Online Learning and Its Applications in Games

Le Cong Dinh

University of Southampton

August 20, 2022

Table of Contents

1. Online Learning
2. Two-player Zero-Sum game
3. Last Round Convergence against No-regret Adversaries
4. Application and Discussion
5. Online Learning with Large-size games
6. Extension to Markov Decision Processes

Why Online Learning?

When our actions affect other people and their actions affect our objectives, we need to consider their incentives, and choose our actions in anticipation of theirs. This increase in complexity also occurs in situations involving multiple decision-making machines (e.g., self-driving cars), automated systems (e.g., algorithmic stock trading), or living organisms (e.g., groups of cells).

General Online Learning Setting

Online Convex Learning

Shalev-Shwartz et al. [2012]

```
Input: A convex set S
For $t = 1, 2, \ldots$
    predict a vector $\mathbf{w}_t \in S$
    receive a convex loss function $f_t$: $S \to \mathbb{R}$
    suffer loss $f_t(\mathbf{w}_t)$
```

The goal of the player is to:

$$\min_{\mathbf{w}} \sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

This minimization is impossible to achieve in adversary setting

No-regret Algorithms

$$Regret_T(\mathbf{u}) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u})$$

If we compare with the best-fixed strategy in the hindsight:

$$Regret_T(S) = \max_{\mathbf{u}\in S} Regret_T(\mathbf{u})$$

Definition 1 (Cesa-Bianchi and Lugosi [2006])

Let $f_1, f_2, \ldots$ be a sequence of loss function played by the environment. An algorithm of the player that generates a sequence of strategies $\mathbf{w}_1, \mathbf{w}_2, \ldots$ is called a *no-regret* algorithm if we have

$$\lim_{T\to\infty} \frac{Regret_T(S)}{T} = 0$$

No-dynamic regret algorithm

$$DynamicRegret_T(S) = \max_{\mathbf{u}_1,\mathbf{u}_2,\ldots\in S} \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

Definition 2 (Dinh et al. [2021b])

Let $f_1, f_2, \ldots$ be a sequence of loss functions played by the environment. An algorithm of the player that generates a sequence of strategies $\mathbf{w}_1, \mathbf{w}_2, \ldots$ is called a *no-dynamic-regret* algorithm if we have

$$\lim_{T\to\infty} \frac{InstantRegret_T(S)}{T} = 0$$

Online Learning: Follow the Leader

The most natural learning rule is to use the strategy that has minimal loss on all past rounds:

Definition 3

The agent is said to play the Follow the Leader (FTL) with σ-strongly convex regularizer: $F(\boldsymbol{x})$ if the agent updates the strategy as follows:

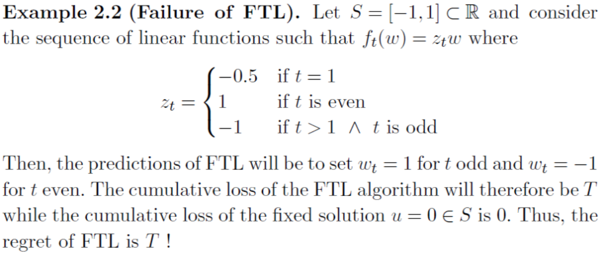
$$\boldsymbol{w}_t = \operatorname*{argmin}_{\boldsymbol{x}\in S} G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_t(\boldsymbol{w}).$$

Failure of FTL

Against a specific loss function (e.g., Quadratic loss function: $f_t(\boldsymbol{w}) = \|\boldsymbol{w} - \boldsymbol{z}_t\|^2$), FTL is a no-regret algorithm with the regret bound:

$$O(log(T)).$$

However, in general cases, FTL **is not** a no-regret algorithm.



Example 2.2 (Failure of FTL). Let $S = [-1, 1] \subset \mathbb{R}$ and consider the sequence of linear functions such that $f_t(w) = z_t w$ where

$$z_t = \begin{cases} -0.5 & \text{if } t = 1 \\ 1 & \text{if } t \text{ is even} \\ -1 & \text{if } t > 1 \ \wedge \ t \text{ is odd} \end{cases}$$

Then, the predictions of FTL will be to set $w_t = 1$ for t odd and $w_t = -1$ for t even. The cumulative loss of the FTL algorithm will therefore be T while the cumulative loss of the fixed solution $u = 0 \in S$ is 0. Thus, the regret of FTL is T !

