

On algorithmically boosting fixed-point computations

Ioannis Avramopoulos*

Nikolaos Vasiloglou*

This paper is a thought experiment on *exponentiating algorithms*. One of the main contributions of this paper is to show that this idea finds material implementation in exponentiating *fixed-point computation algorithms*. Various problems in computer science can be cast as instances of computing a fixed point of a map. In this paper, we present a general method of boosting the convergence of iterative fixed-point computations that we call *algorithmic boosting*, which is a (slight) generalization of algorithmic exponentiation. We first define our method in the general setting of nonlinear maps. Secondly, we restrict attention to convergent linear maps and show that our algorithmic boosting method can set in motion *exponential speedups in the convergence rate*. Thirdly, we show that algorithmic boosting can convert a (weak) non-convergent iterator to a (strong) convergent one. We then consider a *variational approach* to algorithmic boosting providing tools to convert a non-convergent continuous flow to a convergent one. We, finally, discuss implementations of the exponential function, an important issue even for the scalar case.

1 Introduction

The main idea explored in this paper is that *exponentiation is a form of averaging* by which we treat *exponentiation* (for example, raising the elements of a vector to an exponent) and *averaging* (for example, averaging the orbit of a dynamical system) under the same analytical footing. More fundamentally, this paper is an inquiry into the computational foundations of fixed point theory. Various fixed point theorems (such as the Brouwer fixed-point theorem and the Knaster-Tarski theorem) are non-constructive and our ultimate goal is to develop algorithms by which this gap can be bridged. We herein only look at constructive fixed point theorems such as the Perron-Frobenius theorem and the minimax theorem, which are, in fact, related by a theorem of Blackwell [1961].

Our perspective is that of a powerful algorithmic abstraction whose development was driven by a curiosity question, namely, what, if any, role *exponentiation* can play in such a computational theory of fixed points. Exponentiation plays a key role in computational learning theory [Littlestone and Warmuth, 1994, Freund and Schapire, 1996, Schapire and Freund, 2012]. Learning theory is grounded on a different analytical footing than fixed point theory, nevertheless, the approaches are related. It is typically a small step to convert an (online) learning algorithm to a *fixed-point iterator* (for example, simply assuming the adversary is nature will often do the trick) and that was certainly an important inspiration for the theory we develop in this paper.

1.1 OUR MAIN IDEA IN SIMPLE TERMS

Our curiosity had many fruits to bear: Our thesis is that *applying exponentiation to algorithmic problems is a powerful pursuit* and our line of discourse can be appositely framed by asking the question: **How can we exponentiate entire algorithms?** Our idea to answer this question is the observation that, if $f(\cdot)$ is a self-map on the set \mathbb{R} of real numbers, since, by an elementary

*RelationalAI, Inc.

property of the exponential function over the reals, we have that

$$\exp(f(x)) = \sum_{k=0}^{\infty} \frac{1}{k!} f^k(x),$$

we can meaningfully think of exponentiation as an averaging process (in signal processing terms, as a *filter*). In fact, normalizing the previous expression by dividing with $\exp(1)$, we obtain an exponential operator as a convex combination of the powers of f that, as we will prove shortly, inherits the fixed points of f . That is, if x^* is a fixed point f , in that $f(x^*) = x^*$, then

$$\frac{1}{\exp(1)} \exp(f(x^*)) = x^*.$$

To apply this (averaging) idea to algorithms, we need to think of a function as an algorithm and the analogy is immediately apparent if we specifically look at algorithms that compute fixed points: Various problems in computer science can be cast as instances of computing a fixed point of a map. Given a set of vectors, say X , and a self-map $f(\cdot)$ on X , a *fixed point* of f is an $x^* \in X$ such that $f(x^*) = x^*$. In this paper, we focus on one particular method of computing a fixed point of a map, namely, the iterative application of either the map itself or some map that is naturally related to it. We further focus on two particular problem domains, namely, *linear algebra* and *game theory*.

1.2 EXPONENTIATION IN LINEAR ALGEBRA

If X is Euclidean space and f is a linear map, then the action of f on X can be represented by a square matrix, say A . That is, $f(x) = Ax$. In this case, a fixed point of f is a vector $x^* \in X$ such that $Ax^* = x^*$. That is, x^* is an eigenvector of A corresponding to eigenvalue 1, if it exists. A prominent example of an algorithm that can be cast as a problem of computing an eigenvector corresponding to the eigenvalue 1 is PageRank [Brin and Page, 1998]. In fact, our algorithmic boosting theory began as an effort to apply exponentiation to the PageRank algorithm.

1.2.1 Trying to exponentiate PageRank

One approach to apply exponentiation to the PageRank algorithm is in the paradigm of the *multiplicative weights update method* [Arora et al., 2012], which is a very successful paradigm of designing algorithms that manifests in various disciplines of theoretical computer science. For example, in theoretical machine learning, exponentiated multiplicative updates manifest in *boosting*, which refers to a method of producing an accurate prediction rule by combining inaccurate prediction rules [Schapire and Freund, 2012]. A well-established boosting algorithm is AdaBoost [Freund and Schapire, 1997]. Related to AdaBoost is the Hedge algorithm for playing a mathematical game [Freund and Schapire, 1999]. At the heart of AdaBoost and Hedge lies the weighted majority algorithm [Littlestone and Warmuth, 1994] (see also [Freund and Schapire, 1996]), which is also based on exponentiation. It was natural to ponder whether PageRank bears the structure of one of these problems. In fact, if we could spot a gradient we could just exponentiate that. The answer we give in this paper is unlikely to be unique but it motivated basic results in algorithmic exponentiation.

1.2.2 Exponentiating PageRank by exponentiating the Google matrix

The critical point in the development of the theory we lay out in this paper was in realizing that PageRank bears the structure of a linear fixed point problem as it was then a no-brainer to exponentiate the matrix operator itself. An advantage of this approach is that our method applies with

minor or no modifications to other computational link-analysis problems such that *spectral rankings* [Vigna, 2016] and *community detection* [Newman, 2006]. In fact, the *exponentiated power method* we propose and analyze in this paper is a contribution in numerical linear algebra [Golub and van Loan, 1996]. Our method consists in applying power iterations using the *matrix exponential* which is defined by the power series

$$\exp(A) = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$$

directly generalizing the previous definition of the exponential function. Our exponentiated power method boosts the convergence rate of the simple power method by an exponential factor.

1.3 A GENERALIZATION OF THE EXPONENTIAL MATRIX

Once such a powerful idea has been set into place it is natural to try to apply it to fixed-point problems that we don't have efficient algorithms for, for example, Nash equilibrium problems in game theory. In this vein, we generalize the matrix exponential as follows:

1.3.1 Fully exponentially powered boosting

Definition 1. Let X be a convex set of vectors and $f(\cdot)$ a self-map on X . Given $\alpha > 0$, we define the fully exponentially powered self-map $\exp(\alpha f(\cdot))$ as

$$\exp(\alpha f(x)) = \frac{1}{\exp(\alpha)} \sum_{k=0}^{\infty} \frac{1}{k!} \alpha^k f^k(x)$$

where

$$\exp(\alpha) = \sum_{k=0}^{\infty} \frac{1}{k!} \alpha^k$$

and

$$f^0(x) = x \quad f^1(x) = f(x) \quad f^2(x) = f(f(x)) \quad \dots$$

We call α the learning rate of the fully exponentially powered map.

Lemma 1. Let X be a convex set of vectors and $f(\cdot)$ a self-map on X . Then, if x^* is a fixed point of f , that is, if $f(x^*) = x^*$, it is also a fixed point of $\exp(\alpha f(\cdot))$.

Proof. Since x^* is a fixed point of f , we obtain that

$$\exp(\alpha f(x^*)) = \frac{1}{\exp(\alpha)} \sum_{k=0}^{\infty} \frac{1}{k!} \alpha^k f^k(x^*) = \frac{1}{\exp(\alpha)} \sum_{k=0}^{\infty} \frac{1}{k!} \alpha^k x^* = \frac{1}{\exp(\alpha)} \left(\sum_{k=0}^{\infty} \frac{1}{k!} \alpha^k \right) x^* = x^*,$$

where in the last equality we applied the definition of $\exp(\alpha)$. This completes the proof. \square

Lemma 2. Let X be Euclidean space and $f(\cdot)$ a linear self-map on X . Then,

$$\exp(\alpha f(x)) = \frac{1}{\exp(\alpha)} \exp(\alpha A)x.$$

Proof. Simply following the definitions, we obtain that

$$\exp(\alpha f(x)) = \frac{1}{\exp(\alpha)} \sum_{k=0}^{\infty} \frac{1}{k!} \alpha^k f^k(x) = \frac{1}{\exp(\alpha)} \sum_{k=0}^{\infty} \frac{1}{k!} \alpha^k A^k x = \frac{1}{\exp(\alpha)} \exp(\alpha A) x$$

and this completes the proof. \square

Unsure how to call this generalization of exponentiation, we decided to call it *algorithmic boosting*.

1.3.2 Variants of algorithmic boosting

One may ponder if the exponential power series is essential in the definition of algorithmic boosting, and a moment of thought immediately suggests that it is not. We may generalize exponentially powered boosting as follows: Let X be a convex set of vectors. Given a self-map $f(\cdot)$ on X , we define the powered correspondence $\mathbb{P}(f(\cdot))$ as a self-correspondence on X such that $\mathbb{P}(f(x))$ is the ω -limit set of the sequence

$$\left\{ \frac{1}{W_K} \sum_{k=0}^K w_k f^k(x) \right\}_{K=0}^{\infty},$$

where $\{w_k\}_0^{\infty}$ is a sequence of nonnegative scalars and $W_K = \sum_{k=0}^K w_k$. Observe that since X is convex, $\mathbb{P}(f(\cdot))$ is well-defined at every $x \in X$. This definition can be specialized based on the pattern of the sequence $\{w_k\}_0^{\infty}$ of weights used in the averaging process: For example, if $w_k = 1, k = 0, 1, \dots$, we obtain what we may call *geometrically powered boosting* (cf. geometric series), whereas if $w_0 = 1$ and $w_k = 1/k, k = 1, 2, \dots$, we obtain what we may call *harmonically powered boosting* (cf. harmonic series). If there exists a natural number m such that $w_k = 0, k > m$, we obtain truncated series. We will see that these definitions also admit *variational interpretations*.

Starting here, with the exception of Section 2, we use upper-case letters to denote vectors.

1.4 ALGORITHMIC BOOSTING IN GAME THEORY

Algorithmic boosting theory in games is a generalization of a fundamental result in computational learning theory, namely, that the empirical average of iterated Hedge using a fixed learning rate converges to an approximate Nash equilibrium in a zero-sum game [Freund and Schapire, 1999].

