More Efficient Internal-Regret-Minimizing Algorithms

Amy Greenwald, Zheng Li, and Warren Schudy
Brown University, Providence, RI 02912
{amy,ws}@cs.brown.edu and zheng@dam.brown.edu

Abstract

Standard no-internal-regret (NIR) algorithms compute a fixed point of a matrix, and hence typically require $O(n^3)$ run time per round of learning, where n is the dimensionality of the matrix. The main contribution of this paper is a novel NIR algorithm, which is a simple and straightforward variant of a standard NIR algorithm. However, rather than compute a fixed point every round, our algorithm relies on power iteration to estimate a fixed point, and hence runs in $O(n^2)$ time per round.

Nonetheless, it is not enough to look only at the per-round run time of an online learning algorithm. One must also consider the algorithm's convergence rate. It turns out that the convergence rate of the aforementioned algorithm is slower than desired. This observation motivates our second contribution, which is an analysis of a multithreaded NIR algorithm that trades-off between its run time per round of learning and its convergence rate.

1 Introduction

An *online decision problem* (ODP) consists of a series of rounds, during each of which an agent chooses one of n pure actions and receives a reward corresponding to its choice. The agent's objective is to maximize its cumulative rewards. It can work towards this goal by abiding by an *online learning algorithm*, which prescribes a possibly mixed action (i.e., a probability distribution over the set of pure actions) to play each round, based on past actions and their corresponding rewards. The success of such an algorithm is typically measured in a worst-case fashion: specifically, an adversary chooses the sequence of rewards that the agent faces. Hence, the agent—the protagonist—*must* randomize its play; otherwise, it can easily be exploited by the adversary.

The observation that an ODP of this nature can be used to model a single player's perspective in a repeated game has spawned a growing literature connecting computational learning theory—specifically, the subarea of regret minimization—and game theory—specifically, the subarea of learning in repeated games. Both groups of researchers are interested in designing algorithms by which an agent can learn from its past actions, and the rewards associated with those actions, to play actions now and in the future that yield high rewards. More specifically, the entire sequence of actions should yield low regret for not having played otherwise, or equivalently, near equilibrium behavior.

In a seminal paper by Foster and Vohra [FV97], it was established that the empirical distribution of the joint play of a particular class of online learning algorithms, called no-internal-regret (NIR) learners, converges to the set of correlated equilibria in repeated matrix games. However, standard NIR learning algorithms (see Cesa-Bianchi and Lugosi [CBL06] and Blum and Mansour [BM05])[1]—including the algorithm proposed by Foster and Vohra (hereafter, FV)—involve a fixed point calculation during each round of learning, an operation that is cubic[2] in the number of pure actions available to the player. Knowing that fixed point calculations are expensive, Hart and Mas-Colell [HMC00] describe "a simple adaptive procedure" (hereafter, HM) that also achieves the aforementioned convergence result. HM's per-round run time is linear in the number of pure actions.

It is well-known [HMC00] that HM does not exhibit no internal regret in the usual sense, meaning against an *adaptive* adversary—one that can adapt in response to the protagonist's "realized" pure actions (i.e., those that result from sampling his mixed actions). Still, in a recent paper, Cahn [Cah04] has shown that HM's algorithm does exhibit no internal regret against an adversary that is "not too sophisticated." In this paper, we use the terminology *nearly oblivious* to refer to this

[1] The former reference is to a book that surveys the field; the latter reference is to a paper that includes a black-box method for constructing NIR learners from another class of learners called no-external-regret learners.

[2] Strassen [Str69] devised an $O(n^{2.81})$-time algorithm for matrix-matrix multiplication, based on which a fixed point can be computed with the same run time [CLRS01]. Coppersmith and Winograd [CW87] devised an $O(n^{2.36})$-time algorithm for matrix-matrix multiplication, but unlike Strassen's result their result is impractical. For better pedagogy, we quote the "natural" $O(n^3)$ runtime in most of our discussions rather than these better bounds.

type of adversary, because the "not-too-sophisticated" condition is a weakening of the usual notion of an *oblivious* adversary—one who chooses the sequence of rewards after the protagonist chooses its online learning algorithm, but before the protagonist realizes any of its pure actions. Since an oblivious adversary is also nearly oblivious, Cahn's result implies that HM exhibits no internal regret against an oblivious adversary.

As alluded to above, both FV and HM (and all the algorithms studied in this paper) learn a mixed action each round, and then play a pure action: i.e., a sample from that mixed action. One important difference between them, however, which can be viewed at least as a partial explanation of their varying strengths, is that FV maintains as its state the mixed action it learns, whereas HM maintains as its state the pure action it plays. Intuitively, the latter cannot exhibit no internal regret against an adaptive adversary because an adaptive adversary can exploit any dependencies between the consecutively sampled pure actions.

Young [You04] proposes, but does not analyze rigorously, a variant of HM he calls Incremental Conditional Regret Matching (ICRM), which keeps track of a mixed action instead of a pure action, and hence exhibits no internal regret against an adaptive adversary.[3] ICRM has quadratic run time each round. To motivate ICRM, recall that standard NIR algorithms involve a fixed-point calculation. Specifically, they rely on solutions to equations of the form $q = qP_t$, where P_t is a stochastic matrix that encodes the learner's regrets for its actions through time t. Rather than solve this equation exactly, ICRM takes $q_{t+1} \leftarrow q_t P_t$ as an iterative approximation of the desired fixed point.

The regret matrix P_t used in ICRM (and HM) depends on a parameter μ that is strictly larger than the maximum regret per round. This makes ICRM less intuitive than it could be. We show that the same idea also works when the normalizing factor μt is replaced by the actual total regret experienced by the learner. This simplifies the algorithm and eliminates the need for the learner to know or estimate a bound on the rewards. We call our algorithm Power Iteration (PI),[4] because another more intuitive way to view it is as a modification of a standard NIR algorithm (e.g., Greenwald, *et al.* [GJMar]) with its fixed-point calculation replaced by power iteration. Once again, the first (and primary) contribution of this paper is a proof that *using power iteration to estimate a fixed point, which costs only $O(n^2)$ per round, suffices to achieve no-internal-regret against an adaptive adversary.*

Although our PI algorithm is intuitive, the proof that the idea pans out—that PI exhibits NIR against an adaptive adversary—is non-trivial (which may be why Young did not propose this algorithm in the first place). The proof in Hart and Mas-Colell [HMC00] relies on a technical lemma, which states that $\|q_t P_t^z - q_t P_t^{z-1}\|_1$, for some $z > 0$, is small, whenever all the entries on the main diagonal of P_t are at least some uniform constant. With our new definition of P_t, this condition does not hold. Instead, our result relies on a generalization of this lemma in which we pose weaker conditions that guarantee the same conclusion. Specifically, we require only that the trace of P_t be at least $n - 1$. Our lemma may be of independent interest.

Hence, we have succeeded at defining a simple and intuitive, $O(n^2)$ per-round online learning algorithm that achieves no internal regret against an adaptive adversary. However, it is not enough to look only at the per-round run time of an online learning algorithm. One must also consider the algorithm's convergence rate. It turns out that the convergence rates of PI, ICRM, and HM are all slower than desired (their regret bounds are $O(\sqrt{n}t^{-1/10})$), whereas FV's regret bound is $O(\sqrt{n/t})$ (see, for example, Greenwald, *et al.* [GLM06]). This observation motivates our second algorithm.