Figure: Failure of FTL in linear loss function (Shalev-Shwartz et al. [2012])

Online Learning algorithm with No-regret Properties

We now consider general form of no-regret algorithms, namely Follow the Regularized Leader (e.g., see Abernethy et al. [2008]).

Definition 4

The agent is said to play the FTRL with σ-strongly convex regularizer: $F(\boldsymbol{x})$ if the agent updates the strategy as follows:

$$\boldsymbol{w}_t = \operatorname*{argmin}_{\boldsymbol{x} \in S} G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_i(\boldsymbol{w}) + \frac{1}{\mu} F(\boldsymbol{w}).$$

Regret Analysis of FTRL

Theorem 5

Let the agent follows FTRL with the sequence of linear loss function $f_t(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{z}_t \rangle$ for all t, $S = \mathcal{R}^d$ and the regularizer $R(\boldsymbol{w}) = \frac{1}{2\eta}\|\boldsymbol{w}\|^2$. Then for a set $U = \{\boldsymbol{u} : \|\boldsymbol{u}\| \leq B\}$, we have:

$$Regret_T(U) \leq O(\sqrt{T}).$$

Under mild conditions (i.e., the loss functions are Lipschitz), the choice of regularizer and action space S can be relaxed.

FTRL covers a large set of well-known no-regret algorithms. For instance, In the case of Euclidean regularizer, the FTRL becomes the famous Online Mirror Descent with lazy projection (e.g. see Shalev-Shwartz et al. [2012]). If the negative entropy function is used as the regularizer, then FTRL results in a fixed step-size Multiplicative Weight Update (MWU). MWU (Freund and Schapire [1999]) is a very important no-regret algorithm that has been extensively studied in the literature.

Multiplicative Weight Update

Appling FTRL with the simplex action space $S = \Delta_n$ and negative entropy function regularizer:

$$R(\boldsymbol{w}) = \frac{1}{\eta} \sum_i w(i) \log(w(i)),$$

result in the Multiplicative Weight Update Algorithm: (Freund and Schapire [1999])

$$w_{t+1}(i) = w_t(i) \frac{e^{-\mu_t e_i^T \boldsymbol{z}_t}}{Z_t} \quad \text{Where } Z_t \text{ is a normalization factor:}$$

$$Z_t = \sum_{i=1}^{n} w_t(i) e^{-\mu_t e_i^T \boldsymbol{z}_t}, \quad \mu_t \in [0,1) \quad \text{is a parameter of the algorithm}$$

and e_i is the unit-vector with value 1 at the i element.

2. Two-player zero-sum game

This game is described by a matrix $A_{n\times m}$ with entries in $[0,1]$.
The rows of A represent the “pure” strategies of the row player and the columns of A represent the “pure” strategies of the column player.

2. Two-player zero-sum game

This game is described by a matrix $A_{n\times m}$ with entries in $[0, 1]$.
The rows of A represent the "pure" strategies of the row player and the columns of A represent the "pure" strategies of the column player.
If the row player chooses a mixed strategy $x \in \Delta_n$ and the column player chooses a mixed strategy $y \in \Delta_m$, then the payoff the row player receives will be $-x^T Ay$ and the column player payoff is $x^T Ay$.

$$\text{Row player: } \min \; x^T Ay,$$
$$\text{Column player: } \max \; x^T Ay.$$

Online Learning Application: Finding Nash Equilibrium in two-player zero-sum games

The John von Neumann's minimax theorem (Neumann [1928]) :

$$\max_{\boldsymbol{y}\in\Delta_m}\min_{\boldsymbol{x}\in\Delta_n}\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y} \quad = \quad \min_{\boldsymbol{x}\in\Delta_n}\max_{\boldsymbol{y}\in\Delta_m}\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y} \quad = \quad v, \tag{1}$$

for some $v \in \mathbb{R}$. We call a point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying the minimax theorem inequation 1 *the minimax equilibrium of the game*.
Throughout the paper, we use the notation $f(\boldsymbol{x}) \; := \; \max_{\boldsymbol{y}\in\Delta_m}\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y}$. Since $\boldsymbol{A}$ is a non-zero matrix with entries in $[0,1]$, we have $f(\boldsymbol{x}) \geq 0$. Note that $(\boldsymbol{x}_l, \boldsymbol{y}^*)$ which satisfy $f(\boldsymbol{x}_l) - v \leq \epsilon$ are ϵ-Nash equilibria(i.e., $\max_{\boldsymbol{y}\in\Delta_m}\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y} - \boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y} \leq \epsilon$ and $\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y} - \min_{\boldsymbol{x}\in\Delta_n}\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y} \leq \epsilon$) and $\epsilon = 0$ implies $\boldsymbol{x}_l$ is the Nash equilibrium of the row player.

