 7 *Let $\langle G, g, \gamma \rangle$ be a Gordon triple. Given a reward system and a g, Φ-regret-matching algorithm $\mathcal{L}$, the cumulative Φ regret experienced by playing according to $\mathcal{L}$ will be bounded by*

$$\mathbb{E}\left[G(R_t^\Phi)\right] \leq G(0) + t \max_{r,a} \gamma(\rho^\Phi(r,a)) \tag{26}$$

for any reward schedule.

Proof Apply Theorem 6 with $d = |\Phi|$, $x_t = \rho^\Phi(r_t, a_t)$, so $X_t = R_t^\Phi$. Playing according to $\mathcal{L}$ means that $P_t(a_t = a') = \mathcal{L}_t(h)(a')$, so from the regret matching property we get

$$\mathbb{E}_{t-1}\left[\rho^\Phi(r_t, a_t) \cdot g(R_{t-1}^\Phi)\right] = 0 \tag{27}$$

for all t and any r_t. Thus we choose $C(t) = \max_{r,a} \gamma(\rho^\Phi(r,a))$. ■

5 Bounds for Specific Link Functions

We now derive regret bounds for polynomial and exponential regret-matching algorithms by applying Corollary 7 with particular Gordon triples.

5.1 Polynomial Link Functions

We divide our analysis of the polynomial link function, $f_i(x) = (x_i^+)^{p-1}$, into two cases: $p \geq 2$ and $1 < p \leq 2$. In both cases we rely on the following lemmas:

Lemma 8 *If x is a random vector taking values in $\mathbb{R}^n$, then $(\mathbb{E}[\max_i x_i])^q \leq \mathbb{E}\left[\|x^+\|_p^q\right]$ for all $p > 0$ and $q \geq 1$.*

Proof Apply Jensen's inequality and the fact that $\|x\|_\infty \leq \|x^+\|_p$. ■

Given a set of actions A and a set of action transformations $\Phi \subseteq \Phi_{\text{ALL}}(A)$, the *maximal activation*, denoted $\mu(\Phi)$, is computed by maximizing, over all actions $a \in A$, the number of transformations ϕ that alter action a: i.e.,

$$\mu(\Phi) = \max_{a \in A} |\{\phi \in \Phi : \phi(a) \neq a\}| \tag{28}$$

Clearly, $\mu(\Phi) \leq |\Phi|$. In addition, observe that $\mu(\Phi_{\text{EXT}}(A)) = \mu(\Phi_{\text{INT}}(A)) = |A| - 1$.

Lemma 9 *Given an ODP over action set A and a set of action transformations $\Phi \subseteq \Phi_{\text{ALL}}(A)$, $\|\rho^\Phi(r,a)\|_p \leq \sqrt[p]{\mu(\Phi)}$ for any action a and reward function r.*

Proof Rewards are bounded in $[0, 1]$, so regrets are bounded in $[-1, 1]$.

$$\begin{aligned} \|\rho^{\Phi}(r, a)\|_p &= \sqrt[p]{\sum_{\phi \in \Phi} (\rho^{\phi}(r, a))^p} && (29) \\ &\leq \sqrt[p]{\sum_{\phi \in \Phi} \mathbb{1}_{\phi(a) \neq a}} && (30) \\ &\leq \sqrt[p]{\mu(\Phi)} && (31) \end{aligned}$$

■

Lemma 10 *For $p \geq 2$, define $G(x) = \|x^+\|_p^2$,*

$$g_i(x) = \begin{cases} 0 & \textit{if } x = 0, \\ \frac{2(x_i^+)^{p-1}}{\|x\|_p^{p-2}} & \textit{otherwise,} \end{cases}$$

and $\gamma(x) = (p-1)\|x\|_p^2$. Then $\langle G, g, \gamma \rangle$ is a Gordon triple.

Proof See appendix.