As our second contribution, we analyze an alternative algorithm, one which is multithreaded. Again, the basic idea is straightforward: one thread plays the game, taking as its mixed action the most-recently computed fixed point, while the other thread computes a new fixed point. Whenever a new fixed point becomes available, the first thread updates its mixed action accordingly. This second algorithm, which we call MT, for multithreaded, exhibits a trade-off between its run time per round and its convergence rate. If p is an upper bound on the number of rounds it takes to compute a fixed point, MT's regret is bounded by $O(\sqrt{np/t})$. Observe that this regret bound is a function of t/p, the number of fixed points computed so far. If p is small, so that many fixed points have been computed so far, then the run time per round is high, but the regret is low; on the other hand, if p is large, so that only very few fixed points have been computed so far, then the run time per round is low, but the regret is high.

This paper is organized as follows. In Section 2, we define online decision problems and no-regret learning precisely. In Section 3, we define the HM, ICRM, and PI algorithms, and report their regret bounds. In Section 4, we introduce our second algorithm, MT, and report its regret bound. In Section 5, we prove a straightforward lemma that we use in the analysis of all algorithms. In Section 6, we analyze MT. In Section 7, we analyze PI. In Section 8, we present some preliminary simulation experiments involving PI, HM, and MT. In Section 9, we describe some interesting future directions.

2 Formalism

An online decision problem (ODP) is parameterized by a *reward system* $(A, \mathcal{R})$, where A is a set of pure actions and $\mathcal{R}$ is a set of rewards. Given a reward system $(A, \mathcal{R})$, we let $\Pi \equiv \mathcal{R}^A$ denote the set of possible reward vectors.

[3] Our analytical tools can be used to establish Young's claim rigorously.

[4] Both PI and ICRM can be construed as both incremental conditional regret matching algorithms and as power iteration methods. The difference between these algorithms is merely the definition of the matrix P_t, and who named them, not what they are named for per se.

Definition 1 *Given a reward system $(A, \mathcal{R})$, an* online decision problem *can be described by a sequence of reward functions $\langle \tilde{\pi}_t \rangle_{t=1}^{\infty}$, where $\tilde{\pi}_t \in (A^{t-1} \mapsto \Pi)$.*

Given an ODP $\langle \tilde{\pi}_t \rangle_{t=1}^{\infty}$, the particular history $H_t = (\langle a_\tau \rangle_{\tau=1}^{t}, \langle \pi_\tau \rangle_{\tau=1}^{t})$ corresponds to the agent playing a_τ and observing reward vector $\pi_\tau \equiv \tilde{\pi}_\tau(a_1, \ldots a_{\tau-1})$ at all times $\tau = 1, \ldots, t$.

In this paper, we restrict our attention to bounded, real-valued reward systems; as such, we assume WLOG that $\mathcal{R} = [0, 1]$. We also assume the agent's pure action set is finite; specifically, we let $|A| = n$. Still, we allow agents to play mixed actions. That is, an agent can play a probability distribution over its pure actions. We denote by $\Delta(A)$ the set of mixed actions: i.e., the set of all probability distributions over A.

An *online learning algorithm* is a sequence of functions $\langle \tilde{q}_t \rangle_{t=1}^{\infty}$, where $\tilde{q}_t : H_{t-1} \to \Delta(A)$ so that $\tilde{q}_t(h) \in \Delta(A)$ represents the agent's mixed action at time $t \geq 1$, after having observed history $h \in H_{t-1}$. When the history h is clear from context, we abbreviate $\tilde{q}_t(h)$ by q_t. For a given history of length t, let $\hat{q}_t$ be the degenerate probability distribution corresponding to the action actually played at time t: i.e., for all $1 \leq i \leq n$, $(\hat{q}_t)_i = \mathbb{1}(a_t = i)$.[5] Clearly, $\hat{q}_t$ is a random variable.

We are interested in measuring an agent's regret in an ODP for playing as prescribed by some online learning algorithm rather than playing otherwise. We parameterize this notion of "otherwise" by considering a variety of other ways that the agent could have played. For example, it could have played any single action a all the time; or, it could have played a' every time it actually played a. In either case, we arrive at an alternative sequence of play by applying some transformation to each action in the agent's actual sequence of play, and then we measure the difference in rewards obtained by the two sequences, in the worst case. That is the agent's regret.

A transformation of the sort used in the first example above—a constant transformation that maps every action a' in the actual sequence of play to a fixed, alternative action a—is called an *external* transformation. We denote by Φ_{EXT} the set of all external transformations, one per action $a \in A$. Many efficient algorithms, with both fast run time per round and fast convergence rates, are known to minimize regret with respect to Φ_{EXT} (e.g., [LW94, FS97, HMC01]). Here, we are interested in transformations of the second type, which are called *internal* transformations. These transformations can be described by the following set of n-dimensional matrices:

$$\Phi_{\text{INT}} = \{\, \phi^{(a,b)} \;:\; a \neq b, 1 \leq a, b \leq n \,\}$$

where

$$(\phi^{(a,b)})_{ij} = \begin{cases} 1 & \text{if } i \neq a \wedge i = j \\ 1 & \text{if } i = a \wedge j = b \\ 0 & \text{otherwise} \end{cases}$$

For example, if $|A| = 4$, then applying the following transformation to a pure action a yields the third action if a is the second action, and a otherwise:

$$\phi^{(2,3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Φ_{EXT} and Φ_{INT} are the two best-known examples of transformation sets. More generally, a transformation ϕ can be any linear function from $\Delta(A) \to \Delta(A)$. In the definitions that follow, we express reward vectors π as column vectors, mixed actions q as row vectors, and transformations ϕ as n-dimensional matrices.

If, at time τ, an agent plays mixed action q_τ in an ODP with reward vector π_τ, the agent's *instantaneous regret* $(r_\tau)_\phi$ with respect to a transformation ϕ is the difference between the rewards it could have obtained by playing $q_\tau \phi$ and the rewards it actually obtained by playing q_τ: i.e.,

$$(r_\tau)_\phi = q_\tau \phi \pi_\tau - q_\tau \pi_\tau \qquad (1)$$

The agent's *cumulative regret vector* (R_t) through time t is then computed in the obvious way: for $\phi \in \Phi$,

$$(R_t)_\phi = \sum_{\tau=1}^{t} (r_\tau)_\phi \qquad (2)$$

One can also define *pure action* variants of the instantaneous and cumulative regret vectors, as follows:

$$(\hat{r}_\tau)_\phi = \hat{q}_\tau \phi \pi_\tau - \hat{q}_\tau \pi_\tau \qquad (3)$$

and

$$(\hat{R}_t)_\phi = \sum_{\tau=1}^{t} (\hat{r}_\tau)_\phi \qquad (4)$$

One can bound either the expected pure action regret or the (mixed action) regret. To avoid unilluminating complications, we focus on the latter in this work.

Our objective in this work is to establish sublinear bounds on the average internal-regret vector of various online learning algorithms. Equipped with such bounds, we can then go on to claim that our algorithms exhibit no internal regret by applying standard techniques such as the Hoeffding-Azuma lemma (see, for example, Cesa-Bianchi and Lugosi [CBL06]). Note that we cannot establish our results for general Φ. We defer further discussion of this point until Section 9, where we provide a simple counterexample.