1.4.1 Background in game theory

The fundamental solution concept in game theory is the Nash equilibrium, which is a fixed point of the best response correspondence. Nash's proof of the existence of an equilibrium in an N -person game is non-constructive. In this paper, we mainly focus on zero-sum games that readily admit a constructive Nash-equilibrium existence proof (but we also discuss how our results can be extrapolated to the general setting). A fixed-point iterator that converges to a Nash equilibrium insofar eludes us. In this paper, we focus on symmetric zero-sum games. The aforementioned result of Freund and Schapire [1999] is a constructive proof of the existence of a Nash equilibrium in a symmetric zero-sum game. Before continuing let us try to fix terminology.

Given a symmetric bimatrix game (C, C^T) , we denote the corresponding standard (probability) simplex by $\mathbb{X}(C)$. $\overset{\circ}{\mathbb{X}}(C)$ denotes the relative interior of $\mathbb{X}(C)$. The elements (probability vectors) of $\mathbb{X}(C)$ are called *strategies*. We call the standard basis vectors in \mathbb{R}^n *pure strategies* and denote

them by $E_i, i = 1, \dots, n$. *Symmetric Nash equilibria* are precisely those combinations of strategies (X^*, X^*) such that X^* satisfies $\forall Y \in \mathbb{X}(C) : X^* \cdot CX^* - Y \cdot CX^* \geq 0$. We call X^* a *symmetric Nash equilibrium strategy*. A symmetric Nash equilibrium is guaranteed to always exist [Nash, 1951]. A simple fact is that $X^* \in \mathbb{X}(C)$ is a symmetric Nash equilibrium strategy if and only if $(CX^*)_{\max} - X^* \cdot CX^* = 0$. If $(CX^*)_{\max} - X^* \cdot CX^* \leq \epsilon$, X^* is called an ϵ -approximate equilibrium strategy. A symmetric bimatrix game (C, C^T) is zero-sum if C is antisymmetric, that is, $C = -C^T$.

1.4.2 Computing a Nash equilibrium by averaging the powers of a map

A plausible approach to compute a Nash equilibrium in this setting is to use Hedge. Given C , Hedge is given by map $T : \mathbb{X}(C) \rightarrow \mathbb{X}(C)$ where

$$T_i(X) = X(i) \frac{\exp\{\alpha(CX)_i\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}, i = 1, \dots, n, \quad (1)$$

and α is a parameter called the *learning rate*. To compute a symmetric Nash equilibrium of (C, C^T) we can, for example, iterate Hedge using a fixed learning rate starting from an interior strategy $X^0 \in \mathring{\mathbb{X}}(C)$. However, a simple fact is that the sequence $\{T^k(X^0)\}_{k=0}^\infty$ may not converge.

Algorithmic boosting theory factors at this critical moment to obtain convergence. Although the sequence $\{T^k(X^0)\}_{k=0}^\infty$ may not converge, translating the aforementioned result of Freund and Schapire [1999] in the language we are trying to develop in this paper, *geometrically powered boosting of iterated Hedge using a fixed learning rate $\alpha > 0$ converges to an ϵ -approximate Nash equilibrium, where ϵ can be made as small as desired by choosing an accordingly small value of the learning rate α* . What's more, Avramopoulos [2023] shows in a recent contribution to this fundamental question that, under a diminishing learning rate schedule, what we may now call harmonically powered boosting of iterated Hedge converges to an exact Nash equilibrium of a symmetric zero-sum game. In this paper, we develop a set of algorithmic techniques that draw on *convex optimization* to devise an equilibrium fully polynomial time approximation scheme in a symmetric zero-sum game by averaging iterated Hedge. In this vein, we show that starting from the uniform strategy, to compute an ϵ -approximate Nash equilibrium, our scheme requires at most

$$\left\lceil \frac{\ln(n)}{\frac{\epsilon}{2} \ln\left(1 + \frac{\epsilon}{2}\right)} \right\rceil$$

iterations. Our bound nearly exactly matches that of Freund and Schapire [1999] but, since

$$\ln\left(1 + \frac{\epsilon}{2}\right) > \frac{\frac{\epsilon}{2}}{1 + \frac{\epsilon}{2}}, \epsilon > 0,$$

it is better. We also leverage our techniques to obtain matching bounds under a significantly more numerically stable version of Hedge that is obtained by a *double exponentiation* in (1). We discuss in detail our approach below. Before that let's get to our result on the continuous limit of Hedge.

1.4.3 Computing a Nash equilibrium by averaging an orbit of a flow

Taking the long-run average of an orbit of a differential equation is the *variational analogue of algorithmic boosting*. In this paper, we also consider a variational analog of Hedge, namely, the *replicator dynamic* [Taylor and Jonker, 1978]. We first show that Hedge is a convergent and consistent numerical integrator of the replicator dynamic. We then prove that, in a symmetric zero-sum game, the ω -limit set of the average of every orbit of this dynamic starting in the interior of the

simplex is a symmetric Nash equilibrium strategy. Our result informs a line of research regarding the divergent behavior of the replicator dynamic in zero-sum games [Mertikopoulos et al., 2018, Biggar and Shames, 2023] in an elegant fashion: Although orbits may diverge (for example, they can cycle [Akin and Losert, 1984]), the long-run average converges. The theory we lay out in this paper further informs a line of research regarding the divergent behavior of evolutionary dynamics in general [Flokas et al., 2020, Milionis et al., 2022]: Although the orbits themselves may fail to converge, we stipulate that, in the fashion of our result on the replicator dynamic but also in the fashion of [Freund and Schapire, 1999], a variety of carefully constructed averages may converge.

1.5 IMPLEMENTING ALGORITHMIC BOOSTING

We have reached one of the most exciting parts of this research, namely, how to implement boosting. Our contributions in this vein are in the implementation of the exponentiation in Hedge.

1.5.1 Using Padé approximation theory to compute the matrix exponential

One approach to approximately compute the matrix exponential is to truncate its Taylor series. In fact, this is just one out a big list of methods [Moler and van Loan, 1978]. A method that is often used in practice is to compute the Padé approximant. In general, Padé approximations approximate a function by a rational function of a given order. A $[p/q]$ Padé approximant is a ratio of a polynomial of degree p over a polynomial of degree q . For example, the $[3/3]$ Padé approximant of $\exp(x)$, denoted by $\exp_{3/3}(x)$, is

$$\exp_{3/3}(x) = \frac{120 + 60x + 12x^2 + x^3}{120 - 60x + 12x^2 - x^3}.$$

Padé approximations of the matrix exponential follow the same principle [Arioli et al., 1996, Higham, 2005]. Since Padé approximation theory generalizes to non-convergent series, we believe there is fertile ground to apply this theory to implement general algorithmic boosting methods.

1.5.2 Using relative entropy programming to implement Hedge

Once a general framework for the implementation of the exponential function has been set into place, it becomes immediately apparent that the exponentiation operation used in the Hedge map may entail complexities that require careful attention. Krichene et al. [2015] show that the Hedge map is a dual formulation of a convex optimization problem (with the same solution). The exact solution of this problem can be computed to any desirable precision in polynomial time but this implementation requires solving a *relative entropy program* [Chandrasekaran and Shah, 2017] using an *interior-point convex-programming method* [Nesterov and Nemirovski, 1994]. In a numerical experiment we show that there is a large discrepancy between the simple algebraic implementation of Hedge as that is suggested in equation (1) and the robust method that uses convex programming.

1.5.3 Using double exponentiation to obtain a more stable Hedge

It is natural then to ask, for example, to what extent the bounds of [Freund and Schapire, 1999] render practical algorithms to approximately solve zero-sum games (and, therefore, also linear programs) using inexpensive implementations of the exponential function. Our intuition from the aforementioned numerical experiment suggests that the vanilla implementation of the exponential function may not always suffice to obtain the desired performance. Our numerical experiments in Matlab show that there is a deviation in the evolution of the Hedge map under inexpensive

exponentiation and under the robust relative entropy programming implementation *even in the rock-paper-scissors game using a small learning rate*. In this paper, we show how to restore numerical stability, using inexpensive exponentiation in the same numerical environment, first by doubly exponentiating Hedge as in the map

$$T_i(X) = X(i) \frac{\exp\{\alpha \exp\{\beta((CX)_i - (CX)_{\max})\}\}}{\sum_{j=1}^n X(j) \exp\{\alpha \exp\{\beta((CX)_j - (CX)_{\max})\}\}}, i = 1, \dots, n,$$

and then taking the limit as $\beta \rightarrow \infty$, which gives the map

$$T_i(X) = X(i) \frac{\exp\{\alpha Q(i)\}}{\sum_{j=1}^n X(j) \exp\{\alpha Q(j)\}} \quad Q(i) = \begin{cases} 1, i \in \arg \max\{(CX)_j\} \\ 0, i \notin \arg \max\{(CX)_j\} \end{cases} \quad i = 1, \dots, n, .$$

This map is straightforward to implement, it is conceptually simpler than Hedge, it has the same theoretical performance as Hedge, and, as our numerical experiments show, it is numerically stable.

1.6 OTHER RELATED WORK

Closely related to our algorithmic boosting theory is *ergodic theory* that systematically studies *conservative systems* from a similar perspective. The problems we study in this paper are not, in general, conservative. In such a sense, algorithmic boosting theory is a generalization of ergodic theory. We believe that studying ergodic systems from an algorithmic boosting perspective, for example, exponentiating the Hamiltonian operator, is an interesting direction for future work.

The precise role of exponentiation in the multiplicative weights update method is not well-understood. An aspect is explored by Pelillo and Torsello [2006] who report an empirical finding that using exponentiation (in the same fashion as that is used in the multiplicative weights update method) in quadratic optimization significantly increases the convergence rate over more elementary algorithms that obviate exponentiation: For example, Hedge is faster than the *replicator dynamic* or the *discrete-time replicator dynamic*. Our independently performed experiments confirm this.

Our variational perspective on algorithmic boosting is in the paradigm of Wibisono et al. [2016] who initiated the study of acceleration methods in optimization from a variational perspective. In fact, that Hedge is a discretization of the replicator dynamic was an idea drawn from that paradigm.

Although our paper is not the first to propose using the matrix exponential in link analysis [Miller et al., 2001], to the extent of our knowledge, our paper is the first that observes that using the matrix exponential accelerates the classical power method and analyzes the precise impact on the convergence rate, in particular, that the convergence rate increases by an exponential factor.

1.7 OVERVIEW OF THE REST OF THIS PAPER

In Section 2 we present and analyze our exponentiated power method. In Sections 3 and 4 we analyze algorithmic boosting in game theory (both in discrete and continuous time). Sections 5 and 6 further discuss implementations of the Hedge map and Section 7 concludes the paper.