Theorem 6

In two-player zero-sum games, if both players follow no-regret algorithms, then the average strategy of both players convergence to the Nash Equilibrium of the game.

Proof of average convergence to NE

Sketch of proof.

Since both player follows a no-regret algorithm, we then have:

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t - \min_{\boldsymbol{x}\in\Delta_n} \sum_{t=1}^{T} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}_t = O(\sqrt{T}) \implies \min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\bar{\boldsymbol{y}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t - O(\frac{1}{\sqrt{T}}).$$

Similarly, we have:

$$\max_{\boldsymbol{y}\in\Delta_m} \bar{\boldsymbol{x}}^\top \boldsymbol{A}\boldsymbol{y} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t + O(\frac{1}{\sqrt{T}})$$

Thus we have:

$$\bar{\boldsymbol{x}}^\top \boldsymbol{A}\bar{\boldsymbol{y}} \geq \min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\bar{\boldsymbol{y}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t - O(\frac{1}{\sqrt{T}}) = \max_{\boldsymbol{y}\in\Delta_m} \bar{\boldsymbol{x}}^\top \boldsymbol{A}\boldsymbol{y} - O(\frac{1}{\sqrt{T}})$$

$$\bar{\boldsymbol{x}}^\top \boldsymbol{A}\bar{\boldsymbol{y}} \leq \max_{\boldsymbol{y}\in\Delta_m} \bar{\boldsymbol{x}}^\top \boldsymbol{A}\boldsymbol{y} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t + O(\frac{1}{\sqrt{T}}) = \min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\bar{\boldsymbol{y}} + O(\frac{1}{\sqrt{T}}).$$

□

Convergence to Coarse Correlated Equilibrium in general-sum game

If follow the distribution σ over all possible combinations is no worse than always following some fixed strategy, then σ is a coarse correlated Equilibrium i.e.,:

$$\forall i, s_i' \quad \mathbb{E}_{s\sim\sigma} l(s) \leq \mathbb{E}_{s\sim\sigma} l(s_i', s_{-i})$$

Theorem 7

In a general-sum game with a finite number of players and a finite number of strategies for each player. If each player follows a no-regret algorithm, then the empirical distribution of the dynamic converges to a coarse correlated equilibrium.

Non-Last Round Convergence Properties

We consider the simple matching pennies game. The pay-off of the game is:

	head	tail
head	(1,-1)	(-1,1)
tail	(-1,1)	(1,-1)

Table: Payoff in Matching Pennies game (Bailey and Piliouras [2018])

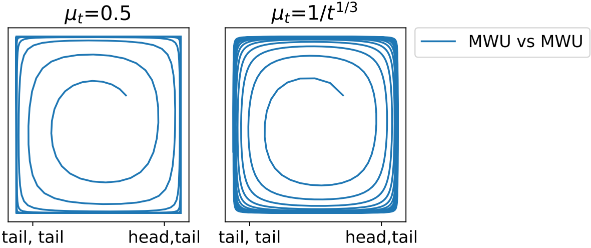


Figure: MWU vs MWU after 2500 iterations with different step sizes

4. Asymmetric setting

The goals of the column player are:

1. no-dynamic-regret
2. stable strategies (i.e the strategy of the column player converges)

4. Asymmetric setting

The goals of the column player are:

1. no-dynamic-regret
2. stable strategies (i.e the strategy of the column player converges)

In order to achieve that, we need the following assumptions:

1. the row player follows a no-regret type algorithms.
2. the column player can estimate his minimax equilibrium strategy.

4. Asymmetric setting

The goals of the column player are:

1. no-dynamic-regret
2. stable strategies (i.e the strategy of the column player converges)

In order to achieve that, we need the following assumptions:

1. the row player follows a no-regret type algorithms.
2. the column player can estimate his minimax equilibrium strategy.

4. Asymmetric setting

The goals of the column player are:

1. no-dynamic-regret
2. stable strategies (i.e the strategy of the column player converges)

In order to achieve that, we need the following assumptions:

1. the row player follows a no-regret type algorithms.
2. the column player can estimate his minimax equilibrium strategy.

Assumption (2) can arise in many applications:

1. asymmetric games where the column player knows the matrix A of the game.