For completeness, here is the formal definition of no-Φ-regret learning:

Definition 2 *Given a finite set of transformations Φ, an online learning algorithm $\langle \tilde{q}_t \rangle_{t=1}^{\infty}$ is said to exhibit* **no-Φ-regret** *if for all $\epsilon > 0$ there exists $t_0 \geq 0$ such that for any ODP $\langle \tilde{\pi}_t \rangle_{t=1}^{\infty}$,*

$$\Pr\left[\exists t > t_0 \; s.t. \; \max_{\phi \in \Phi} \frac{1}{t} \hat{R}_t^{\phi} \geq \epsilon\right] < \epsilon \qquad (5)$$

[5] For predicate p, $\mathbb{1}(p) = \begin{cases} 1 & \text{if } p \\ 0 & \text{otherwise} \end{cases}$.

The relevant probability space in the above definition is the natural one that arises when considering a particular ODP $\langle \tilde{\pi}_t \rangle_{t=1}^{\infty}$ together with an online learning algorithm $\langle \tilde{q}_t \rangle_{t=1}^{\infty}$. The universe consists of infinite sequences of pure actions $\langle a_\tau \rangle_{\tau=1}^{\infty}$ and the measure is defined by the learning algorithm.

We close this section with some notation that appears in later sections:

- We let $a \bullet b = a^T b$ denote the dot product of column vectors a and b.
- For vector $v \in \mathbb{R}^n$, we let v^+ denote the componentwise max of v and the zero vector: i.e., $(v^+)_i = \max(v_i, 0)$.

3 Algorithms

We begin this section by describing HM, the simple adaptive procedure due to Hart and Mas-Colell [HMC00] that exhibits no internal regret against a nearly oblivious adversary, as well as ICRM, a variant of HM due to Young [You04] that exhibits no internal regret against an adaptive adversary. We then go on to present a simple variant of these algorithms, which we call PI, for power iteration, for which we establish the stronger of these two guarantees.

Definition 3 *Define the n-dimensional matrix*

$$N_t = \sum_{\phi \in \Phi_{INT}} (R_t^+)_\phi \phi$$

and the scalar

$$D_t = \sum_{\phi \in \Phi_{INT}} (R_t^+)_\phi$$

At a high-level, HM (Algorithm 1) and ICRM (not shown) operate in much the same way: at each time step t, an action is played and a reward is earned; then, the regret matrix P_t is computed in terms of N_t and D_t, based on which a new action is derived. But the algorithms differ in an important way: specifically, they differ in their "state" (i.e., what they store from one round to the next). In HM, the state is a pure action, so that during each round, the next pure action is computed based on the current pure action. In ICRM, the state is a mixed action.

Like Young's algorithm, the state in our algorithm, PI (Algorithm 2), is a mixed action. But, our algorithm differs from both of the others in our choice of the matrix regret P_t. In PI, $P_t = N_t/D_t$, which is the same matrix as in Greenwald *et al.* [GJMar], for example. Intuitively, N_t/D_t is a convex combination of the transformations in Φ_{INT}, with each $\phi \in \Phi_{\text{INT}}$ weighted by the amount of regret the learner experienced for not having transformed its play as prescribed. In HM and ICRM, P_t is a convex combination of N_t/D_t and the identity matrix. This convex combination depends on a parameter μ, which is an upper bound on the regret per round; typically, $\mu = 2n$.

Algorithm 1 HM [HMC00]

```
Initialize $a_1$ to be an arbitrary pure action.

During each round $t = 1, 2, 3, \ldots$:

  1. Play the pure action $a_t$.
  2. For all $j$,
     let $(\hat{q}_t)_j = \mathbb{1}(a_t = j)$.
  3. Observe rewards $\pi_t$.
  4. Update the regret vector $\hat{R}_t$.
  5. Let the regret matrix $\hat{P}_t = \frac{\hat{N}_t + (\mu t - \hat{D}_t) I}{\mu t}$.
  6. Sample a pure action $a_{t+1}$ from $\hat{q}_t \hat{P}_t$.
```

Algorithm 2 Power Iteration

```
Initialize $q_1$ to be an arbitrary mixed action.

During each round $t = 1, 2, 3, \ldots$:

  1. Sample a pure action $a_t$ from $q_t$.
  2. Play the pure action $a_t$.
  3. Observe rewards $\pi_t$.
  4. Update the regret vector $R_t$.
  5. Define the regret matrix $P_t = \frac{N_t}{D_t}$.
  6. Set the mixed action $q_{t+1} \leftarrow q_t P_t$.
```

HM has a per-round run time linear in the number of pure actions because it updates one row of the regret matrix during each round, namely that row corresponding to the action played. ICRM and PI both have per-round run times dominated by the matrix-vector multiplication in Step 6, and are hence quadratic in the number of pure actions.

We analyze PI (Algorithm 2) in this paper, and obtain the following result:

Theorem 4 *PI (Algorithm 2) exhibits no internal regret against an* adaptive *adversary. Specifically, the bound on its average regret is as follows: for all times t,*

$$\left\| \frac{R_t^+}{t} \right\|_\infty \leq O(\sqrt{n} t^{-1/10})$$

A slight variant of our analysis shows that ICRM has the same bound as PI. Algorithm 1 was previously analyzed by Cahn [Cah04], who showed that if the adversary is nearly oblivious, then HM exhibits NIR. One can combine ideas from Hart and Mas-Colell [HMC00] and our analysis of PI to show that against an oblivious adversary, the bound on HM's average regret is as follows: for all times t,

$$\mathbb{E}\left[\left\|\frac{\hat{R}_t^+}{t}\right\|_\infty\right] \leq O(\sqrt{n}t^{-1/10})$$

Theorem 4 implies, by the Proposition at the bottom of page 1133 in Hart and Mas-Colell [HMC00], that like HM and ICRM, PI also converges to the set of correlated equilibria in self-play.

Note that it is also possible to define a variant of PI with $P_t = N_t/D_t$, which like HM, uses the agent's pure action as its state. We conjecture that this algorithm would exhibit no internal regret against an oblivious (or nearly oblivious) adversary, but do not analyze it because it has no obvious advantages over HM or PI.

4 Multithreaded algorithm

The ICRM and PI algorithms have better per-round run times than standard NIR learning algorithms, but their convergence rates are far worse. Moreover, these algorithms are inflexible: they cannot expend additional effort during a round to improve their convergence rates. In this section, we present a parameterized, multithreaded algorithm (MT) that smoothly trades off between per-round run time and regret.

The idea underlying MT is simply to spread the computation of a fixed point over many time steps, and in the mean time to play the most recent fixed point computed so far. This idea is formalized in Algorithm 3, in which there are two threads. One thread plays the game, taking as its mixed action the most-recently computed fixed point; the other thread works towards computing a new fixed point.

