ZEROTH-ORDER NON-CONVEX LEARNING VIA HIERARCHICAL DUAL AVERAGING

Amélie Héliou[1] Matthieu Martin[1] Panayotis Mertikopoulos[2,1] Thibaud Rahier[1]

[1]Criteo AI Lab

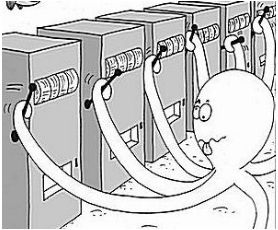
[2]French National Center for Scientific Research (CNRS)

ICML, 2021 — July 18–24, 2021

Gambling in a rigged casino

Hedge / EXP3 against an adversarial multi-armed bandit:

- u_t = *payoff vector* of stage $t = 1, 2, \dots$
- $P_t(a)$ = probability of choosing arm a at stage t (*mixed strategy*)
- $r_t = u_{t,a_t}$ = reward received at stage t from arm $a_t \sim P_t$



$\rightsquigarrow$

$$a_t \sim P_t$$

$$S_{t+1} = S_t + \begin{cases} u_t & \text{(Hedge)} \\ \mathbb{1}_{a_t} / P_t(a_t) \times r_t & \text{(EXP3)} \end{cases}$$

$$P_{t+1} \propto \exp(\eta_{t+1} S_{t+1})$$

Incurred regret: $\operatorname{Reg}(T) = \mathcal{O}(T^{1/2})$

*Gambling in a **continuous** casino*

What if the learner is facing a *continuum* of actions (e.g., in an online auction)?

- **Sequence of events:**
  - Select *action* x_t from compact convex set $\mathcal{K} \subseteq \mathbb{R}^d$
  - Adversary selects *payoff function* $u_t \colon \mathcal{K} \to [0, R]$ (non-convex, Lipschitz)
  - Learner receives reward $r_t = u_t(x_t)$ and the process repeats

*Gambling in a **continuous** casino*

What if the learner is facing a *continuum* of actions (e.g., in an online auction)?

- **Sequence of events:**
  - Select *action* x_t from compact convex set $\mathcal{K} \subseteq \mathbb{R}^d$
  - Adversary selects *payoff function* $u_t : \mathcal{K} \to [0, R]$ (non-convex, Lipschitz)
  - Learner receives reward $r_t = u_t(x_t)$ and the process repeats

- **Static regret:**
  - *Full info:* $\mathcal{O}(\sqrt{T})$ with knowledge of u_t [Krichene et al., 2015]
  - *Bandit:* $\mathcal{O}(T^{(d+1)/(d+2)})$ with knowledge of only r_t [Kleinberg, 2004]

. . . **but requires restart-and-forget with fixed discretization mesh**

*Gambling in a **continuous** casino*

What if the learner is facing a *continuum* of actions (e.g., in an online auction)?

- **Sequence of events:**
  - Select *action* x_t from compact convex set $\mathcal{K} \subseteq \mathbb{R}^d$
  - Adversary selects *payoff function* $u_t\colon \mathcal{K} \to [0, R]$ (non-convex, Lipschitz)
  - Learner receives reward $r_t = u_t(x_t)$ and the process repeats
- **Static regret:**
  - *Full info:* $\mathcal{O}(\sqrt{T})$ with knowledge of u_t [Krichene et al., 2015]
  - *Bandit:* $\mathcal{O}(T^{(d+1)/(d+2)})$ with knowledge of only r_t [Kleinberg, 2004]

  . . . **but requires restart-and-forget with fixed discretization mesh**
- **Dynamic regret** in terms of total variation $V_T = \sum_{t=1}^{T} \|u_{t+1} - u_t\|_\infty$:
  - *Full info:* $\mathcal{O}(T^{2/3} V_T^{1/3})$ with knowledge of u_t [Hériou et al., 2020]
  - *Bandit:* $\mathcal{O}(T^{(d+3)/(d+4)} V_T^{1/(d+4)})$ with knowledge of only r_t [Hériou et al., 2020]

  . . . **but suboptimal**

Our contributions

Combine

- Dual averaging template adapted to *Fisher information metric*
- Novel discretization schedule based on *dimension-wise exploration*

Our contributions

Combine

- Dual averaging template adapted to *Fisher information metric*
- Novel discretization schedule based on *dimension-wise exploration*

Achieve

- *Static regret:*

$$\mathbb{E}[\mathrm{Reg}(T)] = \mathcal{O}(T^{(d+1)/(d+2)})$$

✓ Optimal guarantee, no restarts or precise tuning required

- *Dynamic regret:*

$$\mathrm{DynReg}(T) = \mathcal{O}(T^{(d+2)/(d+3)} V_T^{1/(d+3)})$$

✓ Best known bound, anytime or not

Technical apparatus

- **Fisher information metric**

$$D_{\mathrm{Fish}}(p\|q) = \int_{\mathcal{K}} \left[\frac{d(p-q)}{dq}\right]^2 dp$$