4. Asymmetric setting

The goals of the column player are:

1. no-dynamic-regret
2. stable strategies (i.e the strategy of the column player converges)

In order to achieve that, we need the following assumptions:

1. the row player follows a no-regret type algorithms.
2. the column player can estimate his minimax equilibrium strategy.

Assumption (2) can arise in many applications:

1. asymmetric games where the column player knows the matrix A of the game.
2. the column player can intentionally estimate his minimax equilibirium of game while playing.

Convergence of the row player

Lemma 8

Suppose that the row player follows a common no-regret algorithm such as MWU, OMD, FTRL, LMWU or OMWU. Then, the column player cannot achieve last round convergence and the no-regret property if the row player's strategy does not converge to a minimax equilibrium of the game.

The LRCA Algorithm

Algorithm 1: *L*ast *R*ound *C*onvergence in *A*symmetric algorithm (LRCA)

```
Input: Current iteration $t$, past feedback $x_{t-1}^{\top}A$ of the row player
Output: Strategy $y_t$ for the column player
if $t = 2k-1,\ k \in \mathbb{N}$ then
    $y_t = y^*$
end
if $t = 2k,\ k \in \mathbb{N}$ then
    $e_t := \mathrm{argmax}_{e \in \{e_1, e_2, \dots e_m\}} x_{t-1}^{\top}Ae; \quad f(x_{t-1}) := \max_{y \in \Delta_m} x_{t-1}^{\top}Ay$
    $\alpha_t := \frac{f(x_{t-1}) - v}{\max\ \left(\frac{n}{4}, 2\right)}$
    $y_t := (1-\alpha_t)y^* + \alpha_t e_t$
end
```

MWU vs LRCA

Lemma 9

Assume that the row player follows the MWU algorithm with a non-increasing step size μ_t such that there exists $t' \in \mathbb{N}$ with $\mu_{t'} \leq 1$. If the column player follows LRCA then

$$RE\left(x^*||x_{2k-1}\right) - RE\left(x^*||x_{2k+1}\right) \geq \frac{1}{2}\mu_{2k}\alpha_{2k}(f(x_{2k-1}) - v) \ \ \forall k \in \mathbb{N}: \ 2k \geq t',$$

MWU vs LRCA

Lemma 9

Assume that the row player follows the MWU algorithm with a non-increasing step size μ_t *such that there exists* $t' \in \mathbb{N}$ *with* $\mu_{t'} \leq 1$*. If the column player follows LRCA then*

$$RE\left(x^*||x_{2k-1}\right) - RE\left(x^*||x_{2k+1}\right) \geq \frac{1}{2}\mu_{2k}\alpha_{2k}(f(x_{2k-1}) - v) \ \forall k \in \mathbb{N}: \ 2k \geq t',$$

Theorem 10

Let A *be an* $n \times m$ *non-zero matrix with entries in* $[0,1]$*. Assume that the row player follows the MWU algorithm with a non-increasing step size* μ_t *such that* $\lim_{T\to\infty}\sum_{t=1}^{T}\mu_t = \infty$ *and there exists* $t' \in \mathbb{N}$ *with* $\mu_{t'} \leq 1$*. If the column player plays LRCA then there exists a minimax equilibrium* $\bar{x}^*$*, such that* $\lim_{t\to\infty} RE(\bar{x}^*||x_t) = 0$ *and thus* $lim_{t\to\infty} x_t = \bar{x}^*$ *almost everywhere.*

MWU vs LRCA: coutinue

Lemma 11

In the case of constant learning rate $\mu_t = \mu$, we have the complexity of the algorithm in order to achieve $f(x) - v \leq \epsilon$ is

$$\frac{4\log(n)/\mu}{\epsilon^2}.$$

FTRL vs LRCA

Theorem 12

Assume that the row player follows the FTRL with σ-strongly convex regularizer: F(x) with fixed step size μ. Then if the column player follows the Algorithm 1 (LRCA), there will be last round convergence to the minimax equilibrium.

FTRL vs LRCA

Theorem 12

Assume that the row player follows the FTRL with σ-strongly convex regularizer: F(x) with fixed step size μ. Then if the column player follows the Algorithm 1 (LRCA), there will be last round convergence to the minimax equilibrium.