Theorem 5 *Let $p \geq 1$ be an upper bound on how many time steps it takes to compute a fixed point. MT (Algorithm 3) has per-round run time $O(LS(n)/p + \log n + \rho)$ and regret bound $O(\sqrt{np/t})$, where $LS(n)$ required solve a linear system of equations expressed as an n-dimensional matrix and ρ is the usually negligible run time required to maintain the regret vector (see Section 6). More precisely, for all times t,*

$$\left\|\frac{R_t^+}{t}\right\|_\infty \leq \sqrt{\frac{(n-1)(4p-3)}{t}}$$

Suppose you are playing a game every minute and you have just barely enough computational resources to find a fixed point in the time alloted with a standard NIR learning algorithm ($p = 1$). Further, suppose that it takes 1 day of playing for your regret to fall below a desired threshold. Now, suppose the game changes and you now have to make a move every second. If you set $p = 60$, and continue to compute 1 fixed point per minute, this will require $4 \cdot 60 - 3 \approx 4 \cdot 60$ times more rounds to achieve the same level of regret. But each round is 60 times faster, so the wall-clock time for the same level of regret has increased by a factor of about 4, to a bit under 4 days.

With one extreme parameter setting, namely $p = 1$, MT is just like a standard NIR learning algorithm, and hence has run time $O(n^3)$ per round and regret bound $O(\sqrt{n/t})$. With another extreme parameter setting, namely $p = n^3$, MT has run time $O(\log n)$ per round (as long as regret can be calculated quickly; see the end of Section 6) and regret bound $O(n^2/\sqrt{t})$. The intermediate parameter setting $p = n$ yields an $O(n^2)$ run time per round and an $O(n/\sqrt{t})$ regret bound. This algorithm, therefore, dominates both PI and ICRM, achieving the same run time per round, but a better regret bound, for all values of $t \geq n^{5/4}$.

Algorithm 3 Multithreaded no-internal-regret learning algorithm.

```
Initialize $R$, $N$, $D$ to zero.

First thread: During each round $t = 1, 2, 3, \ldots$:
    • Wait until it is time to take an action.
    • Get the most up-to-date fixed point computed by
      the other thread. Call it $q_t$. (If no fixed point has
      been computed yet, initialize $q_t$ arbitrarily.)
    • Sample a pure action $a_t$ from $q_t$.
    • Play the pure action $a_t$.
    • Observe rewards $\pi_t$.
    • Update the regret vector $R_t$.

Second thread: Repeat forever:
    • Wait until the other thread updates the regret vector
      $R_\tau$ for any $\tau > 0$.
    • Get a copy of $R_\tau$ from the other thread.
    • Compute $N_\tau$ and $D_\tau$.
    • Compute a fixed point of $N_\tau/D_\tau$.
    • Pass this fixed point to the other thread.
```

5 General Analysis

In this section, we derive a key lemma that is used in both our analyses. Specifically, we bound the L_2-norm of the regret vector at time t in terms of two summations from time $\tau = 1$ to t. Each term in the first bounds how close the mixed action played at time τ is to being a fixed point of the regret matrix at some previous time $\tau - w(\tau)$. Each term in the second bounds the regret that could ensue because the mixed action played at time τ is out of date.

Lemma 6 *For any online learning algorithm and any function $w(\cdot) > 0$, we have the following inequality: for* *all times $t > 0$,*

$$\begin{aligned}\left\|R_t^+\right\|_2^2 \quad &\leq \quad 2\sum_{\tau=1}^{t} q_\tau \left(N_{\tau-w(\tau)} - D_{\tau-w(\tau)} I\right) \pi_t \\ &\quad + (n-1)\sum_{\tau=1}^{t}(2w(\tau)-1))\end{aligned}$$

where q_t is the mixed action at time t, and I is the identity matrix.

We prove this lemma using two preliminary lemmas. The first involves simple algebra.

Lemma 7 *For any two vectors $a, b \in \mathbb{R}^d$, with $d \geq 1$, we have the following inequality:*

$$||(a+b)^+||_2^2 \leq ||a^+||_2^2 + 2a^+ \bullet b + ||b||_2^2 \qquad (6)$$

Proof: Both $\|\cdot\|_2^2$ and dot products are additive componentwise, so it suffices to assume a, b are real numbers.

If $a + b \leq 0$ then $|a^+|^2 + 2a^+b + b^2 = (a^+ + b)^2 \geq 0 = |(a+b)^+|^2$.

If $a + b > 0$ then $a^+ + b \geq a + b = (a+b)^+ > 0$. Thus $(a^+ + b)^2 \geq |(a+b)^+|^2$. ■

Lemma 8 *For any learning algorithm and any $t > \tau \geq 0$, we have the following equality:*

$$\begin{aligned}r_t \bullet R_\tau^+ \quad &= \quad q_t \left(N_\tau - D_\tau I\right) \pi_t \\ &= \quad D_\tau q_t \left(\frac{N_\tau}{D_\tau} - I\right) \pi_t\end{aligned}$$

Proof: Standard no-regret arguments about the fixed point (e.g., Theorem 5 in [GLM06]). ■

Note that if q_t is a fixed point of N_τ / D_τ, as in is in FV and MT for appropriate choices of τ, then $r_t \bullet R_\tau^+ = 0$. For example, in the traditional algorithm FV, $r_t \bullet R_{t-1}^+ = 0$.

Proof: [Proof of Lemma 6] Fix a $\tau \in \{1, \ldots, t\}$. By definition, $R_\tau^+ = R_{\tau-1}^+ + r_\tau$. Hence, by applying Lemma 7, we obtain a linear approximation of $\|R_\tau^+\|_2^2 - \left\|R_{\tau-1}^+\right\|_2^2$ with an error term:

$$\begin{aligned}\left\|R_\tau^+\right\|_2^2 \quad &\leq \quad \left\|R_{\tau-1}^+\right\|_2^2 + 2r_\tau \bullet R_{\tau-1}^+ + \|r_\tau\|_2^2 \\ &= \quad \left\|R_{\tau-1}^+\right\|_2^2 + 2r_\tau \bullet R_{\tau-w(\tau)}^+ \\ &\qquad + 2r_\tau \bullet (R_{\tau-1}^+ - R_{\tau-w(\tau)}^+) + \|r_\tau\|_2^2 \\ &\leq \quad \left\|R_{\tau-1}^+\right\|_2^2 + 2r_\tau \bullet R_{\tau-w(\tau)}^+ \\ &\qquad + 2(w(\tau)-1)(n-1) + (n-1) \\ &= \quad \left\|R_{\tau-1}^+\right\|_2^2 + 2r_\tau \bullet R_{\tau-w(\tau)}^+ + (2w(\tau)-1)(n-1) \qquad (7)\end{aligned}$$

The second inequality follows from the fact that $\|r_\tau\|_2^2 \leq (n-1)$.

Now if we apply Lemma 8 and sum over time, this yields:

$$\begin{aligned}&\sum_{\tau=1}^{t}\left(\left\|R_\tau^+\right\|_2^2 - \left\|R_{\tau-1}^+\right\|_2^2\right) \\ &\quad \leq \quad 2\sum_{\tau=1}^{t} q_\tau (N_{\tau-w(\tau)} - D_{\tau-w(\tau)} I) \pi_\tau \\ &\qquad\quad + (n-1)\sum_{\tau=1}^{t}(2w(\tau)-1) \qquad (8)\end{aligned}$$

The summation on the left hand side of this equation collapses to $\left\|R_t^+\right\|_2^2 - \left\|R_0^+\right\|_2^2 = \left\|R_t^+\right\|_2^2$, and the lemma is proved. ■

6 Analysis of MT

Equipped with Lemma 6, the proof of Theorem 5 is quite simple.