2 Algorithmically boosting convergent linear fixed-point iterators

In this section, we consider the power iteration method for computing a dominant eigenvector of an $n \times n$ matrix A , which starts with a random vector x_0 and recursively applies the linear map corresponding to A to x_0 . This process converges to the dominant eigenvector. In this section, we analyze the following idea: Instead of using A itself in the power iterations, use $\exp(\alpha A)$,

where $\alpha > 0$ and $\exp(\cdot)$ is the matrix exponential. We show that our proposed method gives an exponential increase in the convergence rate. Let us start by defining the power iteration method.

2.1 THE POWER ITERATION METHOD

Given an $n \times n$ matrix A assume that its eigenvalues are ordered such that $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$. The power iteration method (for example, see [Golub and van Loan, 1996, Chapter 7.3]) computes the *dominant* (i.e., largest in modulus) eigenvalue λ_1 and corresponding (dominant) eigenvector. To that end, it starts with a $n \times 1$ vector x_0 and iteratively generates a sequence $\{x_k\}$ using the recurrence relation

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}, k = 0, 1, 2, \dots$$

The sequence $\{x_k\}$ converges to the dominant eigenvector provided that x_0 is not orthogonal to the left eigenvector corresponding to λ_1 . Furthermore, under these assumptions, the sequence

$$\left\{ \frac{x_k^T A b_k}{x_k^T x_k} \right\}_{k=0}^{\infty}$$

converges to λ_1 . A typical case in the practical application of the method is that λ_1 is equal to one.

2.2 ANALYSIS WHEN A IS DIAGONALIZABLE

Let us first consider (as a warmup) the power iteration methods assuming A is *diagonalizable*. We consider the simple power iteration method first followed by the exponentiated power method.

2.2.1 The simple power iteration method

So let us assume A is diagonalizable, let $\lambda_1, \lambda_2, \dots, \lambda_n$ be the n eigenvalues of A (counted with multiplicity) and let u_1, u_2, \dots, u_n be the corresponding eigenvectors. Suppose that λ_1 is the dominant eigenvalue, so that $|\lambda_1| > |\lambda_j|$ for $j > 1$. We then have

$$x_0 = c_1 u_1 + c_2 u_2 + \dots + c_n u_n$$

and

$$\begin{aligned} A^k x_0 &= c_1 A^k u_1 + c_2 A^k u_2 + \dots + c_n A^k u_n \\ &= c_1 \lambda_1^k u_1 + c_2 \lambda_2^k u_2 + \dots + c_n \lambda_n^k u_n \\ &= c_1 \lambda_1^k \left(u_1 + \frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1} \right)^k u_2 + \dots + \frac{c_n}{c_1} \left(\frac{\lambda_n}{\lambda_1} \right)^k u_n \right). \end{aligned}$$

Observe now that since

$$\left| \frac{\lambda_j}{\lambda_1} \right| < 1, j > 1$$

we obtain that

$$A^k x_0 \rightarrow c_1 \lambda_1^k u_1$$

and since

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}, k = 0, 1, 2, \dots$$

we obtain that the sequence $\{x_k\}$ converges to (a multiple of) the eigenvector u_1 . The convergence is *geometric* with ratio

$$\left| \frac{\lambda_2}{\lambda_1} \right|,$$

where λ_2 is the second dominant eigenvalue.

2.2.2 The fully exponentially powered iteration method

Like the power method, our *exponentiated power method* computes the dominant eigenvalue and corresponding eigenvector of an $n \times n$ matrix A under the same conditions as the aforementioned power method, using instead the *matrix exponential* of αA , which we denote by $\exp(\alpha A)$, where

$$\exp(A) = I + A + \frac{1}{2}A^2 + \dots,$$

and $\alpha > 0$ is a parameter we call the *learning rate*. The exponentiated power iteration starts with a $n \times 1$ vector x_0 and iteratively generates a sequence $\{x_k\}$ using the recurrence relation

$$x_{k+1} = \frac{\exp(\alpha A)x_k}{\|\exp(\alpha A)x_k\|}, k = 0, 1, 2, \dots$$

Let us prove that the sequence $\{x_k\}$ converges to the dominant eigenvector under the same conditions that the power method converges. To that end, let $\lambda_1, \lambda_2, \dots, \lambda_n$ be the n eigenvalues of A (counted with multiplicity) and let u_1, u_2, \dots, u_n be the corresponding eigenvectors. Suppose that λ_1 is the dominant eigenvalue, so that $|\lambda_1| > |\lambda_j|$ for $j > 1$. We then have

$$x_0 = c_1 u_1 + c_2 u_2 + \dots + c_n u_n$$

and

$$\begin{aligned} \exp(\alpha A)^k x_0 &= c_1 \exp(\alpha A)^k u_1 + c_2 \exp(\alpha A)^k u_2 + \dots + c_n \exp(\alpha A)^k u_n \\ &= c_1 \exp(\alpha \lambda_1)^k u_1 + c_2 \exp(\alpha \lambda_2)^k u_2 + \dots + c_n \exp(\alpha \lambda_n)^k u_n \\ &= c_1 \exp(\alpha \lambda_1)^k \left(u_1 + \frac{c_2}{c_1} (\exp(\alpha(\lambda_2 - \lambda_1)))^k u_2 + \dots + \frac{c_n}{c_1} (\exp(\alpha(\lambda_n - \lambda_1)))^k u_n \right), \end{aligned}$$

where we have used the simple property that if u is an eigenvector of A corresponding to eigenvalue λ , then it is also an eigenvector of $\exp(A)$ with corresponding eigenvalue $\exp(\lambda)$. Observe now that, assuming $\lambda_1 > 0$ since

$$\frac{\exp(\alpha \lambda_j)}{\exp(\alpha \lambda_1)} < 1, j > 1$$

we obtain that

$$\exp(\alpha A)^k x_0 \rightarrow c_1 \exp(\alpha \lambda_1)^k u_1$$

and since

$$x_{k+1} = \frac{\exp(\alpha A)x_k}{\|\exp(\alpha A)x_k\|}, k = 0, 1, 2, \dots$$

we obtain that the sequence x_k converges to (a multiple of) the eigenvector v_1 . The convergence is *geometric* with ratio

$$\exp(\alpha(|\lambda_2| - \lambda_1)),$$

where λ_2 is the second dominant eigenvalue.

2.3 ANALYSIS IN THE GENERAL CASE

In the general case, the fully exponentiated power method gives an exponential speedup in the rate by which the sinusoid of the angle between the iterates and the dominant eigenvector goes to zero. Toward proving this phenomenon, in the rest of this section, we follow [Arbenz, 2016] where it is shown in Chapter 7 that in the simple power method the sinusoid of this angle (between the iterates and the dominant eigenvector) converges to zero geometrically with a rate equal to

$$\left| \frac{\lambda_2}{\lambda_1} \right|.$$

In particular, it is shown that:

Proposition 1. *Let the eigenvalues of the $n \times n$ matrix A be arranged such that $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$. Furthermore, Let u_1 and v_1 be the right and left eigenvectors of A corresponding to λ_1 , respectively. Then the sequence of vectors generated by the power iteration method converges to u_1 in the sense that*

$$\sin(\angle(x_k, u_1)) \leq c \left| \frac{\lambda_2}{\lambda_1} \right|^k$$

provided $v_1^* x_0 \neq 0$.

Our goal is to prove that the exponentiated power iteration exponentially increases the convergence rate and that this increase depends on the learning rate: as the learning rate $\alpha > 0$ increases the convergence rate increases likewise. Our main result in this direction is the following theorem, noting our proof is very similar to the proof of Proposition 1 shown in [Arbenz, 2016, Chapter 7].

Theorem 1. *Let the eigenvalues of the $n \times n$ matrix A be arranged such that $\lambda_1 > |\lambda_2| \geq \dots \geq |\lambda_n|$. Furthermore, Let u_1 and v_1 be the right and left eigenvectors of A corresponding to λ_1 , respectively. Then the sequence of vectors generated by the exponentiated power iteration method with learning rate $\alpha > 0$ converges to u_1 in the sense that*

$$\sin(\angle(x_k, u_1)) \leq c(\exp(\alpha(|\lambda_2| - \lambda_1)))^k$$

provided $v_1^* x_0 \neq 0$.

Proof. Let $\exp(\alpha A) = XJY^*$ be the Jordan normal form of $\exp(\alpha A)$ with $Y^* = X^{-1}$. Then we have that

$$x_k = \exp(\alpha A)x_{k-1} \Rightarrow x_k = XJY^*x_{k-1} \Rightarrow Y^*x_k = JY^*x_{k-1} \Rightarrow Y^*x_k = J^kY^*x_0.$$

The sequence $\{x_k\}$ converges to x_* if and only if the sequence $\{y_k\}$ with $y_k = Y^*x_k$ converges to $y_* = Y^*x_*$. Therefore, we may assume without loss of generality that the matrix being iterated is a Jordan block matrix. To that end, let us iterate J and note that

$$J = \begin{bmatrix} \exp(\alpha\lambda_1) & 0^* \\ 0 & J_2 \end{bmatrix},$$

where λ_1 is the largest in modulus eigenvalue of A , that is, $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$. Let us denote the eigenvector corresponding to $\exp(\alpha\lambda_1)$ by e_1 . We will show that the sequence $\{x_k\}$ converges to e_1 . In particular, we will show that the angle $\angle(x_k, e_1)$ between x_k and e_1 tends to zero as $k \rightarrow \infty$. To that end, let

$$x_k \equiv \begin{bmatrix} x_k^1 \\ y_k^2 \end{bmatrix}$$

with $\|x_k\| = 1$. Note that

$$\sin(\angle(x_k, e_1)) = \|y_k^2\| = \sqrt{\sum_{i=2}^n |x_k(i)|^2}.$$

Omitting the normalization $\|x_k\| = 1$ for convenience, we have

$$\sin(\angle(x_k, e_1)) = \frac{\|y_k^2\|}{\|x_k\|} = \sqrt{\frac{\sum_{i=2}^n |x_k(i)|^2}{\sum_{i=1}^n |x_k(i)|^2}}.$$

From the iteration

$$x_{k+1} = Jx_k, k = 0, 1, 2, \dots$$

we have

$$x_k = \begin{bmatrix} \exp(\alpha\lambda_1) & 0^* \\ 0 & J_2 \end{bmatrix}^k \begin{bmatrix} x_0^1 \\ y_0^2 \end{bmatrix}$$

Defining

$$z_k \equiv \frac{1}{\exp(\alpha\lambda_1)^k} x_k$$

we have

$$z_k = \begin{bmatrix} 1 & 0^* \\ 0 & \frac{1}{\exp(\alpha\lambda_1)} J_2 \end{bmatrix} z_{k-1}.$$

Assuming $z_0^1 = 1$, we have that $z_k^1 = 1$ for all k and

$$z_k^2 = \frac{1}{\exp(\alpha\lambda_1)} J_2 z_{k-1}^2 \quad \frac{1}{\exp(\alpha\lambda_1)} J_2 = \begin{bmatrix} \mu_2 & * & & \\ & \mu_3 & * & \\ & & & \\ & & & \mu_{n-1} & * \\ & & & & \mu_n \end{bmatrix} \quad \mu_k = \frac{\exp(\alpha|\lambda_k|)}{\exp(\alpha\lambda_1)} < 1.$$

