Online Optimization, Learning, and Games (O2LG)

Lesson 4: Learning algorithms

Vinh Thanh Ho*, Panayotis Mertikopoulos

*Faculté des Sciences et Techniques
Université de Limoges

vinh-thanh.ho@unilim.fr

Master of Applied Mathematics

Table of Contents

Learning scheme in discrete time with different feedback types

Input: a finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$.

Repeat for each epoch $n = 1, 2, \ldots$, for all players $i \in \mathcal{N}$,

- Choose mixed strategy $x_{i,n} \in \mathcal{X}_i := \Delta(\mathcal{A}_i)$.
- Choose action $a_{i,n} \sim x_{i,n}$.
- Observe mixed payoff vector $v_i(x_n)$ or pure payoff vector $v_i(a_n)$ or realized payoff $u_i(a_n)$.

Until end

The feedback process

Types of feedback

From best to worst (more to less info):

- Mixed payoff vectors: $v_i(x_n)$ # deterministic vector feedback
- Pure payoff vectors: $v_i(a_n)$ # stochastic vector feedback
- Bandit / Payoff-based: $u_i(a_n)$ # stochastic scalar feedback

Features:

- Vector (mixed / pure payoff vectors) versus Scalar (bandit).
- Deterministic (mixed payoff vectors) versus Stochastic (pure payoff vectors, bandit).

☞ Randomness defined relative to history of play $\mathcal{F}_n := \mathcal{F}(x_1, \ldots, x_n)$.

☞ Other feedback models also possible (noisy / delayed observations,...).

From payoffs to payoff vectors

A closer look on payoff-based (bandit) feedback:

- Draw action $a_{i,n} \in \mathcal{A}_i$ according to mixed strategy $x_{i,n} \in \mathcal{X}_i$.
- Receive payoff $u_i(a_n) = u_i(a_{i,n}, a_{-i,n})$.

How to estimate the payoff $v_{i,a_i}(a_n) = u_i(a_i, a_{-i,n})$ of **another** action $a_i \neq a_{i,n}$?

Importance-weighted estimator

Definition 1 (Importance-weighted estimator)

Fix a payoff vector $v \in \mathbb{R}^{\mathcal{A}}$ and a probability distribution P on $\mathcal{A}$. Then, for a given $a \in \mathcal{A}$, the importance-weighted estimator of v_a relative to P is the **random variable**

$$\hat{v}_a = \frac{v_a}{P_a}\mathbb{1}_a = \begin{cases} \dfrac{v_a}{P_a} & \text{if } a \text{ is drawn,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

An alternative called *loss-based importance-weighted estimator* (here loss $= 1-$ payoff (reward)):

$$\hat{v}_a = 1 - \frac{1 - v_a}{P_a}\mathbb{1}_a.$$

☞ Although the two estimators seem quite similar, it should be noted that the first estimator takes values in $[0, \infty)$ while the second takes values in $(-\infty, 1]$.

Properties of Importance-weighted estimator

Statistical properties of (1) in IWE

- Unbiasedness: $\mathbb{E}[\hat{v}_a] = v_a$.
- Second moment: $\mathbb{E}[\hat{v}_a^2] = \dfrac{v_a^2}{P_a}$.

The oracle model

Definition 2 (Oracle feedback)

A stochastic first-order oracle of $v(x_n)$ is a random vector of the form

$$\hat{v}_n = v(x_n) + U_n + b_n$$

where U_n is zero-mean and $b_n = \mathbb{E}[\hat{v}_n | \mathcal{F}_n] - v(x_n)$ is the bias of $\hat{v}_n$.

Examples

- Mixed payoff vectors: $\hat{v}_{i,n} = v_i(x_{i,n}, x_{-i,n})$.
- Pure payoff vectors: $\hat{v}_{i,n} = v_i(a_{i,n}, a_{-i,n})$.
- Payoff-based: $\hat{v}_{i,n} = \dfrac{u_i(a_{i,n}, a_{-i,n})}{\mathbb{P}(a_{i,n} = a_i)} e_{a_i,n}$.