Proof: [Proof of Theorem 5] For general p, the fixed points may be based on out-of-date regret vectors, but they are never very out of date. Once the fixed point is computed, it is based on data that is p rounds out of date. That fixed point is then used for another p rounds while a replacement is computed. Overall, the fixed point played at time t can be based on a regret vector no more than $2p$ rounds old. More precisely, the τ such that R_τ is used to compute q_t satisfies $t-(2p-1) \leq \tau \leq t-1$.

Now apply Lemma 6 letting $w(\tau)$ be the age of the regret vector used by the second thread in calculating q_τ. Since q_τ is a fixed point of $N_{\tau-w(\tau)}/D_{\tau-w(\tau)}$, it follows that $q_\tau(N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I) = 0$. Thus,

$$\begin{aligned}\left\|R_t^+\right\|_2^2 \quad &\leq \quad 2\sum_{\tau=1}^{t} 0 + (n-1)\sum_{\tau=1}^{t}(2(2p-1)-1) \\ &= \quad (n-1)t(4p-3)\end{aligned}$$

Therefore,

$$\left\|\frac{R_t^+}{t}\right\|_\infty \leq \left\|\frac{R_t^+}{t}\right\|_2 \leq \sqrt{\frac{(n-1)(4p-3)}{t}}$$

and the theorem is proved. ■

A naïve computation of the regret vector would limit the per-round run time of PI to $\Omega(n^2)$. For applications where p is $O(n)$ (or less), this is not a bottleneck, because in that case the $O(n^3/p)$ bound on the run time of the fixed point computation is larger than the $O(n^2)$ run time of the regret vector updates.

If the ODP is a repeated game where the opponents have $O(n)$ joint actions, an agent can simply record the opponents' actions each round in constant time, and then update the regret vector right before solving for a fixed point; this update takes time $O(n^3)$. In this case, if $p = n^3$, then MT's per-round run time is $O(\log n)$.

For general ODPs, where the reward structure may change arbitrarily from one round to the next, keeping track of regret in time $o(n^2)$ per round seems to require random sampling (i.e., bandit techniques; see, for example, Auer *et al.* [ACBFS02]). We leave further investigation of this issue to future work.

Choosing a random action from a probability distribution using a binary search requires $\Theta(\log n)$ time, so ODPs that require extremely quick decisions cannot be handled without further innovation.

7 Analysis of PI

In this section, we analyze PI. By construction, q_τ is not a fixed point but only an approximate fixed point, so $q_\tau(N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I) \neq 0$. Instead, we will show the following:

Lemma 9 *For all times $\tau > 0$ and $0 < w(\tau) < \tau$, $\left\|q_\tau(N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I)\right\|_1 = O\left(\frac{n\tau}{\sqrt{w(\tau)}} + n(w(\tau))^2\right)$*

Deferring the proof of Lemma 9, we first show how to use Lemmas 6 (choose $w(\tau) = \tau^{2/5}$) and 9, to analyze PI:

$$\begin{aligned}
\left\|R_t^+\right\|_2^2 &\leq 2\sum_{\tau=1}^{t} q_\tau\left(N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I\right)\pi_\tau \\
&\quad + (n-1)\sum_{\tau=1}^{t}(2\tau^{2/5} - 1)) \\
&= \sum_{\tau=1}^{t} O\left(\frac{n\tau}{\sqrt{\tau^{2/5}}} + n(\tau^{2/5})^2\right) + (n-1)O(t^{7/5}) \\
&= O(nt^{9/5}) + (n-1)O(t^{7/5}) \\
&= O(nt^{9/5})
\end{aligned}$$

Taking square roots and dividing by t proves Theorem 4:

$$\left\|\frac{R_t^+}{t}\right\|_2 \leq O(\sqrt{n}t^{-1/10})$$

It remains to prove Lemma 9. For the remainder of this section, we use the shorthands $W \equiv w(\tau)$ and $t = \tau - w(\tau)$.

We begin to analyze $q_\tau\left(N_t - D_t I\right)$ by rewriting this expression as the sum of two terms. The first, which would be zero if power iteration converged in W steps, is provably small. The second measures how the matrices P_T change over time; if all the P_T's were equal, this term would be zero. Noting that $q_\tau = q_t\left(\prod_{T=t}^{\tau-1} P_T\right)$, where each $P_T = N_T/D_T$, we derive the two terms as follows:

$$\begin{aligned}
&q_\tau\left(N_t - D_t I\right) \\
&\quad= D_t q_t\left(\prod_{T=t}^{\tau-1} P_T\right)(P_t - I) \\
&\quad= D_t q_t P_t^W (P_t - I) + \\
&\qquad D_t q_t\left(\left[\prod_{T=t}^{\tau-1} P_T\right] - P_t^W\right)(P_t - I) \\
&\quad= D_t q_t\left(P_t^{W+1} - P_t^W\right) + \\
&\qquad D_t\left(\left[\prod_{T=t}^{\tau-1} P_T\right] - P_t^W\right)(P_t - I) \qquad (9)
\end{aligned}$$

We will bound the two terms in Equation 9 in turn. Beginning with the first, the quantity $q_t P_t^W$ can be interpreted as the distribution of a Markov chain with transition matrix P_t and initial distribution q_t after W time steps. Most Markov chains converge to a stationary distribution, so it is intuitively plausible that the related quantity $q_t\left(P_t^{W+1} - P_t^W\right)$ is small. The following lemma, which verifies this intuition, is a strengthening of statement M7 in Hart and Mas-Colell [HMC00]. Our lemma is stronger because our premises are weaker. Whereas their lemma requires that all the entries on the main diagonal of P_t be at least some uniform constant, ours requires only that the sum of P_t's diagonal entries (i.e., its trace) be at least $n-1$. The latter of these two conditions (only) is satisfied by PI's choice of P_t, because each P_t is a convex combination of internal regret transformations/matrices, each of which has trace $n-1$.

Lemma 10 *For all $z > 0$, if P is n-dimensional stochastic matrix that is close to the identity matrix in the sense that $\sum_{i=1}^n P_{ii} \geq n-1$, then $\left\|q(P^z - P^{z-1})\right\|_1 = O(1/\sqrt{z})$ for all n-dimensional vectors q with $\left\|q\right\|_1 = 1$.*

Proof: See Appendix. ■

Now, we can easily bound $D_t = D_{\tau-W}$ by $(n-1)(\tau-W) \leq n\tau$, so the first term in Equation 9 is bounded above by $O(n\tau/\sqrt{W})$. The following lemma bounds the second term in Equation 9:

Lemma 11 *For all times $\tau > 0$ and $0 < w(\tau) < \tau$,*

$$\left\|q_t\left(\left[\prod_{T=t}^{\tau-1} P_T\right] - P_t^W\right)(P_t - I)\right\|_1 = O(nW^2/D_t)$$

The proof of this lemma makes use of the following definition and related fact: the induced L_1-norm of a matrix M is given by

$$\left\|M\right\|_1 = \max_{v\neq 0}\frac{\left\|vM\right\|_1}{\left\|v\right\|_1}$$

and for any n-dimensional vector v and matrix M,

$$\left\|vM\right\|_1 \leq \left\|v\right\|_1\left\|M\right\|_1 \qquad (10)$$

Proof: Since $\|P_t - I\|_1 \le \|P_t\|_1 + \|I\|_1 = 2$, it follows that

$$\left\| q_t \left(\prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) (P_t - I) \right\|_1 \le 2 \left\| q_t \left(\prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) \right\|_1$$