Then

$$\sin(\angle(z_k, e_1)) = \frac{\|z_k^2\|}{\|z_k\|} = \frac{z_k^2}{\sqrt{1 + \|z_k^2\|^2}} \leq \|z_k^2\| \leq \left\| \frac{1}{\exp(\alpha\lambda_1)^k} J_2^k \right\| \|z_0^2\|. \quad (2)$$

At this point, we need the following result:

Proposition 2. *Let $\|\cdot\|$ be any matrix norm and let $\rho(M) = \max_i |\lambda_i(M)|$ denote the spectral radius of matrix M . Then*

$$\lim_{k \rightarrow \infty} \|M^k\|^{\frac{1}{k}} = \rho(M).$$

The previous proposition implies that for any $\epsilon > 0$ there is a positive integer $K(\epsilon)$ such that

$$\|M^k\|^{\frac{1}{k}} \leq \rho(M) + \epsilon, \forall k > K(\epsilon).$$

We apply this result to (2). Thus, for any $\epsilon > 0$, there is a $K(\epsilon)$ with

$$\left\| \left(\frac{1}{\exp(\alpha \lambda_1)} J_2 \right)^k \right\|^{\frac{1}{k}} \leq |\mu_2| + \epsilon, \forall k > K(\epsilon).$$

Combining with (2), we obtain that

$$\sin(\angle(z_k, e_1)) \leq (|\mu_2| + \epsilon)^k \|z_0^2\|.$$

We can choose ϵ such that $|\mu_2| + \epsilon < 1$. Thus, the angle between z_k and e_1 goes to zero with a rate $|\mu_2| + \epsilon$ for any small positive ϵ . Since x_k is a scalar multiple of z_k , the same holds for the angle between x_k and e_1 . Since we can choose ϵ arbitrarily small, we have proved that

$$\sin(\angle(x_k, u_1)) \leq c \left| \frac{\exp(\alpha \lambda_2)}{\exp(\alpha \lambda_1)} \right|^k$$

provided that $x_0^1 = e_1^* x_0 \neq 0$.

We have insofar assumed that we iterate the Jordan block matrix J corresponding to $\exp(\alpha A) = X J Y^*$. Assuming now we iterate using $\exp(\alpha A)$, the sequence $y_k = Y^* x_k$ converges to $y_* = a e_1$ with $a \neq 0$. Therefore, x_k converges to a multiple of $X e_1$, which is an eigenvector associated with the largest eigenvalue λ_1 . The condition $e_1^* y_0 \neq 0$ translates to $e_1^*(Y^* x_0) = (Y e_1)^* x_0 \neq 0$. The first column of Y is a left eigenvector corresponding to λ_1 . This completes the proof. \square

3 Algorithmically boosting non-convergent fixed-point iterators

We have previously claimed that algorithmically powered boosting can convert a non-convergent map to one that converges. In this section, we add rigor to the previous discussion on this matter. Toward making this important point, we consider a symmetric zero-sum game, that is, a symmetric bimatrix game (C, C^T) such that the payoff matrix C is antisymmetric (in that $C^T = -C$). We aim to compute or approximate a Nash equilibrium in such a game by repeatedly applying Hedge starting from an interior strategy $X \in \mathring{X}(C)$. After preliminary results, we prove that iterated Hedge fails to converge in this setting. We then prove that averaging iterated Hedge indeed converges. In particular, we give a fully polynomial time approximation scheme for computing an ϵ -approximate symmetric Nash equilibriums strategy. Our proof techniques in this vein are interesting and novel. In Section 6, we leverage our proof techniques to analyze a doubly exponentiated Hedge map.

3.1 PRELIMINARY PROPERTIES OF HEDGE

Let us repeat the Hedge map for convenience:

$$T_i(X) = X(i) \frac{\exp\{\alpha(CX)_i\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}, \quad i = 1, \dots, n. \quad (3)$$

In this section, $\mathcal{C}(X)$ denotes the carrier of X , that is, the pure strategies that support X . Furthermore, $\hat{\mathbb{C}}$ denotes the class of payoff matrices C whose entries lie in the range $[0, 1]$.

3.1.1 Relative entropy (or Kullback-Leibler divergence)

Our analysis of Hedges relies on the relative entropy function between probability distributions (also called *Kullback-Leibler divergence*). The relative entropy between the $n \times 1$ probability vectors $P > 0$ (that is, for all $i = 1, \dots, n$, $P(i) > 0$) and $Q > 0$ is given by

$$RE(P, Q) \doteq \sum_{i=1}^n P(i) \ln \frac{P(i)}{Q(i)}.$$

However, this definition can be relaxed: The relative entropy between $n \times 1$ probability vectors P and Q such that, given P , for all $Q \in \{\mathcal{Q} \in \mathbb{X} | \mathcal{C}(P) \subset \mathcal{C}(\mathcal{Q})\}$, where \mathbb{X} is a probability simplex of appropriate dimension, is

$$RE(P, Q) \doteq \sum_{i \in \mathcal{C}(P)} P(i) \ln \frac{P(i)}{Q(i)}.$$

We note the well-known properties of the relative entropy [Weibull, 1995, p.96] that (i) $RE(P, Q) \geq 0$, (ii) $RE(P, Q) \geq \|P - Q\|^2$, where $\|\cdot\|$ is the Euclidean distance, (iii) $RE(P, P) = 0$, and (iv) $RE(P, Q) = 0$ iff $P = Q$. Note (i) follows from (ii) and (iv) follows from (ii) and (iii).

3.1.2 The convexity lemma

The following lemma generalizes [Freund and Schapire, 1999, Lemma 2].

Lemma 3. *Let T be as in (3). Then*

$$\forall X \in \mathring{\mathbb{X}}(C) \forall Y \in \mathbb{X}(C) : RE(Y, T(X)) \text{ is a convex function of } \alpha.$$

Furthermore, unless X is a fixed point, $RE(Y, T(X))$ is a strictly convex function of α .

Proof. We have

$$\begin{aligned} \frac{d}{d\alpha} RE(Y, \hat{X}) &= \\ &= \frac{d}{d\alpha} \left(\sum_{i \in \mathcal{C}(Y)} Y(i) \ln \left(\frac{Y(i)}{\hat{X}(i)} \right) \right) \\ &= \frac{d}{d\alpha} \left(\sum_{i \in \mathcal{C}(Y)} Y(i) \ln \left(Y(i) \cdot \frac{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}} \right) \right) \end{aligned}$$

$$\begin{aligned}
&= \frac{d}{d\alpha} \left(\sum_{i \in \mathcal{C}(Y)} Y(i) \ln \left(\frac{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}} \right) \right) \\
&= \sum_{i \in \mathcal{C}(Y)} Y(i) \frac{d}{d\alpha} \left(\ln \left(\frac{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}} \right) \right).
\end{aligned}$$

Furthermore, using $(\cdot)'$ as alternative notation (abbreviation) for $d/d\alpha(\cdot)$,

$$\frac{d}{d\alpha} \left(\ln \left(\frac{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}} \right) \right) = \frac{X(i) \exp\{\alpha(CX)_i\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}} \left(\frac{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}} \right)'$$

and

$$\begin{aligned}
\left(\frac{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}} \right)' &= \frac{\sum_{j=1}^n X(j)(CX)_j \exp\{\alpha(CX)_j\} X(i) \exp\{\alpha(CX)_i\}}{(X(i) \exp\{\alpha(CX)_i\})^2} - \\
&- \frac{X(i)(CX)_i \sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\} \exp\{\alpha(CX)_i\}}{(X(i) \exp\{\alpha(CX)_i\})^2} = \\
&= \frac{\sum_{j=1}^n X(j)(CX)_j \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}} - \\
&- \frac{(CX)_i \sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}}{X(i) \exp\{\alpha(CX)_i\}}.
\end{aligned}$$

Therefore,

$$\frac{d}{d\alpha} RE(Y, \hat{X}) = \frac{\sum_{j=1}^n X(j)(CX)_j \exp\{\alpha(CX)_j\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}} - Y \cdot CX. \quad (4)$$

Furthermore,

$$\begin{aligned}
\frac{d^2}{d\alpha^2} RE(Y, \hat{X}) &= \frac{\left(\sum_{j=1}^n X(j)((CX)_j)^2 \exp\{\alpha(CX)_j\} \right) \left(\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\} \right)}{\left(\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\} \right)^2} - \\
&- \frac{\left(\sum_{j=1}^n X(j)((CX)_j) \exp\{\alpha(CX)_j\} \right)^2}{\left(\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\} \right)^2}.
\end{aligned}$$

Jensen's inequality implies that

$$\frac{\sum_{j=1}^n X(j)((CX)_j)^2 \exp\{\alpha(CX)_j\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}} \geq \left(\frac{\sum_{j=1}^n X(j)((CX)_j) \exp\{\alpha(CX)_j\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}} \right)^2,$$

which is equivalent to the numerator of the second derivative being nonnegative as X is a probability vector. Note that the inequality is strict unless

$$\forall i, j \in \mathcal{C}(X) : (CX)_i = (CX)_j.$$

This completes the proof. \square

3.1.3 A version of the convexity lemma

The following lemma is an analogue of [Bertsekas et al., 2003, Lemma 8.2.1, p. 471].

Lemma 4. *Let $C \in \hat{\mathbb{C}}$. Then, for all $Y \in \mathbb{X}(C)$ and for all $X \in \mathring{\mathbb{X}}(C)$, we have that*

$$\forall \alpha > 0 : RE(Y, T(X)) \leq RE(Y, X) - \alpha(Y - X) \cdot CX + \alpha(\exp\{\alpha\} - 1)\bar{C},$$

where $\bar{C} > 0$ is a scalar that can be chosen independent of X and Y .