Lemma 13

the FTRL with negative entropy regularizer becomes the MWU with constant step size μ. However, when μ varies in each update, then the two algorithms can be significantly different and thus the analysis in Theorem 10 is necessary. The complexity of the algorithm in order to achieve $f(x) - v \leq \epsilon$ is

$$\mathcal{O}(\frac{2n^2\mu}{\epsilon^2}).$$

General No-Regret Algorithms vs LRCA

Definition 14

A no-regret algorithm is *stable* if $\forall t : \mathbf{y}_t = \mathbf{y}^* \implies \mathbf{x}_{t+1} = \mathbf{x}_t$.

Theorem 15

Assume that the row player follows a stable no-regret algorithm and n is the dimension of the row player's strategy. Then, by following LRCA, for any $\epsilon > 0$, there exists $l \in \mathbb{N}$ such that $\frac{Regret_l}{l} = \mathcal{O}(\frac{\epsilon^2}{n})$ and $f(\mathbf{x}_l) - v \leq \epsilon$.

Note that $(\mathbf{x}_l, \mathbf{y}^*)$ which satisfy $f(\mathbf{x}_l) - v \leq \epsilon$ are ϵ-Nash equilibria and $\epsilon = 0$ implies $\mathbf{x}_l$ is the Nash equilibrium of the row player. For no-regret algorithms with optimal regret bound $Regret_l = O(\sqrt{l})$, following Theorem 15, the row player will reach an ϵ-Nash equilibrium in at most $\mathcal{O}(\frac{n^2}{\epsilon^4})$ rounds.

No-dynamic-Regret LRCA

Theorem 16

Assume that the row player follows the above-mentioned no-regret type algorithms: MWU, FTRL. If there exists a fully mixed minimax strategy for the row player, then by following LRCA, the column player will achieve the no-dynamic-regret property with the instant-regret satisfying $R_T \leq IR_T = \mathcal{O}\left(\sqrt{n\log(n)}T^{3/4}\right)$. *Furthermore, in the case the row player uses a constant learning rate, we have* $IR_T = \mathcal{O}\left(\sqrt{n\log(n)}T^{1/2}\right)$.

4. Application and Discussion

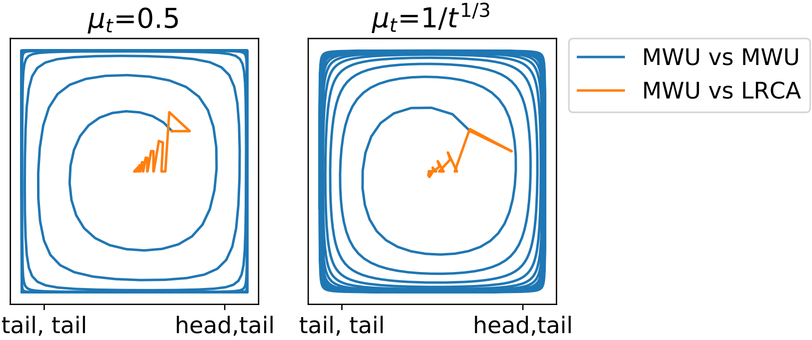


Figure: MWU vs LRCA after 2500 iterations with different step sizes

Application: System Design

The column player is also a designer of the game.
The goal of the column player is to guide his opponent to pick a mixed strategy which is favourable for the system designer.
The column player needs to:

1. design an appropriate payoff matrix A whose unique minimax solution contains the desired mixed strategy of the row player
2. strategically interact with the row player during a sequence of plays in order to guide his opponent to converge to that desired behaviour

See Dinh et al. [2020] for more detail.

Last round convergence in symmetric setting: Optimistic Multiplicative Weight Update

Motivation:

$$\boldsymbol{w}_t = \operatorname*{argmin}_{\boldsymbol{x} \in S} G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_i(\boldsymbol{w}) + \frac{1}{\mu} F(\boldsymbol{w}).$$

Last round convergence in symmetric setting: Optimistic Multiplicative Weight Update

Motivation:

$$\boldsymbol{w}_t = \operatorname*{argmin}_{\boldsymbol{x}\in S} G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_i(\boldsymbol{w}) + \frac{1}{\mu}F(\boldsymbol{w}).$$

$$\boldsymbol{w}_t = \operatorname*{argmin}_{\boldsymbol{x}\in S} G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_i(\boldsymbol{w}) + f_t(\boldsymbol{w}) + \frac{1}{\mu}F(\boldsymbol{w}).$$

Last round convergence in symmetric setting: Optimistic Multiplicative Weight Update