Theorem 11 *Given an action set A and a finite set of action transformations $\Phi \subseteq \Phi_{ALL}(A)$, define $f : \mathbb{R}^{\Phi} \to \mathbb{R}^{\Phi}$ by $f_i(x) = (x_i^+)^{p-1}$, for $2 \leq p < \infty$. At all times t, an (f, Φ)-regret-matching algorithm guarantees*

$$\mathbb{E}\left[\max_{\phi \in \Phi} \frac{1}{t} R_t^{\phi}\right] \leq \sqrt{\frac{p-1}{t}} \sqrt[p]{\mu(\Phi)} \tag{32}$$

for any reward schedule $\{r_t\}_{t=1}^{\infty}$.

Proof Let $\langle G, g, \gamma \rangle$ be the Gordon triple defined in Lemma 10. By Lemma 3, an (f, Φ)-regret-matching algorithm is also a (g, Φ)-regret-matching algorithm. Now,

$$\begin{aligned} \left(\mathbb{E}\left[\max_{\phi \in \Phi} R_t^{\phi}\right]\right)^2 &\leq \mathbb{E}\left[\|(R_t^{\Phi})^+\|_p^2\right] && (33) \\ &= \mathbb{E}\left[G(R_t^{\Phi})\right] && (34) \\ &\leq G(0) + t \max_{r,a} \gamma(\rho^{\Phi}(r, q)) && (35) \\ &\leq t(p-1)\left(\sqrt[p]{\mu(\Phi)}\right)^2 && (36) \end{aligned}$$

Line (33) follows by Lemma 8. Line (35) follows by Corollary 7. Line (36) follows by Lemma 9. Finally, the conclusion follows by taking square roots and dividing by t on both sides. ■

Lemma 12 *For $p \leq 2$, define $G(x) = \|x^+\|_p^p$, $g_i(x) = p(x_i^+)^{p-1}$, and $\gamma(x) = \|x\|_p^p$. Then $\langle G, g, \gamma \rangle$ is a Gordon triple.*

Proof See appendix.

Theorem 13 *Given an action set A and a finite set of action transformations $\Phi \subseteq \Phi_{ALL}(A)$, define $f : \mathbb{R}^{\Phi} \to \mathbb{R}^{\Phi}$ by $f_i(x) = (x_i^+)^{p-1}$, for $1 < p \leq 2$. At all times t, an (f, Φ)-regret-matching algorithm guarantees*

$$\mathbb{E}\left[\max_{\phi \in \Phi} \frac{1}{t} R_t^{\phi}\right] \leq t^{\left(\frac{1}{p}-1\right)} \sqrt[p]{\mu(\Phi)} \tag{37}$$

for any reward schedule $\{r_t\}_{t=1}^{\infty}$.

Proof Let $\langle G, g, \gamma \rangle$ be the Gordon triple defined in Lemma 12. By Lemma 3, an (f, Φ)-regret-matching algorithm is also a (g, Φ)-regret-matching algorithm. Now,

$$\begin{aligned}
\left(\mathbb{E}\left[\max_{\phi \in \Phi} R_t^{\phi} \right] \right)^p &\leq \mathbb{E}\left[\left\| \left(R_t^{\Phi} \right)^+ \right\|_p^p \right] && (38) \\
&= \mathbb{E}\left[G\left(R_t^{\Phi} \right) \right] && (39) \\
&\leq G(0) + t \max_{r,a} \gamma(\rho^{\Phi}(r,q)) && (40) \\
&\leq t\ \mu(\Phi) && (41)
\end{aligned}$$

Line (38) follows by Lemma 8. Line (40) follows by Corollary 7. Line (41) follows by Lemma 9. Finally, the conclusion follows by taking p-roots and dividing by t on both sides. ■

5.2 Exponential Link Functions

Lemma 14 *Define* $G(x) = \frac{1}{\eta} \ln \left(\sum_i e^{\eta x_i} \right)$, $g_i(x) = \frac{e^{\eta x_i}}{\sum_j e^{\eta x_j}}$, *and* $\gamma(x) = \frac{\eta}{2} \|x\|_\infty^2$. *Then* $\langle G, g, \gamma \rangle$ *is a Gordon triple.*

Proof See appendix.