Table of Contents

The exponential weights scheme

Basic idea:

- Score actions by aggregating payoff vector estimates (provided by oracle or otherwise).
- Choose an action with probability exponentially proportional to its score.
- Rinse / repeat

Algorithm 1 Exponential weights in discrete time (ExpWeight)

Require: finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$; stochastic first-order oracle $\hat{v}$.
Initialize: $y_{i,1} \in \mathbb{R}^{\mathcal{A}_i}$, $i = 1, \ldots, N$ and step-sizes $\{\gamma_n\}$.
for all epoch $n = 1, 2, \ldots$, **for all players** $i \in \mathcal{N}$ **do**
  set $x_{i,n} \propto \exp(y_{i,n})$ ▷ mixed strategy
  play $a_{i,n} \sim x_{i,n}$ ▷ choose action
  get $\hat{v}_{i,n} \in \mathbb{R}^{\mathcal{A}_i}$ ▷ payoff model
  set $y_{i,n+1} \longleftarrow y_{i,n} + \gamma_n \hat{v}_{i,n}$ ▷ update scores
end for

Assumptions

Assumptions 1 (Oracle feedback)

The oracle feedback sequence $\hat{v}_n = v(x_n) + U_n + b_n$ has

- Bias: $\|b_n\| \leq B_n$.
- Variance: $\mathbb{E}[\|U_n\|^2 | \mathcal{F}_n] \leq \sigma_n^2$.
- Second moment: $\mathbb{E}[\|\hat{v}_n\|^2 | \mathcal{F}_n] \leq M_n^2$.

Variants

Variant scoring schemes

- Decreasing $\gamma_n \Rightarrow$ feedback enters the algorithm with decreasing weight. # step-size γ_n
- $\gamma_n = 1$, but $x_{i,n} \propto \exp(\eta_n y_{i,n}) \Rightarrow$ feedback enters the algorithm with the same weight. # learning rate η_n

Learning rate η:

- When the learning rate is large, we concentrate on the action with the largest estimated scores (i.e. cumulative payoffs) and the resulting algorithm exploits aggressively.
- For small learning rates, the action is more uniform, and the algorithm explores more frequently.
- There are many ways to tune the learning rate, including allowing it to vary with time.

Model 1: ExpWeight with mixed payoff vector feedback

If players observe mixed payoff vectors:

$$\hat{v}_{i,n} = v_i(x_{i,n}, x_{-i,n}).$$

Oracle features:

- Deterministic: no randomness!
- Bias: $B_n = 0$. # Why?
- Variance: $\sigma_n^2 = 0$.
- Second moment: $M_n^2 = \mathcal{O}(1)$.

☞ Also known as Multiplicative Weights Update (Arora et al. 2012).

Model 2: ExpWeight with pure payoff vector feedback

If players observe pure payoff vectors:

$$\hat{v}_{i,n} = v_i(a_{i,n}, a_{-i,n}).$$

Oracle features:

- Stochastic: random action selection.
- Bias: $B_n = 0$. # Why? Note $\mathcal{F}_n = x_n$ and $\mathbb{E}[v_i(a)|x_n] = \mathbb{E}_{a\sim x_n}[v_i(a)]$.
- Variance: $\sigma_n^2 = \mathcal{O}(1)$.
- Second moment: $M_n^2 = \mathcal{O}(1)$.

☞ Also known as Hedge. (Auer et al. 1995, Auer et al. 2002)

Model 3: ExpWeight with bandit feedback

If players observe realized payoffs only:

$$\hat{v}_{i,n} = \frac{u_i(a_{i,n}, a_{-i,n})}{\mathbb{P}(a_{i,n} = a_i)} e_{a_{i,n}}.$$

Oracle features:

- Stochastic: random action selection.
- Bias: $B_n = 0$.
- Variance: $\sigma_n^2 = \mathcal{O}(1/\min_i \min_{a_i} x_{i,a_i,n})$.
- Second moment: $M_n^2 = \mathcal{O}(1/\min_i \min_{a_i} x_{i,a_i,n})$.

☞ Also known as EXP3 (EXPonential weights algorithm for EXPloration and EXPloitation).

(Auer et al. 1995; Auer et al. 2002)

Model 4: ExpWeight with bandit feedback

If players observe realized payoffs only:

$$\hat{v}_{i,n} = \frac{u_i(a_{i,n}, a_{-i,n})}{\mathbb{P}(a_{i,n} = a_i)} \, e_{a_{i,n}}.$$

Oracle features:

- Stochastic: random action selection.
- Explicit exploration: draw $a_{i,n} \sim x_{i,n}$ with prob. $1 - \varepsilon_n$, otherwise uniformly.
- Bias: $B_n = \mathcal{O}(\varepsilon_n)$.
- Variance: $\sigma_n^2 = \mathcal{O}(1/\varepsilon_n^2)$.
- Second moment: $M_n^2 = \mathcal{O}(1/\varepsilon_n^2)$.