Proof. Since, by Lemma 3, $RE(Y, T(X)) - RE(Y, X)$ is a convex function of α , we have by the aforementioned secant inequality that, for $\alpha > 0$,

$$RE(Y, T(X)) - RE(Y, X) \leq \alpha (RE(Y, T(X)) - RE(Y, X))' = \alpha \cdot \frac{d}{d\alpha} RE(Y, T(X)). \quad (5)$$

Straight calculus (cf. Lemma 3) implies that

$$\frac{d}{d\alpha} RE(Y, T(X)) = \frac{\sum_{j=1}^n X(j)(CX)_j \exp\{\alpha(CX)_j\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}} - Y \cdot CX.$$

Using Jensen's inequality in the previous expression, we obtain

$$\frac{d}{d\alpha} RE(Y, T(X)) \leq \frac{\sum_{j=1}^n X(j)(CX)_j \exp\{\alpha(CX)_j\}}{\exp\{\alpha X \cdot CX\}} - Y \cdot CX. \quad (6)$$

Note now that

$$\exp\{\alpha x\} \leq 1 + (\exp\{\alpha\} - 1)x, x \in [0, 1], \quad (7)$$

an inequality used in [Freund and Schapire, 1999, Lemma 2]. Using $C \in \hat{\mathbb{C}}$, (6) and (7) imply that

$$\frac{d}{d\alpha} RE(Y, T(X)) \leq \frac{X \cdot CX}{\exp\{\alpha X \cdot CX\}} - Y \cdot CX + (\exp\{\alpha\} - 1) \frac{\sum_{j=1}^n X(j)(CX)_j^2}{\exp\{\alpha X \cdot CX\}}$$

and since $\exp\{\alpha X \cdot CX\} \geq 1$ (again by the assumption that $C \in \hat{\mathbb{C}}$), we have

$$\frac{d}{d\alpha} RE(Y, T(X)) \leq X \cdot CX - Y \cdot CX + (\exp\{\alpha\} - 1) \sum_{j=1}^n X(j)(CX)_j^2.$$

Choosing $\bar{C} = \max \left\{ \sum X(j)(CX)_j^2 \right\}$ and combining with (5) yields the lemma. \square

3.1.4 An instability lemma

The following lemma is crucial in deriving divergence results on multiplicative weights in general. We can prove it in two ways, one invoking the aforementioned convexity lemma and the other by simply invoking Jensen's inequality. We show both proofs.

Lemma 5. *Let $Y, X \in \mathbb{X}(C)$ such that $\mathcal{C}(Y) \subseteq \mathcal{C}(X)$ and such that $Y \neq X$. If X is not a fixed point, then*

$$X \cdot CX - Y \cdot CX \geq 0 \Rightarrow \forall \alpha > 0 : RE(Y, T(X)) - RE(Y, X) > 0.$$

First proof of Lemma 5. Let $\hat{X} = T(X)$. We have, by Jensen's inequality, that

$$\begin{aligned}
RE(Y, \hat{X}) - RE(Y, X) &= \sum_{i \in \mathcal{C}(Y)} Y(i) \ln \frac{Y(i)}{\hat{X}(i)} - \sum_{i \in \mathcal{C}(Y)} Y(i) \ln \frac{Y(i)}{X(i)} \\
&= - \sum_{i \in \mathcal{C}(Y)} Y(i) \ln \hat{X}(i) + \sum_{i \in \mathcal{C}(Y)} Y(i) \ln X(i) \\
&= \sum_{i \in \mathcal{C}(Y)} Y(i) \ln \frac{X(i)}{\hat{X}(i)} \\
&= \sum_{i \in \mathcal{C}(Y)} Y(i) \ln \left(\frac{\sum_{j=1}^n X(j) \exp\{\alpha E_j \cdot CX\}}{\exp\{\alpha E_i \cdot CX\}} \right) \\
&= \ln \left(\sum_{j=1}^n X(j) \exp\{\alpha E_j \cdot CX\} \right) - \sum_{i \in \mathcal{C}(Y)} Y(i) \ln (\exp\{\alpha E_i \cdot CX\}) \\
&= \ln \left(\sum_{j=1}^n X(j) \exp\{\alpha E_j \cdot CX\} \right) - \alpha \sum_{i \in \mathcal{C}(Y)} Y(i) (E_i \cdot CX) \\
&\geq \alpha \sum_{j=1}^n X(j) (E_j \cdot CX) - \alpha \sum_{i \in \mathcal{C}(Y)} Y(i) (E_i \cdot CX) \\
&= \alpha (X \cdot CX - Y \cdot CX).
\end{aligned}$$

Therefore,

$$X \cdot CX - Y \cdot CX \geq 0 \Rightarrow \forall \alpha > 0 : RE(Y, \hat{X}) - RE(Y, X) \geq 0.$$

If $\max_{i \in \mathcal{C}(X)} \{(CX)_i\} > \min_{i \in \mathcal{C}(X)} \{(CX)_i\}$, since Jensen's inequality is strict,

$$X \cdot CX - Y \cdot CX \geq 0 \Rightarrow \forall \alpha > 0 : RE(Y, \hat{X}) - RE(Y, X) > 0.$$

This completes the proof. \square

Second proof of Lemma 5. By the convexity lemma (Lemma 3), unless X is a fixed point, $RE(Y, \hat{X})$ is strictly convex. (4) implies that

$$\left. \frac{d}{d\alpha} RE(Y, \hat{X}) \right|_{\alpha=0} = (X - Y) \cdot CX \geq 0.$$

Noting that $RE(Y, \hat{X})|_{\alpha=0} = RE(Y, X)$ completes the proof. \square

3.2 DIVERGENCE OF ITERATED HEDGE

Let us now get to the proof that iterated Hedge diverges, in particular, in symmetric zero-sum games equipped with an interior equilibrium. Given an antisymmetric matrix C equipped with an interior equilibrium, say X^* , it is simple to show that X^* satisfies the relation

$$\forall X \in \mathbb{X}(C) : (X^* - X) \cdot CX = 0.$$

As an example, consider the *rock-paper-scissors game*, which is a zero-sum symmetric bimatrix game with payoff matrix

$$C = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}.$$

C is anti-symmetric, that is, $C^T = -C$, therefore, for all X , $X \cdot CX = 0$. Furthermore, $X^* = (1/3, 1/3, 1/3)$ is the unique equilibrium strategy, implying after straight algebra that, for all $X \in \mathbb{X}(C)$, $(X^* - X) \cdot CX = 0$. Lemma 5 implies that starting anywhere in the interior of $\mathbb{X}(C)$ other than the uniform strategy (which is the equilibrium strategy), under any sequence of positive learning rates, the relative entropy distance between X^* and $T^k(X^0)$ diverges to ∞ as $k \rightarrow \infty$.

3.3 PRECISE BOUNDS ON EQUILIBRIUM APPROXIMATION

Let us now prove that, in sharp contrast, averaging iterated Hedge converges. In fact, we give an equilibrium fully polynomial time approximation scheme: Given any desired equilibrium approximation error ϵ , we compute a fixed learning rate α such that the average of iterated Hedge converges to an ϵ -approximate Nash equilibrium of the corresponding symmetric zero-sum game.

Lemma 6. *Let $C \in \hat{\mathbb{C}}$. Then, for all $Y \in \mathbb{X}(C)$ and for all $X \in \mathring{\mathbb{X}}(C)$, we have that*

$$\forall \alpha > 0 : RE(Y, T(X)) \leq RE(Y, X) - \alpha(Y - X) \cdot CX + \alpha(\exp\{\alpha\} - 1).$$

Proof. Using the assumption $C \in \hat{\mathbb{C}}$, we have

$$\sum_{j=1}^n X(j)(CX)_j^2 \leq \sum_{j=1}^n X(j)(CX)_j \leq 1.$$

Therefore,

$$\frac{d}{d\alpha} RE(Y, T(X)) \leq X \cdot CX - Y \cdot CX + (\exp\{\alpha\} - 1).$$

Combining with (5) yields the lemma. \square

Lemma 7. *Let $C \in \hat{\mathbb{C}}$, $X^k \equiv T^k(X^0)$, $X^0 \in \mathring{\mathbb{X}}(C)$, and assume $\alpha > 0$ is held constant. Then,*

$$\forall \theta > 0 \forall Y \in \mathbb{X}(C) : \frac{1}{K+1} \sum_{k=0}^K (Y - X^k) \cdot CX^k \leq (\exp\{\alpha\} - 1) + \theta \quad (8)$$

where $K \geq \lfloor RE(Y, X^0)/(\alpha\theta) \rfloor$. If X^0 is the uniform strategy, then (8) holds after $\lfloor \ln(n)/(\alpha\theta) \rfloor$ iterations and continues to hold thereafter.

Proof. Assume for the sake of contradiction that (8) does not hold, that is,

$$\frac{1}{K+1} \sum_{k=0}^K (Y - X^k) \cdot CX^k > (\exp\{\alpha\} - 1) + \theta. \quad (9)$$

Invoking Lemma 6,

$$RE(Y, X^{k+1}) \leq RE(Y, X^k) - \alpha(Y - X^k) \cdot CX^k + \alpha(\exp\{\alpha\} - 1).$$

Summing over $k = 0, \dots, K$, we obtain

$$RE(Y, X^{K+1}) \leq RE(Y, X^0) - \alpha \sum_{k=0}^K (Y - X^k) \cdot CX^k + (K+1)\alpha(\exp\{\alpha\} - 1)$$

and, therefore,

$$\frac{RE(Y, X^{K+1})}{K+1} \leq \frac{RE(Y, X^0)}{K+1} - \frac{\alpha}{K+1} \sum_{k=0}^K (Y - X^k) \cdot CX^k + \alpha(\exp\{\alpha\} - 1). \quad (10)$$

Substituting then (9) in (10) we obtain

$$\frac{RE(Y, X^{K+1})}{K+1} \leq \frac{RE(Y, X^0)}{K+1} - \alpha\theta,$$

which implies that

$$RE(Y, X^{K+1}) \leq RE(Y, X^0) - (K+1)\alpha\theta$$

and, therefore, that

$$RE(Y, X^0) \geq (K+1)\alpha\theta.$$

But this contradicts the previous definition of K and completes the proof.

The second part of the lemma is implied from the observation that a convex function is maximized at the boundary and, in our particular case, the vertices of the probability simplex. \square

Theorem 2. *Let $C \in \hat{\mathbb{C}}$ be such that it has been obtained by an affine transformation on a anti-symmetric matrix. Then starting at the uniform strategy, the average of iterated Hedge converges to an ϵ -approximate symmetric Nash equilibrium strategy in at most*

$$\left\lceil \frac{\ln(n)}{\frac{\epsilon}{2} \ln(1 + \frac{\epsilon}{2})} \right\rceil$$

iterations using a fixed learning rate equal to $\ln(1 + \epsilon/2)$.

Proof. This theorem is a simple implication of Lemma 7. Note that since Y is arbitrary in (8), we may write it as

$$\max_{i=1}^n \left\{ C \left(\frac{1}{K+1} \sum_{k=0}^K X^k \right) \right\} - \frac{1}{K+1} \sum_{k=0}^K X^k \cdot CX^k \leq (\exp\{\alpha\} - 1) + \theta.$$

Using the notation

$$\bar{X}^K \equiv \frac{1}{K+1} \sum_{k=0}^K X^k$$

and using also the assumption that C has been obtained by an affine transformation on a antisymmetric matrix, we obtain that

$$(C\bar{X}^K)_{\max} - \bar{X}^K \cdot C\bar{X}^K \leq (\exp\{\alpha\} - 1) + \theta.$$

Letting $\theta = \epsilon/2$ and $\alpha = \ln(1 + \epsilon/2)$ and applying Lemma 7, we obtain the theorem. \square

We note that the previous analysis nearly exactly matches the bound in [Freund and Schapire, 1999, Section 6.1] although these bounds have been obtained using different analytical routes.