Theorem 15 *Given an action set* A *and a finite set of action transformations* $\Phi \subseteq \Phi_{\scriptscriptstyle ALL}(A)$, *define* $f : \mathbb{R}^{\Phi} \to \mathbb{R}^{\Phi}$ *by* $f_i(x) = e^{\eta x_i}$, *for* $\eta > 0$. *At all times* t, *an* (f, Φ)*-regret-matching algorithm guarantees*

$$\mathbb{E}\left[\max_{\phi \in \Phi} \frac{1}{t} R_t^{\phi} \right] \leq \frac{\ln m}{\eta t} + \frac{\eta}{2} \tag{42}$$

for any reward schedule $\{r_t\}_{t=1}^{\infty}$.

Proof Let $\langle G, g, \gamma \rangle$ be the Gordon triple defined in Lemma 12. By Lemma 3, an (f, Φ)-regret-matching algorithm is also a (g, Φ)-regret-matching algorithm. Note that $G(0) = \frac{1}{\eta} \ln m$ and $\gamma(\rho^{\Phi}(r,q)) = \frac{\eta}{2} \|\rho^{\Phi}(r,q)\|_\infty^2 \leq \frac{\eta}{2}$. Also observe that

$$\begin{aligned}
\max_i x_i &= \max_i \ln e^{x_i} && (43) \\
&= \ln \max_i e^{x_i} && (44) \\
&\leq \ln \sum_i e^{x_i}. && (45)
\end{aligned}$$

Now we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{t} \max_{\phi \in \Phi} R_t^{\phi} \right] &\leq \frac{1}{t}\ \mathbb{E}\left[\frac{1}{\eta} \ln \sum_{\phi \in \Phi} e^{\eta R_t^{\phi}} \right] && (46) \\
&= \frac{1}{t}\ \mathbb{E}\left[G(R_t^{\Phi}) \right] && (47) \\
&\leq \frac{1}{t} \left(G(0) + t \max_{r,a} \gamma(\rho^{\Phi}(r,q)) \right) && (48) \\
&\leq \frac{\ln m}{\eta t} + \frac{\eta}{2} && (49)
\end{aligned}$$

Line (48) follows by Corollary 7. ■

$f_i(x)$	Condition	Bound for finite $\Phi \subseteq \Phi_{\text{ALL}}$	Bound for Φ_{EXT},Φ_{INT}
$(x_i^+)^{p-1}$	$p \geq 2$	$\sqrt{\frac{p-1}{t}}\sqrt[p]{m}$	$\sqrt{\frac{p-1}{t}}\sqrt[p]{n-1}$
$(x_i^+)^{p-1}$	$1 < p \leq 2$	$t^{\left(\frac{1}{p}-1\right)}\sqrt[p]{m}$	$t^{\left(\frac{1}{p}-1\right)}\sqrt[p]{n-1}$
$e^{\eta x_i}$	$\eta > 0$	$\frac{\ln m}{\eta t} + \frac{\eta}{2}$	$\frac{\ln m}{\eta t} + \frac{\eta}{2}$

Table 1: Bounds for polynomial and exponential regret-matching algorithms ($n = |A|$ and $m = |\Phi|$).

5.3 Summary of Bounds

We have considered two classes of algorithms, polynomial and exponential regret matching, each of which has a single parameter, p and η, respectively. Table 1 summarizes the bounds we derived on $\mathbb{E}\left[\max_{\phi\in\Phi}\frac{1}{t}R_t^\phi\right]$. Our two analyses of polynomial Φ-regret matching (Theorems 11 and 13) agree when $p = 2$. In addition, for finite $\Phi \subseteq \Phi_{\text{ALL}}$, our bounds on polynomial Φ-regret matching for $p \geq 2$ agree with Cesa-Bianchi and Lugosi [2003]. For polynomial external and internal regret matching, however, we improve on the bounds that can be immediately derived from their results. Though the improvement is small for external regret matching (from a bound proportional to $\sqrt[p]{n}$ to a bound proportional to $\sqrt[p]{n-1}$, where $n = |A|$), it is more significant for internal regret matching (from $n^{(2/p)}$ to $(n-1)^{(1/p)}$).