- **Regularizer** $h(q) = \int_{\mathcal{K}} \theta(q)\, dq$ that is *strongly convex* relative to D_{Fish}

$$h(\lambda p + (1-\lambda) q) \leq \lambda h(p) + (1-\lambda) h(q) - \tfrac{1}{2} K \lambda (1-\lambda) D_{\mathrm{Fish}}(p\|q)$$

- **Choice map**

$$Q(y) = \arg\max_q \int_{\mathcal{K}} [q \cdot y - \theta(q)]\, dq$$

- **Dual averaging template**

$$y^+ \leftarrow y + \hat{u} \qquad p^+ \leftarrow Q(\eta y^+)$$

- **Examples:** negentropy (logit choice), log-barrier, ...

The splitting mechanism

Zooming in on areas of interest:

- **Given:** *partition* $\mathcal{P}$ of $\mathcal{K}$, coordinate i of $\mathbb{R}^d$
- **If** a *splitting event* occurs
- **Then** subdivide each $\mathcal{S} \in \mathcal{P}$ along x_i in two subsets of equal volume

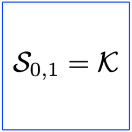


Figure: Example of the 3 first *splitting events* for $\mathcal{K} = [0, 1]^2$

The splitting mechanism

Zooming in on areas of interest:

- **Given:** *partition* $\mathcal{P}$ of $\mathcal{K}$, coordinate i of $\mathbb{R}^d$
- **If** a *splitting event* occurs
- **Then** subdivide each $\mathcal{S} \in \mathcal{P}$ along x_i in two subsets of equal volume

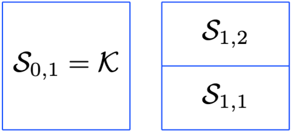


Figure: Example of the 3 first *splitting events* for $\mathcal{K} = [0,1]^2$

The splitting mechanism

Zooming in on areas of interest:

- **Given:** *partition* $\mathcal{P}$ of $\mathcal{K}$, coordinate i of $\mathbb{R}^d$
- **If** a *splitting event* occurs
- **Then** subdivide each $\mathcal{S} \in \mathcal{P}$ along x_i in two subsets of equal volume

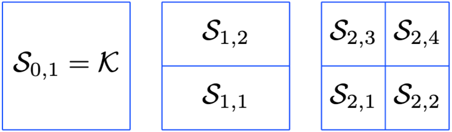


Figure: Example of the 3 first *splitting events* for $\mathcal{K} = [0, 1]^2$

The splitting mechanism

Zooming in on areas of interest:

- **Given:** *partition* $\mathcal{P}$ of $\mathcal{K}$, coordinate i of $\mathbb{R}^d$
- **If** a *splitting event* occurs
- **Then** subdivide each $\mathcal{S} \in \mathcal{P}$ along x_i in two subsets of equal volume

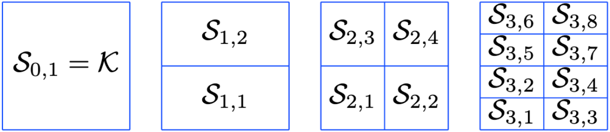


Figure: Example of the 3 first *splitting events* for $\mathcal{K} = [0,1]^2$

The splitting mechanism

Zooming in on areas of interest:

- **Given:** *partition* $\mathcal{P}$ of $\mathcal{K}$, coordinate i of $\mathbb{R}^d$
- **If** a *splitting event* occurs
- **Then** subdivide each $\mathcal{S} \in \mathcal{P}$ along x_i in two subsets of equal volume

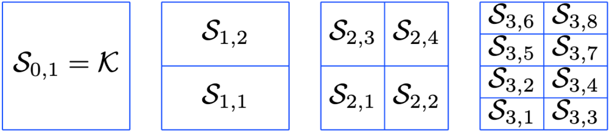


Figure: Example of the 3 first *splitting events* for $\mathcal{K} = [0,1]^2$

Increase resolution along dimensions cyclically, with logarithmic frequency

Hierarchical dual averaging

Algorithm Hierarchical dual averaging (HDA)

Require: learning rate η_t; splitting schedule υ_t; payoff estimator $\hat{u}_t$; compatible regularizer θ

1: **initialize:** $S_1 \leftarrow 0$; $\mathcal{P}_1 \leftarrow \{\mathcal{K}\}$
2: **for** $t = 1, 2, \dots$ **do**
3: **draw** $x_t \sim P_t = Q^{\mathcal{P}_t}(\eta_t S_t)$ # choose action from current cover
4: **get** $\hat{u}_t$ # observe / construct estimate
5: **set** $S_{t+1} \leftarrow S_t + \hat{u}_t$ # update score on cover
6: **if** $\lfloor \upsilon_t \rfloor = \lfloor \upsilon_{t-1} \rfloor + 1$ **then** $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_t^+$ # split cover when υ_t crosses an integer
7: **else** $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_t$ # leave cover “as is” otherwise
8: **end if**
9: **end for**