4 A variational perspective on algorithmic boosting

Our definition of algorithmic boosting of discrete maps extends in a natural manner to continuous flows. In this section, we consider algorithmically boosting the replicator dynamic, which is given by the following differential equation:

$$\dot{X}(i) = X(i) ((CX)_i - X \cdot CX), \quad i = 1, \dots, n.$$

In fact, from the perspective of computing Nash equilibria in symmetric bimatrix games (and, more generally, solving *variational inequalities* over the standard simplex), Hedge can be meaningfully understood as a discretization of the replicator dynamic. Our first task in this section is to prove this duality between Hedge and the replicator dynamic. Then, as our main result in this section, we prove that the ω -limit set of the long-time average of the replicator dynamic in a symmetric zero-sum game consists entirely of Nash equilibria in this game. In this result, we observe a phenomenon that is analogous to that of the previous section, namely, that although an orbit may not in itself converge to the desired fixed point, by algorithmically boosting the orbit we obtain convergence.

4.1 HEDGE IS A DISCRETIZATION OF THE REPLICATOR DYNAMIC

In this part of this section, we assume that C is an, in general, nonlinear operator. Let us first note that *consistency* and *convergence* are standard properties numerical integrators satisfy (for example, see [Burden et al., 2011]). That Hedge is a consistent numerical integrator for the replicator dynamic rests on the observation that

$$\left. \frac{d}{d\alpha} \left(X(i) \frac{\exp\{\alpha(CX)_i\}}{\sum_{j=1}^n X(j) \exp\{\alpha(CX)_j\}} \right) \right|_{\alpha=0} = X(i) ((CX)_i - X \cdot CX), \quad i = 1, \dots, n.$$

We note that under a stochastic model of evolution, a similar observation has been leveraged for the study of dynamics in *congestion games* by Kleinberg et al. [2009]. Using the previous observation, we also prove convergence by comparing the error Hedge generates relative to the Euler method, which approximates the replicator dynamic using iterates generated by the difference equation

$$W^{k+1}(i) = W^k(i) + \alpha W(i) ((CW)_i - W \cdot CW), \quad k = 0, \dots, K.$$

Starting from the interior of the simplex, for any finite K , there exists α such that Euler's method remains in the interior. Therefore, for small enough time step, Euler's method remains well-defined given any number of finite iterations. Euler's method is convergent under the assumption that the replicator equation is Lipschitz and under the assumption that the second derivative of the solution trajectory with respect to time is bounded. We have the following theorem:

Theorem 3. *Under the aforementioned assumptions that ensure that the Euler method is a convergent numerical integrator for the replicator dynamic and under the further assumption that $\{\max\{\|CX\| \mid X \in \mathbb{X}(C)\}\} < \infty$, Hedge is a convergent numerical integrator for the replicator dynamic.*

Proof. The Taylor expansion of T at $\alpha = 0$ gives

$$T_i(X) = X(i) + \alpha X(i)(E_i \cdot CX - X \cdot CX) + \frac{\alpha^2}{2} \left. \frac{d^2 T_i(X)}{d\alpha^2} \right|_{\alpha=\xi_i}.$$

Using the notation

$$F_i(X) \equiv X(i)(E_i \cdot CX - X \cdot CX),$$

we obtain

$$T_i(X) = X(i) + \alpha F_i(X) + \frac{\alpha^2}{2} \frac{d^2 T_i(X)}{d\alpha^2} \Big|_{\alpha=\xi_i}.$$

Let us assume F is Lipschitz with constant L . Furthermore, under the assumption that $\{\max\{\|CX\| | X \in \mathbb{X}(C)\} < \infty$, there exists a positive constant M such that

$$\forall i = 1, \dots, n \quad \forall X \in \mathbb{X}(C) : \left| \frac{d^2 T_i(X)}{d\alpha^2} \right| \leq M.$$

To show convergence, note that Hedge gives

$$X^{k+1}(i) = X^k(i) + \alpha F_i(X^k) + \frac{\alpha^2}{2} \frac{d^2 T_i(X)}{d\alpha^2} \Big|_{\alpha=\xi_i^k}$$

whereas Euler's method gives

$$W^{k+1}(i) = W^k(i) + \alpha F_i(W^k).$$

Subtracting these equations, we obtain

$$X^{k+1}(i) - W^{k+1}(i) = X^k(i) - W^k(i) + \alpha(F_i(X^k) - F_i(W^k)) + \frac{\alpha^2}{2} \frac{d^2 T_i(X)}{d\alpha^2} \Big|_{\alpha=\xi_i^k}$$

and hence

$$|X^{k+1}(i) - W^{k+1}(i)| \leq |X^k(i) - W^k(i)| + \alpha |F_i(X^k) - F_i(W^k)| + \frac{\alpha^2}{2} \left| \frac{d^2 T_i(X)}{d\alpha^2} \Big|_{\alpha=\xi_i^k} \right|$$

Since, as noted above, F is Lipschitz with parameter L and

$$\left| \frac{d^2 T_i(X)}{d\alpha^2} \Big|_{\alpha=\xi_i^k} \right| \leq M$$

we obtain

$$|X^{k+1}(i) - W^{k+1}(i)| \leq (1 + \alpha L) |X^k(i) - W^k(i)| + \frac{\alpha^2 M}{2}.$$

To proceed further, we need the following lemma:

Lemma 8. *If s and t are positive real numbers, $\{a_i\}_{i=0}^k$ is a sequence satisfying $a_0 \geq -t/s$ and*

$$a_{i+1} \leq (1 + s)a_i + t, \forall i = 0, \dots, k-1$$

then

$$a_{i+1} \leq \exp\{(i+1)s\} \left(a_0 + \frac{t}{s} \right) - \frac{t}{s}.$$

Proof. See [Burden et al., 2011]. □

Applying Lemma 8, we further obtain

$$|X^{k+1}(i) - W^{k+1}(i)| \leq \exp\{(k+1)\alpha L\} \left(|X^0(i) - W^0(i)| + \frac{\alpha M}{2L} \right) - \frac{\alpha M}{2L}$$

which implies

$$|X^{k+1}(i) - W^{k+1}(i)| \leq (\exp\{(k+1)\alpha L\} - 1) \frac{\alpha M}{2L}.$$

Therefore, as $\alpha \rightarrow 0$, we have that $|X^{k+1}(i) - W^{k+1}(i)| \rightarrow 0$, and, given that Euler's method is convergent, Hedge is similarly convergent. □

4.2 CONVERGENCE OF THE LONG-RUN AVERAGE OF THE REPLICATOR DYNAMIC

Theorem 4. *Let C be the payoff matrix of a symmetric zero-sum game (C, C^T) . Furthermore, let $X_\alpha, \alpha \in [0, t]$ be an orbit of the replicator dynamic*

$$\dot{X} = X(i)((CX)_i - X \cdot CX), i = 1, \dots, n.$$

Then the ω -limit set of the long-run average

$$\frac{1}{t} \int_0^t X_\omega d\omega$$

consists entirely of symmetric Nash equilibrium strategies of (C, C^T) .

Proof. Let

$$\bar{X}_t = \frac{1}{t} \int_0^t X_\omega d\omega$$

where $X_\omega, \omega \in [0, t]$ is a trajectory of the replicator dynamic. Furthermore, let $\{\bar{X}_{t_n}\}_0^\infty$ be a convergent subsequence such that $\{\bar{X}_{t_n}\} \rightarrow \bar{X}$. That is, \bar{X} is in the ω -limit set of the long-time average \bar{X}_t . We will show that \bar{X} is a Nash equilibrium strategy. To that end, let

$$\left\{ \frac{1}{t_m} \int_0^{t_m} (X_\omega \cdot CX_\omega) d\omega \right\}_{m=0}^\infty$$

be a convergent subsequence of the sequence of long-time averages. Note now that letting $Y \in \mathbb{X}(C)$ be arbitrary, since

$$\int_0^t \frac{d}{d\omega} RE(Y, X_\omega) d\omega = RE(Y, X_t) - RE(Y, X_0),$$

we obtain that

$$\liminf_{m \rightarrow \infty} \left\{ \frac{1}{t_m} \int_0^{t_m} \frac{d}{d\omega} RE(Y, X_\omega) d\omega \right\} \geq 0$$

where we have used that the relative entropy function is nonnegative. Straight calculus (for example, see [Weibull, 1995, p. 98]) gives that

$$\frac{d}{d\omega} RE(Y, X_\omega) = X_\omega \cdot CX_\omega - Y \cdot CX_\omega.$$

Combining the previous relations, we obtain

$$\forall Y \in \mathbb{X}(C) : Y \cdot C\bar{X} \leq \lim_{m \rightarrow \infty} \left\{ \frac{1}{t_m} \int_0^{t_m} (X_\omega \cdot CX_\omega) d\omega \right\}. \quad (11)$$

Now let $\{X_{t_k}\}_0^\infty$ be a convergent subsequence of the sequence

$$\left\{ \frac{1}{t_m} \int_0^{t_m} (X_\omega \cdot CX_\omega) d\omega \right\}_{m=0}^\infty$$

that converges to X . Furthermore, let E_i be a pure strategy in the carrier of X . Then, since the relative entropy is bounded, we obtain that

$$\lim_{k \rightarrow \infty} \left\{ \frac{1}{t_k} \int_0^{t_k} \frac{d}{d\omega} RE(E_i, X_\omega) d\omega \right\} = 0,$$

which implies that

$$(C\bar{X})_i = \lim_{m \rightarrow \infty} \left\{ \frac{1}{t_m} \int_0^{t_m} (X_\omega \cdot CX_\omega) d\omega \right\}.$$

Therefore, for every pure strategy E_i in the carrier of X , we have that

$$(C\bar{X})_i = \lim_{m \rightarrow \infty} \left\{ \frac{1}{t_m} \int_0^{t_m} (X_\omega \cdot CX_\omega) d\omega \right\}$$

and, for every pure strategy E_j outside the carrier of X , we have that

$$(C\bar{X})_j \leq \lim_{m \rightarrow \infty} \left\{ \frac{1}{t_m} \int_0^{t_m} (X_\omega \cdot CX_\omega) d\omega \right\}$$

which implies that

$$(C\bar{X})_{\max} = \lim_{m \rightarrow \infty} \left\{ \frac{1}{t_m} \int_0^{t_m} (X_\omega \cdot CX_\omega) d\omega \right\}.$$

Assuming now the game is zero-sum so that C is an antisymmetric matrix (which implies that, for all x , $x \cdot Cx = 0$), we obtain from the previous inequality that

$$(C\bar{X})_{\max} = 0 = \bar{X} \cdot C\bar{X},$$

and, therefore, \bar{X} is a Nash equilibrium as claimed. \square

5 The importance of a correct implementation of exponentiation

A perspective on algorithmic boosting is that it generalizes the operation of taking the long-run average of an orbit. We have insofar focused on symmetric zero-sum games, where the long-run average of iterated Hedge had been known to converge to an approximate Nash equilibrium prior to our results. In this section, we focus on a symmetric bimatrix game that is not zero-sum, wherein the long-run average of iterated Hedge can, in principle, diverge. It is an open question if this is indeed the case. In this section, our main contribution is a numerical phenomenon whereby using the standard implementation of the exponential function in the computation of iterated Hedge, the average diverges whereas using an accurate implementation of the exponential function (in a fashion customized to this particular setting of computing Nash equilibria) the average converges. The example where we document this phenomenon is a great environment for testing iterative algorithms for computing Nash equilibria using the principles of algorithmic boosting.