Finally, observe that for any finite $\Phi \subseteq \Phi_{\text{ALL}}$, polynomial regret matching has the property that

$$\lim_{t\to\infty}\mathbb{E}\left[\max_{\phi\in\Phi}\frac{1}{t}R_t^\phi\right] = 0 \tag{50}$$

while exponential Φ-regret matching has the property that

$$\lim_{t\to\infty}\mathbb{E}\left[\max_{\phi\in\Phi}\frac{1}{t}R_t^\phi\right] = \frac{\eta}{2} \tag{51}$$

In particular, the bound for any polynomial algorithm is eventually better than the bound than for exponential algorithm.

5.4 Optimal Parameters

In this section, we consider the task of setting the parameters in both the polynomial and exponential regret-matching algorithms.

The bound for polynomial regret matching derived in Theorem 13 for $1 < p \leq 2$ is strictly decreasing in p; hence $p = 2$ is the optimal setting. By considering the partial derivative with respect to p of the bound derived in Theorem 11 for $p \geq 2$, we find that we can minimize this bound by setting $p = p^*(\Phi)$, where $p^*(\Phi) = \ln\mu(\Phi) + \sqrt{\ln^2\mu(\Phi) - 2\ln\mu(\Phi)}$ for $\mu(\Phi) \geq e^2$, and $p^*(\Phi) = 2$ otherwise. Combining these results, we see that in fact $p = p^*(\Phi)$ is optimal for all $p > 1$.[5] Cesa-Bianchi and Lugosi [2003] suggest setting $p = 2\ln|\Phi|$, which is suboptimal even when $\mu(\Phi) = |\Phi|$. Still, if we choose $p = 2\ln\mu(\Phi)$, although this choice is suboptimal, as Gentile [2003] observes, it yields a simpler bound, which differs from the optimal only by lower order terms.

Considering the partial derivative with respect to the parameter η of the bound for exponential regret matching derived in Theorem 15, we find that the optimal setting of η depends on the time t at which we want to optimize the bound. The best bound at time t^* is obtained by setting $\eta = \eta^*(\Phi, t^*)$, where

$$\eta^*(\Phi, t) = \sqrt{\frac{2\ln|\Phi|}{t}}. \tag{52}$$

[5] This result is very similar to the optimal parameter for the p-norm regression algorithm calculated by Gentile [2003].

Plugging $\eta^*(\Phi, t^*)$ into Equation 42 we find that an optimized exponential regret-matching algorithm guarantees

$$\mathbb{E}\left[\max_{\phi\in\Phi} \frac{1}{t} R^{\phi}_{t^*}\right] \leq \sqrt{\frac{2\ln|\Phi|}{t^*}}. \tag{53}$$

(This result was obtained by Cesa-Bianchi and Lugosi [2003].) For large enough action sets ($|A| \geq 4$ for Φ_{EXT}, $|A| \geq 13$ for Φ_{INT}), an $\eta^*(\Phi, t^*)$ exponential algorithm will have a lower bound than any polynomial algorithm at $t = t^*$. For small action sets, however, an optimal polynomial algorithm will have a lower bound than any exponential algorithm for all t.

6 Future Work

In ongoing work, we are exploring alternative link functions. For example, we are studying a class of link functions that constitute a spectrum from the polynomial to the exponential. We also hope to further generalize our framework to accommodate both link functions and transformations that vary over time. Finally, we plan to investigate regret-matching algorithms for naïve online decision problems.

Appendix

Lemma 16 *Let $p \geq 2$, and suppose $x, y \in R^n$ such that $\vec{0}$ does not lie on the line segment between them (inclusive). Then* (16) *is satisfied by $G(x) = \|x^+\|_p^2$,*

$$g_i(x) = \begin{cases} 0 & \textit{if } x = \vec{0} \\ \frac{2(x_i^+)^{p-1}}{\|x\|_p^{p-2}} & \textit{otherwise} \end{cases} \tag{54}$$

and $\gamma(x) = (p-1)\|x\|_p^2$. [6]

Proof Let U be an open convex set containing x and $x + y$ but not $\vec{0}$ (e.g., take the union of small enough open balls along the line segment between them). Observe that G is twice continuously differentiable on U and $\nabla G = g$. By Taylor expansion,

$$G(x+y) = G(x) + g(x) \cdot y + \frac{1}{2}\sum_{i,j} \left.\frac{\partial^2 G}{\partial x_i \partial x_j}\right|_u y_i y_j \tag{55}$$

for some u.