Hence, it suffices to bound $\left\| q_t \left(\prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) \right\|_1$. To do so, we first note that

$$\begin{aligned} & \prod_{s=0}^{W-1} P_{t+s} - P_t^W \\ = & \sum_{s=0}^{W-1} \left(\prod_{u=0}^{s} P_{t+u} P_t^{W-s-1} - \prod_{u=0}^{s-1} P_{t+u} P_t^{W-s} \right) \\ = & \sum_{s=0}^{W-1} \left(\prod_{u=0}^{s-1} P_{t+u} (P_{t+s} - P_t) P_t^{W-s-1} \right) \end{aligned} \qquad (11)$$

Next, we multiply both sides of Equation 11 by q_t and take the L_1-norm. Then, we apply Equation 10 and the facts that $\|q_t\|_1 = 1$ and $\|P_{t+s}\|_1 = 1$, for all $s = 0, \cdots, W-1$, to obtain the following:

$$\begin{aligned} & \left\| q_t \left(\prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) \right\|_1 \\ = & \left\| \sum_{s=0}^{W-1} q_t \left(\prod_{u=0}^{s-1} P_{t+u} \right) (P_{t+s} - P_t) P_t^{W-s-1} \right\|_1 \\ \le & \sum_{s=0}^{W-1} \left\| q_t \left(\prod_{u=0}^{s-1} P_{t+u} \right) (P_{t+s} - P_t) P_t^{W-s-1} \right\|_1 \\ \le & \sum_{s=0}^{W-1} \|q_t\|_1 \left(\prod_{u=0}^{s-1} \|P_{t+u}\|_1 \right) \|P_{t+s} - P_t\|_1 \|P_t\|_1^{W-s-1} \\ = & \sum_{s=0}^{W-1} \|P_{t+s} - P_t\|_1 \end{aligned} \qquad (12)$$

The first inequality in the above derivation follows from the triangle inequality. The second follows from the fact that the norm of a product is bounded above by the product of the norms. To understand the final quantity (Equation 12) intuitively, consider two coupled Markov chains, one of which uses P_t as its transition matrix, and the other of which uses P_{t+s}. These Markov chains lead to different distributions to the extent that they have different transition matrices.

Since $P_{t+s} = N_{t+s}/D_{t+s}$, it follows that:

$$\begin{aligned} & \|P_{t+s} - P_t\|_1 \\ = & \left\| \frac{N_{t+s}}{D_{t+s}} - \frac{N_t}{D_t} \right\|_1 \\ \le & \left\| \frac{N_{t+s}}{D_{t+s}} - \frac{N_{t+s}}{D_t} \right\|_1 + \frac{\|N_{t+s} - N_t\|_1}{D_t} \\ = & \|N_{t+s}\|_1 \frac{|D_{t+s} - D_t|}{(D_{t+s} D_t)} + \frac{\|N_{t+s} - N_t\|_1}{D_t} \end{aligned}$$

The inequality in this derivation follows from the triangle inequality.

Only $n-1$ of the $n(n-1)$ internal transformations affect any particular action and rewards are between 0 and 1, so $|D_{t+1} - D_t|$ is bounded by $n-1$. The induced L_1-norm of a matrix is the maximum row sum, after taking the absolute value of each entry; hence, $\|N_{t+1} - N_t\|_1$ is bounded by $n-1$. Further, $\|N_{t+s}\|_1 \le D_{t+s}$, $|D_{t+s} - D_t| \le s(n-1)$, and $\|N_{t+s} - N_t\|_1 \le s(n-1)$, so we conclude that $\|P_{t+s} - P_t\|_1 \le 2s(n-1)/D_t$. Summing over s from 0 to $W-1$ yields the desired $O(nW^2/D_t)$. ■

8 Experiments

We ran some simple experiments on the repeated Shapley game to see whether the theoretical bounds we derived match what is observed in practice. An instance of the internal regret-matching (IRM) algorithm[6] of Greenwald *et al.* [GLM06] was played against PI, HM with $\mu = 5$, and MT with $p = 10$. Our results are plotted in Figures 1, 2, 3 and 4 (the fourth figure summarizes all our results).

Each experiment was repeated 50 times, and each ensuing data series is plotted with two lines, delimiting the 95% confidence interval. The "true" line corresponding to infinitely many runs probably lies somewhere between the two plotted lines. Note the logarithmic scales of the axes, so powers such as $1/\sqrt{t}$ appear as straight lines.

What we observe is twofold: (i) PI does much better in practice than it does in theory, achieving better performance than HM and MT (see Figure 4); and (ii) MT does substantially worse than IRM, with the ratio similar to the $\sqrt{4(10)-3} \approx 6$ predicted by theory.

9 Discussion

Standard no-internal-regret (NIR) algorithms rely on a fixed point computation, and hence typically require $O(n^3)$ run time per round of learning. The main contribution of this paper is a novel NIR algorithm, which is a simple and straightforward variant of a standard NIR algorithm, namely that in Greenwald [GJMar]. Rather than compute a fixed point every round, our algorithm relies on power iteration to estimate a fixed point, and hence runs in $O(n^2)$ time per round.

One obvious question that comes to mind is: can power iteration be used in algorithms that minimize Φ-regret, for arbitrary Φ? The answer to this question is no, in general. For example, consider an ODP with two actions, and only one action transformation ϕ, which swaps the two actions: i.e.,

$$\phi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

A standard Φ-regret-minimizing algorithm would play the fixed-point of this matrix, which is uniform randomization. However, PI would learn a predictable sequence

[6] This algorithm is a close cousin of FV, and has the same regret bound.

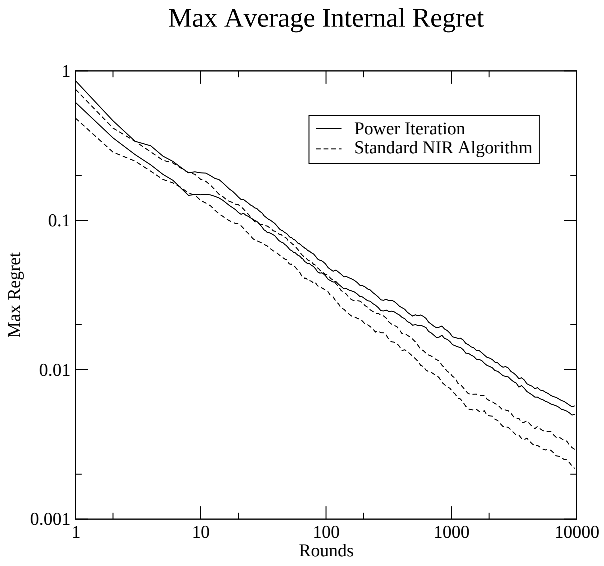

Figure 1: IRM and PI playing Shapley. 95% confidence interval of average of 50 runs shown.