5.1 THE SHAPLEY GAME

To the extent of our knowledge, the first study of the average of iterated Hedge outside the realm of zero-sum environments was by Daskalakis et al. [2010]. [Daskalakis et al., 2010, Theorem 1] shows divergence of the average of iterated Hedge playing against iterated Hedge (they consider a 2-player setting) in Shapley’s 3×3 symmetric bimatrix game whose payoff matrix is

$$C = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}$$

and whose unique Nash equilibrium is the symmetric Nash equilibrium corresponding to the uniform strategy $(1/3, 1/3, 1/3)$. Their result casts doubt that learning algorithms (used as fixed point iterators) can compute Nash equilibria. In this section (and broadly in this paper), we cast hope.

5.2 THE SYMMETRIC SHAPLEY GAME

In the rest of this section, we are concerned with the following symmetrization of Shapley’s game:

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the unique Nash equilibrium is the symmetric Nash equilibrium corresponding to the uniform strategy. In our numerical experiment, which is carried out in Matlab, we compare two different implementations of Hedge. The first implementation uses the aforementioned algebraic expression for generating iterates where the exponential function is implemented by Matlab’s `exp()` routine. The second implementation is based on a formulation of Hedge as the solution of a convex optimization problem, in particular, as Krichene et al. [2015] point out

$$T(X) = \arg \min \{ RE(Y, X) - \alpha Y \cdot CX \mid Y \in \mathbb{X}(C) \},$$

where $RE(Y, X)$ is the relative entropy distance (Kullback-Leibler divergence) between probability vectors Y and X (as defined earlier). Note that this optimization problem is a *relative entropy*

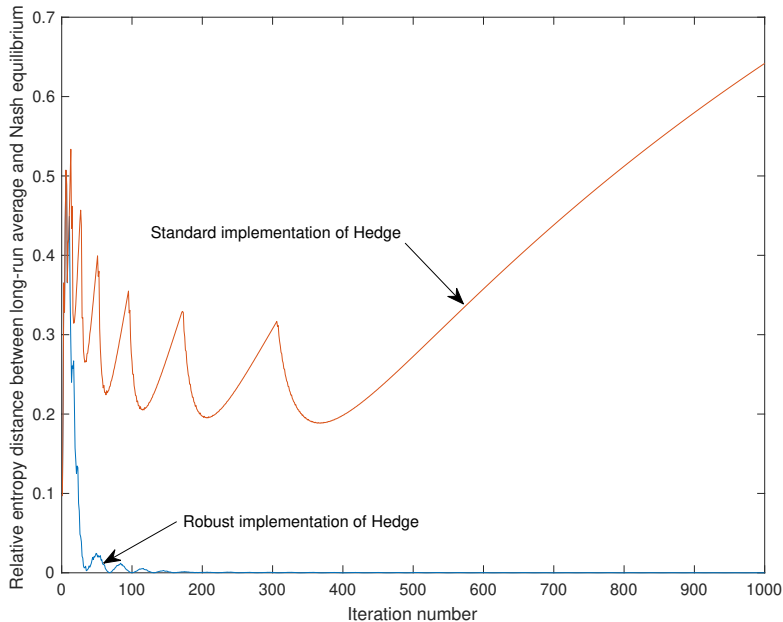


Figure 1: Comparison of the standard implementation of Hedge and the robust implementation using relative entropy optimization. The learning rate is equal to 10 and the initial condition is identical. The figure illustrates the large discrepancy between the two different implementations of the same algorithm.

program [Chandrasekaran and Shah, 2017] that admits a polynomial-time *interior point method* for its exact solution [Nesterov and Nemirovski, 1994]. In our experiment, we initialize both implementations with the probability vector $(0.1, 0.2, 0.3, 0.2, 0.1, 0.1)$ and use a learning rate equal to 10. Figure 1 plots the relative entropy distance between the long-run average of the iterates under the two implementations and the Nash equilibrium strategy. It is clear in the figure that the standard implementation gives divergence whereas the robust implementation using relative entropy programming (implemented using Matlab’s `fmincon` interior point solver) gives convergence.

5.3 DISCUSSION

In closing this section, we would like to make two key observations. The first is that the long-run average of the replicator dynamic converges in the symmetric Shapley game. This is simple to check using Matlab’s standard numerical integrator, namely, `ode45`. The second observation is that the principle of exponentiating and normalizing (for example, projecting onto the standard simplex) is used in a variety of machine learning tasks and in software code that is deployed in the field. Our experiment clearly demonstrates that it is not unlikely that implementations of the exponential function can trigger behavior different from what is expected or sought for simply because caution has not been paid to the correct implementation of exponentiation. Our hope is our experiment, and broadly this paper, squarely places the importance of correct implementations of exponentiation as a desideratum in field deployments of machine learning and fixed-point computation systems.

6 A simplified numerically stable Hedge map

In this section, we aim to restore stability in the Hedge map without resorting to relative entropy programming. To that end, we devise a map that is conceptually simpler than Hedge and easier to implement. Our map is obtained by the doubly exponentiated Hedge map

$$T_i(X) = X(i) \frac{\exp\{\alpha \exp\{\beta((CX)_i - (CX)_{\max})\}\}}{\sum_{j=1}^n X(j) \exp\{\alpha \exp\{\beta((CX)_j - (CX)_{\max})\}\}}, \quad i = 1, \dots, n \quad (12)$$

as $\beta \rightarrow \infty$, which gives the map which gives the map

$$T_i(X) = X(i) \frac{\exp\{\alpha Q(i)\}}{\sum_{j=1}^n X(j) \exp\{\alpha Q(j)\}} \quad Q(i) = \begin{cases} 1, & i \in \arg \max\{(CX)_j\} \\ 0, & i \notin \arg \max\{(CX)_j\} \end{cases} \quad i = 1, \dots, n. \quad (13)$$

Our analysis parallels that of Section 3. The analogue of Lemma 4 is:

Lemma 9. *Let $C \in \hat{\mathbb{C}}$ and T be as in (13) where the learning rate $\alpha = a(CX)_{\max}$. Then, for all $Y \in \arg \max\{(CX)_j\}$ and for all $X \in \mathring{\mathbb{X}}(C)$, we have*

$$RE(Y, T(X)) - RE(Y, X) \leq -a(Y \cdot CX - X \cdot CX) + a \left(\frac{(\exp\{a\} - 1) - a}{a} \right).$$

Proof. Letting T be as in (12), it is a tedious calculation but it is straightforward to verify given Lemma 3 that

$$\begin{aligned} \frac{d}{d\alpha} RE(Y, T(X)) &= \frac{\sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \exp\{\alpha \exp\{\beta((CX)_j - (CX)_{\max})\}\}}{\sum_{j=1}^n X(j) \exp\{\alpha \exp\{\beta((CX)_j - (CX)_{\max})\}\}} - \\ &\quad - \sum_{i \in \mathcal{C}(Y)} Y(i) \exp\{\beta((CX)_i - (CX)_{\max})\}. \end{aligned}$$

Using Jensen's inequality in the previous expression, we obtain

$$\begin{aligned} \frac{d}{d\alpha} RE(Y, T(X)) &\leq \frac{\sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \exp\{\alpha \exp\{\beta((CX)_j - (CX)_{\max})\}\}}{\exp\left\{\alpha \sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\}\right\}} - \\ &\quad - \sum_{i \in \mathcal{C}(Y)} Y(i) \exp\{\beta((CX)_i - (CX)_{\max})\}, \end{aligned}$$

which implies

$$\begin{aligned} \frac{d}{d\alpha} RE(Y, T(X)) &\leq \sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \times \\ &\quad \times \exp\left\{\alpha \left(\exp\{\beta((CX)_j - (CX)_{\max})\} - \sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \right)\right\} - \\ &\quad - \sum_{i \in \mathcal{C}(Y)} Y(i) \exp\{\beta((CX)_i - (CX)_{\max})\}. \end{aligned}$$

Integrating with respect to α gives

$$\begin{aligned}
RE(Y, T(X)) - RE(Y, X) &\leq \sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \times \\
&\times \left(\exp\{\beta((CX)_j - (CX)_{\max})\} - \sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \right)^{-1} \times \\
&\times \left(\exp \left\{ \alpha \left(\exp\{\beta((CX)_j - (CX)_{\max})\} - \sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \right) \right\} - 1 \right) - \\
&- \alpha \sum_{i \in \mathcal{C}(Y)} Y(i) \exp\{\beta((CX)_i - (CX)_{\max})\}.
\end{aligned}$$

Rearranging and cancelling terms, the right-hand-side of the previous inequality is

$$\begin{aligned}
&\sum_{j=1}^n X(j) \frac{\exp\{\beta(CX)_j\}}{\sum_{j=1}^n X(j) \exp\{\beta(CX)_j\}} \left(\frac{\exp\{\beta(CX)_j\}}{\sum_{j=1}^n X(j) \exp\{\beta(CX)_j\}} - 1 \right)^{-1} \times \\
&\times \left(\exp \left\{ \alpha \left(\sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \right) \right\} \left(\frac{\exp\{\beta(CX)_j\}}{\sum_{j=1}^n X(j) \exp\{\beta(CX)_j\}} - 1 \right) \right\} - 1 \right) - \\
&- \alpha \sum_{i \in \mathcal{C}(Y)} Y(i) \exp\{\beta((CX)_i - (CX)_{\max})\}.
\end{aligned}$$

As $\beta \rightarrow \infty$, the previous expression simplifies to

$$\begin{aligned}
&\left(\frac{\exp\{\beta(CX)_{\max}\}}{\sum_{j=1}^n X(j) \exp\{\beta(CX)_j\}} - 1 \right)^{-1} \times \\
&\times \left(\exp \left\{ \alpha \left(\sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \right) \right\} \left(\frac{\exp\{\beta(CX)_{\max}\}}{\sum_{j=1}^n X(j) \exp\{\beta(CX)_j\}} - 1 \right) \right\} - 1 \right) - \\
&- \alpha \sum_{i \in \mathcal{C}(Y)} Y(i) \exp\{\beta((CX)_i - (CX)_{\max})\}. \tag{14}
\end{aligned}$$