Motivation:

$$\boldsymbol{w}_t = \underset{\boldsymbol{x}\in S}{\operatorname{argmin}}\, G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_i(\boldsymbol{w}) + \frac{1}{\mu} F(\boldsymbol{w}).$$

$$\boldsymbol{w}_t = \underset{\boldsymbol{x}\in S}{\operatorname{argmin}}\, G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_i(\boldsymbol{w}) + f_t(\boldsymbol{w}) + \frac{1}{\mu} F(\boldsymbol{w}).$$

$$\boldsymbol{w}_t = \underset{\boldsymbol{x}\in S}{\operatorname{argmin}}\, G_t(\boldsymbol{w}) = \sum_{i=1}^{t-1} f_i(\boldsymbol{w}) + f_{t-1}(\boldsymbol{w}) + \frac{1}{\mu} F(\boldsymbol{w}).$$

Last round convergence in symmetric setting: OMWU

Algorithm 2: Optimistic Multiplicative Weights Update

Input: learning rate $\eta > 0$, exploiting rate $\alpha > 0$,
$\boldsymbol{f}_1 = \boldsymbol{f}_2 = [1/n, \ldots, 1/n]$.
Output: Next update

$$\boldsymbol{f}_{t+1}(i) = \frac{\boldsymbol{f}_t(i) e^{\eta(2{e_i}^\top \boldsymbol{A}\boldsymbol{y}_t - {e_i}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})}}{\sum_j \boldsymbol{f}_t(j) e^{\eta(2{e_j}^\top \boldsymbol{A}\boldsymbol{y}_t - {e_j}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})}},$$

e_i denotes the unit-vector with weight of 1 at i-component.

Theorem 17 (Daskalakis and Panageas [2018])

In a two-player zero-sum game with unique Nash equilibrium, if both players follow OMWU with sufficiently small learning rate η, then the dynamic converges last round to the Nash Equilibrium of the game. Furthermore, OMWU can achieve a near-optimal convergence rate (up to logarithm factor) to CCE in general-sum games.

Online Learning with Large-size games

Understanding games with large action spaces is a critical topic in a variety of fields from economics to operations research and artificial intelligence. Conventional no-regret algorithms require the computational complexity to depend on the size of the game (i.e., size of game matrix $\boldsymbol{A}$), thus when the game size is large, these algorithms will surrender.

Online Learning with Large-size games

Understanding games with large action spaces is a critical topic in a variety of fields from economics to operations research and artificial intelligence. Conventional no-regret algorithms require the computational complexity to depend on the size of the game (i.e., size of game matrix $\boldsymbol{A}$), thus when the game size is large, these algorithms will surrender. We solve this problem by proposing a new algorithm: online single oracle, a combination of the double oracle method and conventional no-regret algorithms MWU.

Online Single Oracle

```
Input: Player's pure strategy set $\Pi$
Init. effective strategies set: $\Pi_0 = \Pi_1 = \{\boldsymbol{a}^j\}, \boldsymbol{a}^j \in \Pi$
for $t = 1$ to T do
    if  $\Pi_t = \Pi_{t-1}$ then
        Compute $\boldsymbol{\pi}_t$ by the MWU
    else if $\Pi_t \neq \Pi_{t-1}$ then
        Start a new time window $T_{i+1}$ and
        Reset $\boldsymbol{\pi}_t = \left[1/|\Pi_t|, \dots, 1/|\Pi_t|\right]$,   $\bar{\boldsymbol{l}} = \mathbf{0}$
    end if
    Observe $\boldsymbol{l}_t$ and update the average loss in $T_i$: $\bar{\boldsymbol{l}} = \sum_{t \in T_i} \boldsymbol{l}_t / |T_i|$
    Calculate the best-response: $\boldsymbol{a}_t = \arg\min_{\boldsymbol{\pi} \in \Pi} \langle \boldsymbol{\pi}, \bar{\boldsymbol{l}} \rangle$
    Update the set of strategies: $\Pi_{t+1} = \Pi_t \cup \{\boldsymbol{a}_t\}$
end for
Output: $\boldsymbol{\pi}_T$, $\Pi_T$
```

Regret Bound of OSO (Dinh et al. [2021c])

Theorem 18 (Regret Bound of OSO)

Let $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_T$ be a sequence of loss vectors played by an adversary, and $\langle \cdot, \cdot \rangle$ be the dot product, OSO is a no-regret algorithm with

$$\frac{1}{T}\Big(\sum_{t=1}^{T} \langle \boldsymbol{\pi}_t, \boldsymbol{l}_t \rangle - \min_{\boldsymbol{\pi} \in \Pi} \sum_{t=1}^{T} \langle \boldsymbol{\pi}, \boldsymbol{l}_t \rangle\Big) \leq \frac{\sqrt{k \log(k)}}{\sqrt{2T}},$$

where $k = |\Pi_T|$ is the size of the effective strategy set in the final time window.