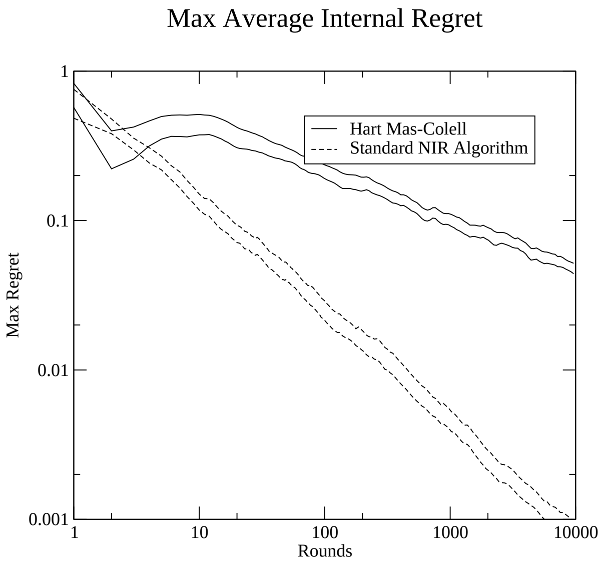

Figure 2: IRM and HM with $\mu = 5$ playing Shapley. 95% confidence interval of average of 50 runs shown.

of mixed actions, namely $q, 1-q, q, 1-q, \ldots$. Since an adversary could easily exploit this alternating sequence of plays, the idea does not immediately apply to arbitrary Φ. The part of the proof that is specific to Φ_{INT} is P_t having trace $n-1$, allowing us to use Lemma 10.

Another related question is: can power iteration be used in other NIR algorithms? For example, Cesa-Bianchi and Lugosi [CBL03] and Greenwald *et al.* [GLM06] present a class of NIR algorithms, each one of which is based on a potential function. Similarly, Blum and Mansour [BM05] present a method of constructing NIR learners from no-external-regret (NER) learners. We conjecture that the power iteration idea could be applied to any of these NIR algorithms, but we have not yet thoroughly explored this question.

Our admittedly limited experimental investigations reveal that perhaps PI's convergence rate in practice is not as bad as the theory predicts, but further study is certainly warranted. Another interesting question along the same lines is: would another iterative linear solving method, specifically one that is more sophisticated than power iteration, such as biconjugate gradient, yield better results, either in theory or in practice?

Acknowledgments

We are grateful to Dean Foster for originally suggesting that we try out power iteration in our experiments with regret-minimizing algorithms. We are also grateful to Casey Marks for providing much of the code for our experiments and to Yuval Peres for assistance simplifying the proof of Lemma 10. This research was supported in part by the Sloan Foundation.

References

[ACBFS02] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. *Siam J. of Computing*, 32(1):48–77, 2002.

[BM05] A. Blum and Y. Mansour. From external to internal regret. In *Proceedings of the 2005 Computational Learning Theory Conferences*, pages 621–636, June 2005.

[Cah04] A. Cahn. General procedures leading to correlated equilibria. *International Journal of Game Theory*, 33(1):21–40, December 2004.

[CBL03] N. Cesa-Bianchi and G. Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261, 2003.

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[CLRS01] Cormen, Leiserson, Rivest, and Stein. *Introduction to Algorithms*, chapter 28, pages 757–758. MIT Press, 2nd edition, 2001.

[CW87] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. In *STOC '87: Proceedings of the nineteenth annual ACM Symposium on Theory of Computing*, pages 1–6. ACM Press, New York, NY USA, 1987.

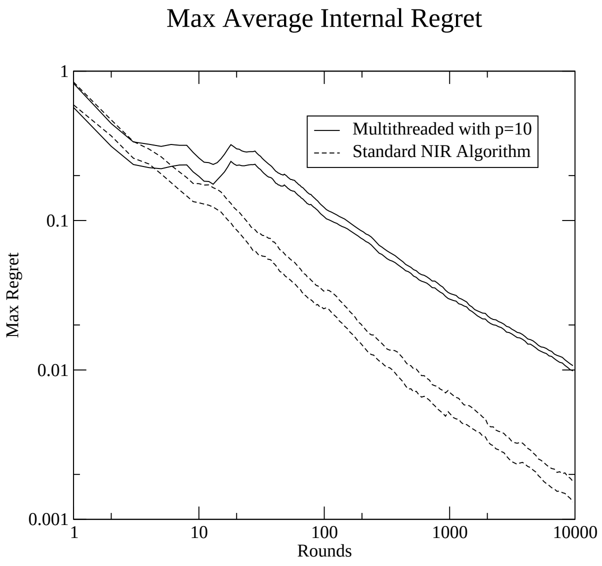

Figure 3: IRM and MT with $p = 10$ playing Shapley. 95% confidence interval of average of 50 runs shown.

[FS97] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[FV97] D. Foster and R. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21:40–55, 1997.

[GJMar] A. Greenwald, A. Jafari, and C. Marks. A general class of no-regret algorithms and game-theoretic equilibria. In Amitabha Gupta, Johan van Benthem, and Eric Pacuit, editors, *Logic at the Crossroads: An Interdisciplinary View*, volume 2. Allied Publishers, To Appear.

[GLM06] A. Greenwald, Z. Li, and C. Marks. Bounds for regret-matching algorithms. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006.

[HMC00] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.

[HMC01] S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26–54, 2001.

[Lin92] Torgny Lindvall. *Lectures on the Coupling Method*, chapter II.12, pages 41–47. Wiley, 1992.

[LW94] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212 – 261, 1994.

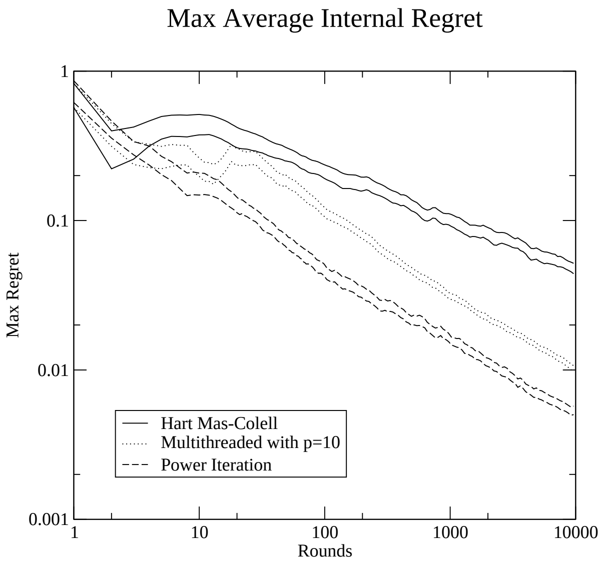

Figure 4: Summary of Figures 1, 2 and 3.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.

[You04] P. Young. *Strategic Learning and its Limits.* Oxford University Press, Oxford, 2004.

A Proof of technical Lemma 10

Note: in this proof, we use N, M and t for meanings unrelated to those in the main body of the paper. Don't be confused.

Let M be an n by n matrix which is row-stochastic (vectors should be multiplied on the left as in qM) and has trace at least $n - 1$. We want to show:

$$\max_{\|q\|_1=1} \left\| q(M^W - M^{W-1}) \right\|_1 = O(1/\sqrt{W}).$$

For any two probability measures μ and ν on probability space Ω, their total variation distance is defined to be

$$||\mu - \nu||_{TV} = \frac{1}{2} \sum_{a \in \Omega} |\mu(a) - \nu(a)|. \qquad (13)$$

We denote by $\mu(X)$ the distribution of random variable X. Let $Q(t)$ denote the state of a Markov chain with transition matrix M and initial distribution q after t time steps. Our desired conclusion can be recast in Markov chain language as

$$||\mu(Q(W)) - \mu(Q(W-1))||_{TV} = O(1/\sqrt{W})$$

for all initial distributions q.