NB: $Q^{\mathcal{P}}$ is the *choice map* induced by h on a cover $\mathcal{P}$ of $\mathcal{K}$

Hierarchical exponential weights

Algorithm Hierarchical exponential weights (HEW)

Require: learning rate η_t; splitting schedule v_t; payoff estimator $\hat{u}_t$; compatible regularizer θ

initialize: $S_1 \leftarrow 0$; $\mathcal{P}_1 \leftarrow \{\mathcal{K}\}$
for $t = 1, 2, \dots$ **do**
    draw $x_t \sim P_t = \Lambda^{\mathcal{P}_t}(\eta_t S_t)$ # logit choice from current cover
    set $\hat{u}_t(x) = R - \dfrac{\mathbb{1}(x \in \mathcal{S}_t)}{P_t(x \in \mathcal{S}_t)}[R - u_t(x_t)]$ # importance weighted estimator
    set $S_{t+1} \leftarrow S_t + \hat{u}_t$ # update score on cover
    if $\lfloor v_t \rfloor = \lfloor v_{t-1} \rfloor + 1$ **then** $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_t^+$ # split cover when v_t crosses an integer
    else $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_t$ # leave cover "as is" otherwise
    end if
end for

NB: $\Lambda^{\mathcal{P}}$ is the *logit choice map* on a cover $\mathcal{P}$ of $\mathcal{K}$, i.e., $\Lambda^{\mathcal{P}}(y_{\mathcal{S}}) \propto \exp(y_{\mathcal{S}})$, $\mathcal{S} \in \mathcal{P}$

HEW guarantees

Order-optimal regret bounds under hierarchical exponential weights

Theorem (Static regret)

- **Assume:** *learning rate* $\eta_t \propto t^{-(d+1)/(d+2)}$*; splitting schedule* $\upsilon_t = \frac{d}{d+2} \log_2 t$
- **Then:** *HEW enjoys* $\mathrm{Reg}(T) = \mathcal{O}\big(T^{\frac{d+1}{d+2}}\big)$

Theorem (Dynamic regret)

- **Assume:** *total variation* $V_T = \mathcal{O}(T^{\nu})$*; learning rate* $\eta_t \propto t^{-(1-\nu)(d+1)/(d+3)}$*; splitting schedule* $\upsilon_t = \frac{d(1-\nu)}{d+3} \log_2 t$
- **Then:** *HEW enjoys* $\mathrm{DynReg}(T) = \mathcal{O}\big(T^{\frac{d+2}{d+3}} V_T^{\frac{1}{d+3}}\big)$

Experiments

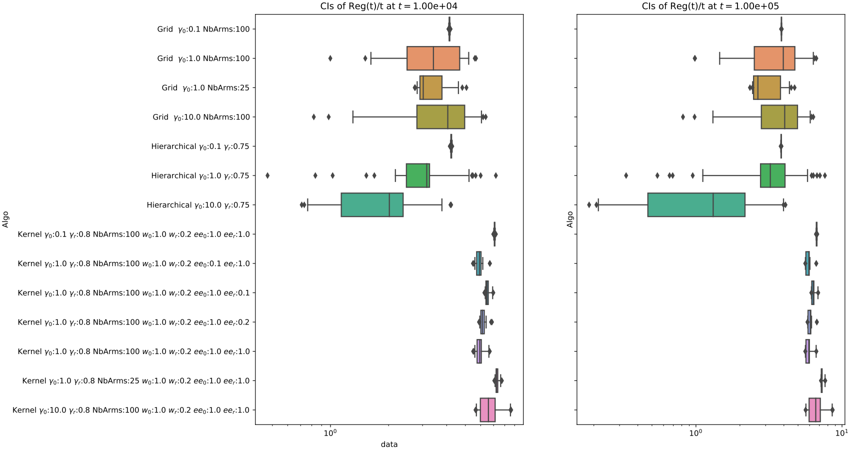


Figure: Comparative performance against a `Gauss2D` adversary.

References

A. Héliou, M. Martin, P. Mertikopoulos, and T. Rahier. Online non-convex optimization with imperfect feedback. In *NeurIPS '20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

A. Héliou, M. Martin, P. Mertikopoulos, and T. Rahier. Zeroth-order non-convex learning via hierarchical dual averaging. In *ICML '21: Proceedings of the 38th International Conference on Machine Learning*, 2021.

R. D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *NIPS' 04: Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, 2004.

W. Krichene, M. Balandat, C. Tomlin, and A. Bayen. The Hedge algorithm on a continuum. In *ICML '15: Proceedings of the 32nd International Conference on Machine Learning*, 2015.