Performance of ODO

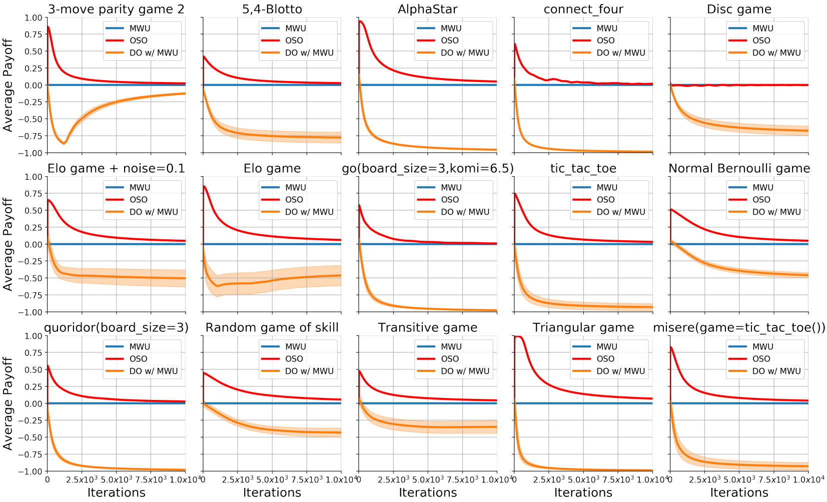


Figure: Average payoff against MWU adversary

Performance of ODO

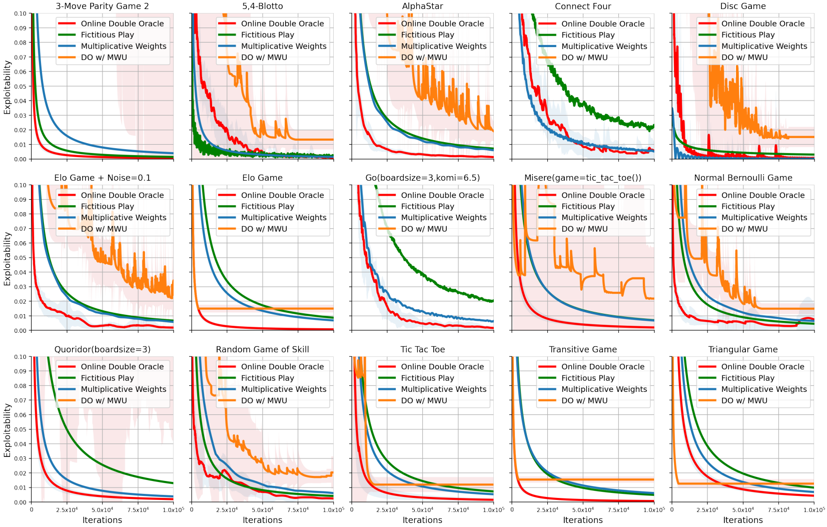


Figure: Convergence to Nash Equilibrium

Performance of ODO

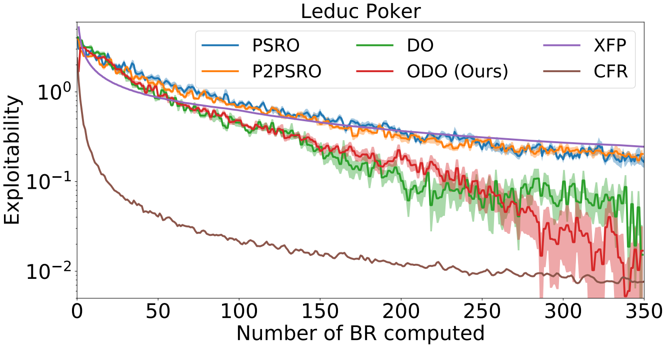


Figure: Convergence to Nash Equilibrium in large size normal form game with 3^{396} pure strategies