By calculation

$$\left.\frac{\partial^2 G}{\partial x_i \partial x_j}\right|_u = 2(2-p)(u_i^+)^{p-1}(u_j^+)^{p-1}\|(u)^+\|_p^{2-2p} \text{ for } i \neq j \tag{56}$$

and

$$\left.\frac{\partial^2 G}{\partial x_i^2}\right|_u = 2(2-p)(u_i^+)^{2p-2}\|(u)^+\|_p^{2-2p} + 2(p-1)(u_i^+)^{p-2}\|(u)^+\|_p^{2-p} \tag{57}$$

[6] Cesa-Bianchi and Lugosi [2003] fail to consider that $G(x) = \|x^+\|_p^2$ is not differentiable at $\vec{0}$ and therefore Taylor's theorem does not apply when $\vec{0}$ lies between x and y.

Then

$$\begin{aligned}
\frac{1}{2}\sum_{i,j}\left.\frac{\partial^2 G}{\partial x_i \partial x_j}\right|_u y_i y_j &= \sum_{i,j}(2-p)(u_i^+)^{p-1}(u_j^+)^{p-1}\|u^+\|_p^{2-2p} y_i y_j \\
&\quad +\sum_i (p-1)(u_i^+)^{p-2}\|u^+\|_p^{2-p} y_i^2 \qquad (58)\\
&= (2-p)\|u^+\|_p^{2-2p}(\sum_i (u_i^+)^{p-1} y_i)^2 \\
&\quad +(p-1)\sum_i (u_i^+)^{p-2}\|u^+\|_p^{2-p} y_i^2 \qquad (59)\\
&\leq (p-1)\|u^+\|_p^{2-p}\sum_i (u_i^+)^{p-2} y_i^2 \qquad (60)\\
&\leq (p-1)\|u^+\|_p^{2-p}\left(\sum_i \left((u_i^+)^{p-2}\right)^{\frac{p}{p-2}}\right)^{\frac{p-2}{p}}\left(\sum_i |y_i|^p\right)^{2/p} \qquad (61)\\
&= (p-1)\|y\|_p^2 \qquad (62)
\end{aligned}$$

Line (58) follows from equations (56) and (57). Line (60) follows because $p \geq 2$. For $p > 2$, line (61) follows from Hölder's inequality, and then line (62) follows algebraically. For $p = 2$, line (62) follows directly from line (60). ■

Lemma 10 *For $p \geq 2$ define $G(x) = \|x^+\|_p^2$,*

$$g_i(x) = \begin{cases} 0 & \text{if } x = 0, \\ \frac{2(x_i^+)^{p-1}}{\|x\|_p^{p-2}} & \text{otherwise,} \end{cases}$$

and $\gamma(x) = (p-1)\|x\|_p^2$. Then $\langle G, g, \gamma \rangle$ is a Gordon triple.

Proof If $\vec{0}$ does not lie on the line segment between x and $z = x + y$, then apply Lemma 16. Otherwise $\exists \lambda \in [0,1]$ such that $\lambda x + (1-\lambda)z = \vec{0}$

- Case 1: $z \in \mathbb{R}^m_{\leq 0}$.
  Then $G(x+y) = G(z) = G(\vec{0}) = 0$. It suffices to prove that

$$||x^+||_p^2 + (p-1)||y||_p^2 + 2\frac{(x^+)^{p-1}\cdot y}{||x||_p^{p-2}} \geq 0 \qquad (63)$$

  which we can reduce to

$$||x^+||_p^p + (p-1)||y||_p^2||x||_p^{p-2} + 2(x^+)^{p-1}\cdot y \geq 0 \qquad (64)$$