☞ Also known as EXP3 with Explicit Exploration. (Lattimore et al. 2020, Shalev-Shwartz 2012)

Implementation of ExpWeight algorithm

Task

1. Write the ExpWeight algorithm with mixed payoff vector feedback where
   - the initial action scores are set to 0,
   - the step-sizes γ_n are set to 1,
   - for each player i, x_i is considered as a probability distribution P on $\mathcal{A}_i$. Obviously, it is updated at each epoch n.
2. By using Python, implement this algorithm in the example of Prisoner's Dilemma in the previous lessons.
3. What does the sequence of play look like? Compare with the behaviour of replicator dynamics.

Table of Contents

Dynamics and rationality

Are game-theoretic solution concepts consistent with the players' dynamics?

- Do dominated strategies die out in the long run?
- Are Nash equilibria stationary?
- Are they stable? Are they attracting?
- Do the dynamics always converge?
- What other behaviors can we observe?
- ...

Dominated strategies

Suppose $a_i \in \mathcal{A}_i$ is dominated by $a_i' \in \mathcal{A}_i$.

- Consistent payoff gap: $v_{i,a_i}(x) \le v_{i,a_i'}(x) - c$, for some $c > 0$.
- Corresponding scores:

$$y_{i,a_i,n} = \sum_{k=1}^{n} \gamma_k [v_{i,a_i}(x_k) + b_{i,a_i,k} + U_{i,a_i,k}].$$

$$y_{i,a_i',n} = \sum_{k=1}^{n} \gamma_k [v_{i,a_i'}(x_k) + b_{i,a_i',k} + U_{i,a_i',k}].$$

- Difference in scores less clear: with $\beta_n = b_{i,a_i,n} - b_{i,a_i',n}$ and $\xi_n = U_{i,a_i,n} - U_{i,a_i',n}$,

$$y_{i,a_i,n} - y_{i,a_i',n} \le -c \sum_{k=1}^{n} \gamma_k + \sum_{k=1}^{n} \gamma_k \beta_k + \sum_{k=1}^{n} \gamma_k \xi_k.$$

The law of large numbers (LLN)

Strong law of large numbers

Let ξ_n, $n = 1, 2, \ldots$, be a sequence of i.i.d. random variables with $\mathbb{E}[\xi_n] = 0$ and $\mathbb{E}[\xi_n^2] < \infty$. Then the sample mean

$$\bar{\xi}_n = \frac{1}{n}\sum_{k=1}^{n} \xi_k \text{ converges to 0 with probability 1.}$$

Martingales and their limits

Apply law of large numbers to the noise term $\text{noise}_n := \sum_{k=1}^{n} \gamma_k \xi_k$?

✗ Increments are not i.i.d..

✗ Re-scaling by γ_n.

Definition 3 (Martingales)

A discrete-time martingale is a discrete-time stochastic process (i.e., a sequence of random variables) $\{S_n\}_{n=1,2,\dots}$ such that

- $\mathbb{E}[|S_n|] < \infty$ for all $n = 1, 2, \dots$
- $\mathbb{E}[S_{n+1} | S_n, \dots, S_1] = S_n$.

Intuition: The second condition means that the conditional expected value of the next observation, given all the past observations, is equal to the most recent observation.

Martingales and their limits

Strong law of large numbers for martingales (Hall et al. 1980)

Let $S_n = \sum_{k=1}^{n} \gamma_k \xi_k$, $n = 1, 2, \ldots$, be a martingale with $\mathbb{E}[\xi_n^2] < \infty$. Then

$$\frac{S_n}{\sum_{k=1}^{n} \gamma_k} \text{ converges to } 0 \text{ with probability } 1.$$

Back to dominated strategies

- Recall:

$$y_{i,a_i,n} - y_{i,a_i',n} \le -c\sum_{k=1}^{n}\gamma_k + \sum_{k=1}^{n}\gamma_k\beta_k + \sum_{k=1}^{n}\gamma_k\xi_k$$

where $\beta_n = b_{i,a_i,n} - b_{i,a_i',n}$ and $\xi_n = U_{i,a_i,n} - U_{i,a_i',n}$.