As $\beta \rightarrow \infty$, we have that

$$\left(\sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \right) \left(\frac{\exp\{\beta(CX)_{\max}\}}{\sum_{j=1}^n X(j) \exp\{\beta(CX)_j\}} - 1 \right) \rightarrow 1 - \sum_{j \in \arg \max\{(CX)_j\}} X(j).$$

Using the inequality $\exp\{ax\} \leq 1 + (\exp\{a\} - 1)x$, $x \in [0, 1]$, (14) simplifies to

$$(\exp\{\alpha\} - 1) \left(\sum_{j=1}^n X(j) \exp\{\beta((CX)_j - (CX)_{\max})\} \right) - \alpha \left(\sum_{i \in \mathcal{C}(Y)} Y(i) \exp\{\beta((CX)_i - (CX)_{\max})\} \right)$$

and as $\beta \rightarrow \infty$, the previous expression further simplifies to

$$(\exp\{\alpha\} - 1) \left(\sum_{j \in \arg \max\{(CX)_j\}} X(j) \right) - \alpha \left(\sum_{i \in \mathcal{C}(Y) \cap \arg \max\{(CX)_j\}} Y(i) \right).$$

Letting $\alpha = a(CX)_{\max}$ and using again the inequality $\exp\{ax\} \leq 1 + (\exp\{a\} - 1)x$, $x \in [0, 1]$, we obtain

$$(\exp\{a\} - 1) \left(\sum_{j \in \arg \max\{(CX)_j\}} X(j) \right) (CX)_{\max} - a(CX)_{\max}. \quad (15)$$

Since

$$X \cdot CX = \left(\sum_{j \in \arg \max\{(CX)_j\}} X(j) \right) (CX)_{\max} + \sum_{j \notin \arg \max\{(CX)_j\}} X(j)(CX)_j,$$

we obtain that

$$\left(\sum_{j \in \arg \max\{(CX)_j\}} X(j) \right) (CX)_{\max} \leq X \cdot CX,$$

which implies that (15) is upper bounded by

$$(\exp\{a\} - 1) X \cdot CX - a(CX)_{\max}.$$

Straight algebra completes the proof. \square

Lemma 10. *Let $C \in \hat{\mathbb{C}}$, $X^k \equiv T^k(X^0)$, $X^0 \in \mathring{\mathbb{X}}(C)$, T as in (13), and assume $\alpha = a(CX^k)_{\max}$. Then,*

$$\forall \theta > 0 : \frac{1}{K+1} \sum_{k=0}^K (CX^k)_{\max} - \frac{1}{K+1} \sum_{k=0}^K X^k \cdot CX^k \leq \frac{(\exp\{a\} - 1) - a}{a} + \theta \quad (16)$$

where $K \geq \lfloor RE(Y, X^0)/(a\theta) \rfloor$. If X^0 is the uniform strategy, then (16) holds after $\lfloor \ln(n)/(a\theta) \rfloor$ iterations and continues to hold thereafter.

Proof. The proof is entirely analogous to that of Lemma 7 using Lemma 9. \square

Theorem 5. *Let $C \in \hat{\mathbb{C}}$ be such that it has been obtained by an affine transformation on a anti-symmetric matrix. Then starting (13) at the uniform strategy, in at most*

$$\left\lfloor \frac{\ln(n)}{\frac{\epsilon}{2} \ln(1 + \frac{\epsilon}{2})} \right\rfloor$$

iterations using a variable learning rate equal to $\ln(1 + \epsilon/2)(CX^k)_{\max}$, the average of iterated (13) converges to an

$$\left(\frac{\epsilon}{2} + \frac{\frac{\epsilon}{2} - \ln(1 + \frac{\epsilon}{2})}{\ln(1 + \frac{\epsilon}{2})} \right) \text{-approximate}$$

symmetric Nash equilibrium strategy, closing matching the bound of Hedge.

Proof. This theorem is a simple implication of Lemma 10. Using the notation

$$\bar{X}^K \equiv \frac{1}{K+1} \sum_{k=0}^K X^k$$

note that

$$\frac{1}{K+1} \sum_{k=0}^K (CX^k)_{\max} \geq (C\bar{X}^K)_{\max}.$$

Using also the assumption that C has been obtained by an affine transformation on a antisymmetric matrix, we obtain that

$$(C\bar{X}^K)_{\max} - \bar{X}^K \cdot C\bar{X}^K \leq \frac{(\exp\{a\} - 1) - a}{a} + \theta.$$

Letting $\theta = \epsilon/2$ and $a = \ln(1 + \epsilon/2)$ and applying Lemma 7, we obtain the theorem. \square

7 Summary and future work

In this paper, we developed a theory of *algorithmic exponentiation* based on the *matrix exponential* and its generalization to *nonlinear* fixed-point computation systems. (1) Using this theory, we showed that exponentiating convergent *iterated linear maps* (for example, the Google matrix in PageRank) gives an exponential increase in their convergence rate. (2) A slight generalization of algorithmic exponentiation is *algorithmic boosting*, the principle of computing fixed points by averaging the powers of an iterated map. Focusing on game theory, we devised a simplified version of Hedge that attains similar equilibrium approximation bounds in symmetric zero-sum games while at the same time being more numerically stable than the Hedge map itself. Our simplified algorithm was obtained and analyzed by *doubly exponentiating* and letting the learning rate of the upper exponent go to infinity. (3) We also considered a *variational approach* to algorithmic boosting in the same game-theoretic setting and showed that every limit point of the long-run average of the replicator dynamic (in a symmetric zero-sum game) is a Nash equilibrium. (4) Our theory suggests that using, for example, Padé approximations to compute the matrix exponential, using relative entropy programming to implement Hedge, and using a second exponent in Hedge (the learning rate going to infinity) are different implementations of the same algorithmic boosting principle.

Our paper raises a host of interesting questions but we single out one question as perhaps the most important one, namely, to develop techniques for computing Nash equilibria (whether in zero-sum games or in the general case) using Padé approximation theory: We have previously discussed how Padé approximation theory can be used to compute the matrix exponential. In fact, this theory also applies to the approximation of any function even if its Taylor expansion diverges. This suggests a general method to compute a Nash equilibrium, namely, to prove that an algorithmically powered version of a dynamic converges to a Nash equilibrium and then to approximate that Nash equilibrium using Padé approximation theory. That such an idea should render a polynomial-time algorithm to compute a Nash equilibrium in a symmetric zero-sum game should be a low-hanging fruit. What is perhaps more compelling is to generalize this idea to more general classes of symmetric bimatrix games and, what not, even under an arbitrary matrix.

Acknowledgments

The idea that Hedge is a discretization of the replicator dynamic is by Professor Yannis Kevrekidis. The first author would like to thank him for this contribution and other helpful discussions.

References

- E. Akin and V. Losert. Evolutionary dynamics of zero-sum games. *Journal of Mathematical Biology*, 20:231–258, 1984.
- P. Arbenz. Lecture notes on solving large eigenvalue problems. <https://people.inf.ethz.ch/arbenz/ewp/Lnotes/lsevp.pdf>, 2016.
- M. Arioli, B. Codenotti, and C. Fassino. The Padé method for computing the matrix exponential. *Linear Algebra and Its Applications*, 240:111–130, 1996.
- S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: A meta-algorithm and its applications. *Theory of Computing*, 8:121–164, 2012.
- I. Avramopoulos. Computational principles manifesting in learning symmetric equilibria by exponentiated dynamics. SSRN preprint <https://ssrn.com/abstract=4367128>, 2023.
- D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- O. Biggar and I. Shames. The replicator dynamic, chain components and the response graph. In *Proc. 34th International Conference on Algorithmic Learning Theory*, 2023.
- D. Blackwell. Minimax and irreducible matrices. *Journal of Mathematical Analysis and Applications*, 3:37–39, 1961.
- S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.
- R. L. Burden, J. Little, and J. D. Faires. *Numerical Analysis*. Brooks/Cole, Cengage Learning, Boston, MA, ninth edition, 2011.
- V. Chandrasekaran and P. Shah. Relative entropy optimization and its applications. *Mathematical Programming, Series A*, 161:1–32, 2017.
- C. Daskalakis, R. Frongillo, C. H. Papadimitriou, G. Pierrakos, and G. Valiant. On learning algorithms for Nash equilibria. In *Proc. 3rd International Symposium on Algorithmic Game Theory (SAGT 2010)*, pages 114–125, 2010.
- L. Flokas, E. V. Vlatakis-Gkaragkounis, T. Lianas, P. Mertikopoulos, and G. Piliouras. No-regret learning and mixed Nash equilibria: They do not mix. In *Proc. 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.
- Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proc. Ninth Annual Conference on Computational Learning Theory*, 1996.
- 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(1):119–139, 1997.

- Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.
- G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, third edition, 1996.
- N. J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal in Matrix Analysis and Its Applications*, 26:1179–1193, 2005.
- R. Kleinberg, G. Piliouras, and E. Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proc. 41st Annual ACM Symposium on the Theory of Computing (STOC '09)*, pages 533–542, 2009.
- W. Krichene, B. Drighes, and A. M. Bayen. Online learning of Nash equilibria in congestion games. *SIAM Journal on Control and Optimization*, 53:1056–1081, 2015.
- N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.
- P. Mertikopoulos, C. Papadimitriou, and G. Piliouras. Cycles in adversarial regularized learning. In *Proc. Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2018.
- J. Milonis, C. Papadimitriou, G. Piliouras, and K. Spendlove. Nash, Conley, and computation: Impossibility and incompleteness in game dynamics. arXiv eprint 2203.14129 (cs.GT), 2022.
- J. C. Miller, G. Rae, and F. Schaefer. Modifications of Kleinberg’s HITS algorithm using matrix exponentiation and web log records. In *Proc. SIGIR’01*, 2001.
- C. Moler and C. van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20(4):801–836, 1978.
- J. F. Nash. Non-cooperative games. *The Annals of Mathematics, Second Series*, 54(2):286–295, Sept. 1951.
- Y. Nesterov and A. Nemirovski. *Interior-point polynomial algorithms in convex programming*. Society of Industrial and Applied Mathematics, Philadelphia, 1994.
- M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), 2006.
- M. Pelillo and A. Torsello. Payoff-monotonic game dynamics and the maximum clique problem. *Neural Computation*, 18:1215–1258, 2006.
- R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. The MIT Press, Cambridge, Massachusetts and London, England, 2012.
- P. Taylor and L. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 16:76–83, 1978.
- S. Vigna. Spectral ranking. *Network Science*, 4(4):433–445, 2016.
- J. W. Weibull. *Evolutionary Game Theory*. MIT Press, 1995.
- A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *PNAS*, 113(47), Nov. 2016.