  By Holder's inequality

$$||y||_p^2||x||_p^{p-2} \geq \sum_i |y_i|^2 \cdot |x_i|^{p-2} \qquad (65)$$

  So it suffices to prove for any $a, b \in \mathbb{R}$

$$(a^+)^p + (p-1)b^2 \cdot |a|^{p-2} + 2(a^+)^{p-1}\cdot b \geq 0 \qquad (66)$$

In fact we have

$$\begin{aligned}
& (a^+)^p + (p-1)b^2 \cdot |a|^{p-2} + 2(a^+)^{p-1} \cdot b && (67)\\
\geq \; & (a^+)^p + (p-1)b^2 \cdot (a^+)^{p-2} + 2(a^+)^{p-1} \cdot b && (68)\\
= \; & (a^+)^{p-2}((a^+)^2 + (p-1)b^2 + 2a^+ \cdot b) \; (p > 2) && (69)\\
\geq \; & (a^+)^{p-2}((a^+)^2 + b^2 + 2a^+ \cdot b) && (70)\\
= \; & (a^+)^{p-2}(a^+ + b)^2 && (71)\\
\geq \; & 0 && (72)
\end{aligned}$$

- Case 2: $z \in \mathbb{R}^m_{\geq 0}$.
  Then $x \in \mathbb{R}^m_{\leq 0}$, so $G(x) = 0$ and $g(x) = \vec{0}$. Also $G(x+y) \leq G(y) \leq (p-1)\|y\|_p^2$

- Case 3: $z \notin \mathbb{R}^m_{\geq 0} \cup \mathbb{R}^m_{\leq 0}$.
  Observe $x \notin \mathbb{R}^m_{\geq 0} \cup \mathbb{R}^n_{\leq 0}$, so $\vec{0}$ does not lie between x^+ and z. Now we can apply Lemma 16 to x^+ and $z - x^+$, yielding
  $$G(z) \leq G(x^+) + g(x^+) \cdot (z - x^+) + \gamma(z - x^+) \tag{73}$$
  $G(x^+) = G(x)$. Also, for j such that $x_j^+ \neq x_j$, $x_j \leq 0$ so $g_j(x^+) = g_j(x) = 0$. Thus $g_j(x^+)(z_j - x_j^+) \leq g_j(x)(z_j - x_j)$ and $g(x^+) \cdot (z - x^+) = g(x) \cdot y$. Additionally $y_j = z_j - x_j \geq z_j - x_j^+$ for all j so $\|z - x^+\| \leq \|y\|$. We get
  $$G(x^+) + g(x^+) \cdot (z - x^+) + \gamma(z - x^+) \leq G(x) + g(x) \cdot y + \gamma(y) \tag{74}$$
  which gives us the desired result. ■

Lemma 12 *For $1 < p \leq 2$ define $G(x) = \|x^+\|_p^p$, $g_i(x) = p(x_i^+)^{p-1}$, and $\gamma(x) = \|x\|_p^p$. Then $\langle G, g, \gamma \rangle$ is a Gordon triple.*

Proof Because $\|x^+\|_p^p = \sum_i \left(x_i^+\right)^p$, it suffices to show that for any $a, b \in \mathbb{R}$, $((a+b)^+)^p \leq (a^+)^p + p(a^+)^{p-1}b + |b|^p$ and obtain the desired result from a component-wise proof.

- Case 1: $b \geq 0$. Define the function $h_c(z) = z^p + p(c^+)^{p-1}z + p(c^+) - ((c+z)^+)^p$ for any fixed c. Then $h'_c(z) = pz^{p-1} + p(c^+)^{p-1} - p((c+z)^+)^{p-1}$. We use the basic inequality $x^\alpha + y^\alpha \geq (x+y)^\alpha$ for $x > 0, y > 0, 0 \leq \alpha \leq 1$. For $z \geq 0$ the inequality yields $h'_c(z) \geq 0$, so h_c is a non-decreasing function on $[0, \infty)$ and $h_a(b) \geq h_a(0)$. The conclusion follows.