- By LLN, the drift term $\text{drift}_n := c\sum_{k=1}^{n}\gamma_k$ is dominant if the bias vanishes and the noise is "not too large".

- If the drift dominates, then

$$\frac{x_{i,a_i,n}}{x_{i,a_i',n}} = \exp(y_{i,a_i,n} - y_{i,a_i',n}) \to 0 \quad \text{when } n \to \infty.$$

Elimination of dominated strategies

Elimination of dominated strategies

If ExpWeight is run with $\sum_k \gamma_k = \infty$, $\sum_k \gamma_k B_k < \infty$, and $\sum_k \gamma_k^2 \sigma_k^2 < \infty$, then dominated strategies become extinct with probability 1.

Stochastic stability

Definition 4 (Stochastic stability)

We say that $x^* \in \mathcal{X}$ is stochastically stable under x_n if, for every confidence level $\delta > 0$ and for every neighborhood $\mathcal{U}$ of x^*, there exists a neighborhood $\mathcal{U}_1$ of x^* such that

$$\mathbb{P}(x_n \in \mathcal{U} \text{ for all } n = 1, 2, \ldots | x_1 \in \mathcal{U}_1) \geq 1 - \delta.$$

Intuition: If x_n starts close enough to x^*, it remains close enough with arbitrarily high probability.

Stochastic asymptotic stability

Definition 5 (Stochastic asymptotic stability)

Let $x_n \in \mathcal{X}$, $n = 1, 2, \ldots$, be a discrete-time stochastic process. We then say that $x^* \in \mathcal{X}$ is:

- Attracting if, for every confidence level $\delta > 0$, there exists a neighborhood $\mathcal{U}_1$ of x^* such that
$$\mathbb{P}(x_n \to x^* \text{ as } n \to \infty | x_1 \in \mathcal{U}_1) \geq 1 - \delta.$$
- Stochastically asymptotically stable if it is stochastically stable and attracting.

Intuition: if x_n starts close enough to x^* then, with arbitrarily high probability, it remains close enough and eventually converges to x^*.

Discrete-time version of the "folk theorem"

Theorem 1 (Giannou et al. 2021)

Let $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ *and suppose that* ExpWeight *is run under the following assumptions:*

$$\sum_n \gamma_n = \infty, \qquad \sum_n \gamma_n B_n < \infty, \qquad \sum_n \gamma_n^2 \sigma_n^2 < \infty.$$

Then:

1. x^* *is the limit of* x_n *with positive probability* $\Rightarrow$ x^* *is a Nash equilibrium.*
2. x^* *is stochastically stable* $\Rightarrow$ x^* *is a Nash equilibrium.*
3. x^* *is stochastically asymptotically stable* $\Leftrightarrow$ x^* *is a strict Nash equilibrium.*

Summary

This lesson

- Different types of feedback: mixed payoff vectors, pure payoff vectors, bandit.
- Scalar2Vector: Importance-weighted estimator.
- Stochastic first-order oracle model.
- Dominated strategies become extinct.
- Stochastic stability $\Rightarrow$ Nash equilibrium.
- Stochastic asymptotic stability $\iff$ strict equilibrium.

Next lesson

- Online optimization
- Regret minimization

References

[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. In: *Theory of Computing* 8.6 (2012), pp. 121–164 (cited at slide -16).

[2] Peter Auer et al. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In: *Proceedings of IEEE 36th Annual Foundations of Computer Science*. 1995, pp. 322–331 (cited at slides -15, -14).

[3] Peter Auer et al. The Nonstochastic Multiarmed Bandit Problem. In: *SIAM Journal on Computing* 32.1 (2002), pp. 48–77 (cited at slides -15, -14).

[4] Angeliki Giannou et al. From Learning with Partial Information to Bandits: Only Strict Nash Equilibria are Stable. In: *Proceedings of the 34th Annual Conference on Learning Theory*. 2021 (cited at slide -1).

[5] P. Hall and C.C. Heyde. *Martingale Limit Theory and its Application*. Elsevier, 1980 (cited at slide -6).

[6] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020 (cited at slide -13).

[7] Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. In: *Foundations and Trends® in Machine Learning* 4.2 (2012), pp. 107–194 (cited at slide -13).