We know that $\sum_i m_{i,i} \geq n - 1$ where $m_{i,j}$ is the element of the ith row and jth column in matrix M. All $m_{i,i}$ are at most 1, so there can be at most one state, call it p, satisfying $m_{p,p} < 1/2$. If no such state exists the Lemma was already shown as step M7 of [HMC00], so assume it exists.[7] We define a new matrix N as the unique solution to $m_{i,j} = \frac{1}{2}(n_{i,j} + \delta_{i,j})$ if $i \neq p$, and $m_{p.j} = n_{p,j}$ otherwise, where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.[8] It is easy to check N is row stochastic and is therefore the transition matrix of some Markov Chain. For simplicity we denote by N the Markov Chain with transition matrix N and initial distribution q. Let $B(0), B(1), ...$ be the random walk associated with Markov Chain N.

One can easily create a random walk of the Markov chain M from the walk of N as follows. If the current state is the special state p, do what the N-chain did. Otherwise, flip a coin, following N with probability 50% and remaining at the same state otherwise. Formally, define index I_t inductively by $I_0 = 0$ and $I_t = I_{t-1}+$a symmetric Bernoulli distribution, if $B(I_{t-1}) \neq p$, and $I_t = I_{t-1} + 1$ otherwise. Then it can be shown that $Q(t) = B(I_t)$ has distribution qM^t. We need to show the distributions of $B(I_t)$ and $B(I_{t-1})$ are close to each other.

For $i \geq 0$, define $X_i = \#\{\, t \geq 0 \;:\; I_t = i \,\}$. Intuitively, X_i is the number of steps the M chain takes while the N chain is in state $B(i)$. Our proof hinges on the easily seen fact that the X_is are *independent* random variables. If $B(i) = p$, $X_i = 1$ and if $B(i) \neq p$, X_i is geometrically distributed with mean 2.[9] Let $T_i = \sum_{j=0}^{i-1} X_i$. One can see that $I_W = \max\{\, i \;:\; T_i \leq W \,\}$.

In order to prove the conclusion we need the following two lemmas.

Lemma 12 *Let $f(y,z)$ be a function, Y, Y' and Z be random variables with Z pairwise independent of Y and Y'. Let $I = f(Y,Z)$ and $I' = f(Y',Z)$ be random variables. Then*

$$||\mu(I) - \mu(I')||_{TV} \leq ||\mu(Y) - \mu(Y')||. \qquad (14)$$

Proof: The total variation distance between $\mu(X)$ and $\mu(X')$ is equal to minimum possible probability that $X \neq X'$ over couplings of X and X'. A coupling of Y and Y' can be extended into a coupling of I and I' trivially. ■

Lemma 13 *Let $S_k = \sum_{i=1}^{k} X_i$ where X_i's are independent identically distributed random variables with geometric distributions with mean 2, then $||\mu(S_k) - \mu(1 + S_k)||_{TV} = O(n^{-1/2})$.*

Proof: Equation II.12.4 of [Lin92]. ■

The probability that $I_W < W/3$ is bounded by the probability that the sum of W Bernoulli random variables with mean $1/2$ is less than $W/3$. A standard Chernoff bound therefore shows that the event E that $I_W < W/3$ is exponentially unlikely. Condition on the path $B(0), B(1), \ldots$ and event E not happening. We can therefore write $I_W = \max\{\, i \;:\; \sum_{j=W/3+1}^{i} X_i \leq W - T_{W/3} \,\}$ and $I_{W-1} = \max\{\, i \;:\; \sum_{j=W/3+1}^{i} X_i \geq W - 1 - T_{W/3} \,\}$. Now by Lemma 12, associating Z with the vector-valued random variable $\{X_i\}_{i=W/3+1}^{\infty}$, Y with $W - T_{W/3}$ and Y' with $W - 1 - T_{W/3}$, we see that it suffices to bound the total variation distance between $T_{W/3}$ and $T_{W/3} + 1$.

Define $k = \#\{\, 0 \leq i \leq W/3 \;:\; B(i) \neq b \,\}$. Every visit to b is followed by a visit to another state with probability at least $1/2$, so with exponentially high probability over the choices of the $B(i)$, $k \geq W/12$. Condition on the event $k \geq W/12$. By definition $T_{W/3} = (W/3 - k) + \sum_{i:B(i)\neq b} X_i$, so it suffices to analyze the variation distance between the sum of $k \geq W/12$ geometric random variables and the same shifted by 1. By Lemma 13, this is $O(1/\sqrt{W})$.

Therefore we obtain

$$||\mu(I_W) - \mu(I_{W-1})||_{TV} = O(W^{-1/2}) \qquad (15)$$

Back to M chain $B(I_W)$ we could have

$$\begin{aligned} Pr(B(I_W) = i) &= \sum_a Pr(B(I_W) = i | I_W = a) Pr(I_W = a) \\ &= \sum_a Pr(B(a) = i) Pr(I_W = a) \end{aligned}$$

and similarly

$$Pr(B(I_{W-1}) = i) = \sum_a Pr(B(a) = i) Pr(I_{W-1} = a)$$

Thus

$$\begin{aligned} &||\mu(B(I_W) - \mu(B(I_{W-1})||_{TV} \\ &= \frac{1}{2} \sum_i |Pr(B(I_W) = i) - Pr(B(I_{W-1}) = i)| \\ &\leq \frac{1}{2} \sum_i \sum_a Pr(B(a) = i) |Pr(I_W = a) - Pr(I_{W-1} = a)| \\ &= \frac{1}{2} \sum_a |Pr(I_W = a) - Pr(I_{W-1} = a)| \\ &= ||\mu(I_W) - \mu(I_{W-1})||_{TV} \\ &= O(W^{-1/2}) \end{aligned}$$

This concludes the proof.

We remark that this lemma can also be proved in a more self-contained manner via a Markov chain coupling. The motivating story follows.

Charlie and Eve are walking drunkenly between the n neighborhood bars. If Charlie is in a good bar, each time step he first flips a coin to decide whether or not he should leave that bar. If he decides to leave, he then makes a probabilistic transition to some bar (perhaps the same one). If Charlie is in a bad bar, he always leaves. Eve starts one time step later than Charlie at the same initial bar. Eve makes her decision to leave or not

[7] One can see that the current proof also holds if no such state exists.

[8] I.e., $n_{i,j} = 2m_{i,j} - \delta_{i,j}$ if $i \neq p$, and $n_{p.j} = m_{p,j}$ otherwise.

[9] Note that our "geometrically distributed" variables have support $1, 2, \ldots$, so $\Pr(X_i = k) = (1/2)^k$ for $k \geq 1$.

independently of Charlie, but reuses Charlie's choices of where to go next. However, if Eve ever catches up with Charlie, she switches to just following him around. A natural question to ask is how likely Eve and Charlie are to be at the same bar after t time steps? Note that if you look at Eve's motions and ignore Charlie's, she behaves exactly like Charlie does.

The connection to the present lemma is that Charlie's distribution corresponds to M^t and Eve's to M^{t-1}. Standard arguments relating total variation distance to couplings show that if Eve and Charlie usually finish at the same bar, their probability distributions must be quite similar.