- Case 2: $b \leq 0$. If suffices to prove that $((a-d)^+)^p \leq (a^+)^p - p(a^+)^{p-1}d + d^p$ for $d \geq 0$. We define $h_c(z) = z^p - p(c^+)^{p-1}z + (c^+)p - ((c-z)^+)^p$ and proceed as in case 1.

■

Lemma 14 *For $\eta > 0$ define $G(x) = \frac{1}{\eta}\ln\left(\sum_i e^{\eta x_i}\right)$, $g_i(x) = \frac{e^{\eta x_i}}{\sum_j e^{\eta x_j}}$, and $\gamma(x) = \frac{\eta}{2}\|x\|_\infty^2$. Then $\langle G, g, \gamma \rangle$ is a Gordon triple.*

Proof We use the same technique as in the proof of Lemma 10. Observe that G is smooth and $\nabla G = g$. By Taylor expansion,
$$G(x+y) = G(x) + g(x) \cdot y + \frac{1}{2}\sum_{i,j} \left.\frac{\partial^2 G}{\partial x_i \partial x_j}\right|_u y_i y_j \tag{75}$$
for some u.

By calculation we obtain

$$\left.\frac{\partial^2 G}{\partial x_i \partial x_j}\right|_u = -\frac{\eta e^{\eta u_i} e^{\eta u_j}}{(\sum_i e^{\eta u_i})^2} \text{ for } i \neq j \tag{76}$$

and

$$\left.\frac{\partial^2 G}{\partial x_i^2}\right|_u = -\frac{\eta e^{\eta u_i} e^{\eta u_i}}{(\sum_i e^{\eta u_i})^2} + \frac{\eta e^{\eta u_i}}{\sum_i e^{\eta u_i}} \tag{77}$$

then

$$\begin{aligned}
\frac{1}{2}\sum_{i,j} \left.\frac{\partial^2 G}{\partial x_i \partial x_j}\right|_u y_i y_j &= \sum_{i,j} -\frac{\eta e^{\eta u_i} e^{\eta u_j}}{(\sum_i e^{\eta u_i})^2} y_i y_j + \sum_i \frac{\eta e^{\eta u_i}}{\sum_i e^{\eta u_i}} y_i^2 \\
&= -\eta \left(\frac{\sum_i e^{\eta u_i} y_i}{\sum_i e^{\eta u_i}} \right)^2 + \sum_i \frac{\eta e^{\eta u_i}}{\sum_i e^{\eta u_i}} y_i^2 \\
&\leq \sum_i \frac{\eta e^{\eta u_i}}{\sum_i e^{\eta u_i}} y_i^2 \\
&\leq \sum_i \frac{\eta e^{\eta u_i}}{\sum_i e^{\eta u_i}} \|y\|_\infty^2 \\
&= \eta \|y\|_\infty^2
\end{aligned}$$

■

Acknowledgements

We thank David Gondek for insightful discussions.

References

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 322–331. ACM Press, November 1995.

David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6: 1–8, 1956.

Avrim Blum and Yishay Mansour. From internal to external regret. In *COLT '05: Proceedings of the Eighteenth Annual Conference on Computational Learning Theory*, 2005.

Nicolò Cesa-Bianchi and Gábor Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261, 2003.

D. Foster and R. Vohra. Regret in the on-line decision problem, 1995.

Dean Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29: 7–35, 1999.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Computational Learing Theory*, pages 325–332, 1996.

Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 334–343, 1997.

Claudio Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.

Geoffrey J. Gordon. Regret bounds for prediction problems. In *COLT '99: Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 29–40, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-167-4. doi: http://doi.acm.org/10.1145/307400.307410.

Amy Greenwald and Amir Jafari. A general class of no-regret algorithms and game-theoretic equilibria. In *Proceedings of the 2003 Computational Learning Theory Conference*, pages 1–11, August 2003.

J. Hannan. Approximation to Bayes risk in repeated plays. In M. Dresher, A.W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.

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

Ehud Lehrer. A wide range no-regret theorem. *Games and Economic Behavior*, 42(1):101–115, 2003.

Casey Marks, Amy Greenwald, and David Gondek. Varieties of regret in online prediction. Technical Report CS-04-09, Department of Computer Science, Brown University, July 2004.