Extension to Online Markov Decision Processes

We consider OMDPs where at each round $t \in \mathbb{N}$, an adversary can choose the loss function $\boldsymbol{l}_t$ based on the agent's history $\{\pi_1, \pi_2, \ldots, \pi_{t-1}\}$. At time t, given state $x_t \in S$, the agent chooses an action $a_t \in A$, then the agent moves to a new random state x_{t+1} which is determined by the fixed transition model $P(x_{t+1}|x_t, a_t)$. Simultaneously, the agent receives an immediate loss $\boldsymbol{l}_t(x_t, a_t)$, in which the loss function $\boldsymbol{l}_t : S \times A \to \mathbb{R}$ is bounded in $[0, 1]$ and chosen by the adversary from a simplex $\Delta_L := \{\boldsymbol{l} \in \mathbb{R}^{|S||A|} | \boldsymbol{l} = \sum_{i=1}^{L} x_i \boldsymbol{l}_i,\ \sum_{i=1}^{L} x_i = 1,\ x_i \geq 0\ \forall i\}$ where $\{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_L\}$ are the loss vectors of the adversary.
The goal of the agent is to have minimum policy regret with respect to the best fixed policy in hindsight:

$$R_T(\pi) = \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi_t}(X_t, A_t)\right] - \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi}(X_t^{\pi}, A_t^{\pi})\right], \qquad (2)$$

where $\boldsymbol{l}_t^{\pi_t}$ denotes the loss function at time t while the agent follows $\pi_1, \ldots, \pi_T$ and $\boldsymbol{l}_t^{\pi}$ is the adaptive loss function against the fixed policy π of the agent.

MDP-Online Oracle Expert

Algorithm 3: MDP-Online Oracle Expert(Dinh et al. [2021a])

```
Input: Sets $A_0^1, \dots A_0^S$ of effective strategy set in each state
for $t = 1$ to $\infty$ do
    $\pi_t = BR(\bar{\boldsymbol{l}})$
    if $\pi_t(s,.) \in A_{t-1}^s$ for all $s$ then
        $A_t^s = A_{t-1}^s$ for all $s$
        Using the expert algorithm $B_s$ with effective strategy set $A_t^s$ and the feedback $Q_{\pi_t,\boldsymbol{l}_t}(s,.)$
    else if there exists $\pi_t(s,.) \notin A_{t-1}^s$ then
        $A_t^s = A_{t-1}^s \cup \pi_t(s,.)$   if $\pi_t(s,.) \notin A_{t-1}^s$
        $A_t^s = A_{t-1}^s \cup a$   if $\pi_t(s,.) \in A_{t-1}^s$ where a is randomly selected from the set $A/A_{t-1}^s$.
        Reset the expert algorithm $B_s$ with effective strategy set $A_t^s$ and the feedback $Q_{\pi_t,\boldsymbol{l}_t}(s,.)$
    end if
    $\bar{\boldsymbol{l}} = \sum_{i=\bar{T}_i}^{T} \boldsymbol{l}_t$
end for
```

Regret Bound of MDP-OOE

Theorem 19 (Dinh et al. [2021a])

Suppose the agent uses MDP-OOE in our online MDPs setting, then the policy regret can be bounded by:

$$R_T(\pi) = \mathcal{O}(\sqrt{\tau^2 Tk \log(k)} + \sqrt{T \log(L)}).$$

Discussion

Further questions?

Jacob Abernethy, Elad E Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory, COLT 2008*, pages 263–273, 2008.

James P Bailey and Georgios Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 321–338, 2018.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. *arXiv preprint arXiv:1807.04252*, 2018.

Le Cong Dinh, Nick Bishop, and Long Tran-Thanh. Exploiting no-regret algorithms in system design. *arXiv preprint arXiv:2007.11172*, 2020.

Le Cong Dinh, David Henry Mguni, Long Tran-Thanh, Jun Wang, and Yaodong Yang. Online markov decision processes with non-oblivious strategic adversary. *arXiv preprint arXiv:2110.03604*, 2021a.

Le Cong Dinh, Tri-Dung Nguyen, Alain B Zemhoho, Long Tran-Thanh, et al. Last round convergence and no-dynamic regret in asymmetric repeated games. In *Algorithmic Learning Theory*, pages 553–577. PMLR, 2021b.

Le Cong Dinh, Yaodong Yang, Zheng Tian, Nicolas Perez Nieves, Oliver Slumbers, David Henry Mguni, Haitham Bou Ammar, and Jun Wang. Online double oracle. *arXiv preprint arXiv:2103.07780*, 2021c.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

J v Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.