Dying Experts: Efficient Algorithms with Optimal Regret Bounds

Hamid Shayestehmanesh, Sajjad Azami and Nishant Mehta

University of Victoria

Problem Setup

Standard Decision-Theoretic Online Learning

For round $t = 1, 2, \dots$

1. Learner plays a probability vector $\boldsymbol{p}_t$ over K experts
2. Nature reveals a loss vector $\boldsymbol{\ell}_t$
3. Learner suffers $\hat{\ell}_t = \boldsymbol{p}_t \cdot \boldsymbol{\ell}_t = \sum_{i=1}^{K} p_{i,t}\, \ell_{i,t}$

The classic notion of regret:

$$R_E(1,T) = \sum_{t=1}^{T} \hat{\ell}_t - \min_{i \in [K]} \sum_{t=1}^{T} \ell_{i,t}$$

The set of experts can be changing

$\rightarrow$ studied as Sleeping Experts/Specialists

The adversary chooses availability set A_t at each round as well

Using classic notion of regret (R_E) is not reasonable anymore

Ranking Regret

Define π to be an ordering over the set of initial experts $E = \{e_1, e_2, \dots, e_K\}$

For example: $\pi_i = (e_3, e_1, e_2)$

Let Π be the set of all possible orderings of E

Denote by $\sigma^t(\pi)$ the first alive expert of ordering π in round t

$$R_{\Pi}(1,T) = \sum_{t=1}^{T} \hat{\ell}_t - \min_{\pi \in \Pi} \sum_{t=1}^{T} \ell_{\sigma^t(\pi),t}$$

Previous Results on Sleeping Experts

- Fully-Adversarial setting
- Recall that regret of Hedge is $\mathcal{O}(\sqrt{T \log K})$
- Strategy in Kleinberg et. al. (2010) is to create all $K!$ orderings, we get $\mathcal{O}(\sqrt{TK \log K})$ with respect to ranking regret
- They also prove $\Omega(\sqrt{TK \log K})$
- Kanade and Steinke (2014) showed existence of a no-regret efficient algorithm for the sleeping experts setting implies the existence of an efficient algorithm for the problem of PAC learning DNFs

Dying Experts

- We are interested in a more restricted version of sleeping experts
- Motivated by disqualification or expiration of experts (e.g. fairness)
- The experts can only go to sleep (never wake up)

Can we get better results (regret/computation) in the seemingly easier case we are interested in?

Summary of Our Results

Question	Answer
Can we improve $\mathcal{O}(\sqrt{TK \log K})$?	No
Matching lower bound?	Yes
Ok, how about efficiency?	Yes

Question: what information can help improve regret?

$\rightarrow$ *Order of dying*

Question (Known order of dying)	Answer
Can we improve the upper bound?	Yes
Matching lower bound?	Yes
Efficiency?	Yes

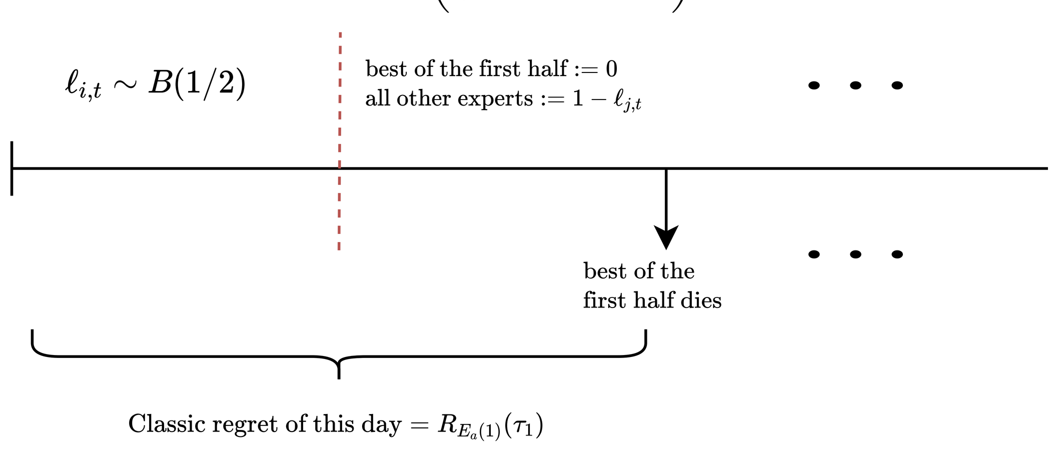
Lower bound (Unknown Order)

Theorem: When the order of dying is unknown, the regret of any algorithm is $\Omega(\sqrt{mT \log K})$.

Proof Sketch

- Partition the T rounds into $m+1$ days of equal length
- Each day is a game decoupled from the previous ones (goal: no prior info for algorithm)
- The days are split into two halves
- First half: $\ell_{i,t} \sim Bernouli(1/2)$
- The best expert of the first half suffers no loss on the second half, the others will suffer $1 - \ell_{j,t}$
- We show $R_{\Pi}(1,T) = \sum_{s=1}^{m+1} R_{E_a(s)}(\tau_s)$
- Using DTOL minimax regret, we get:

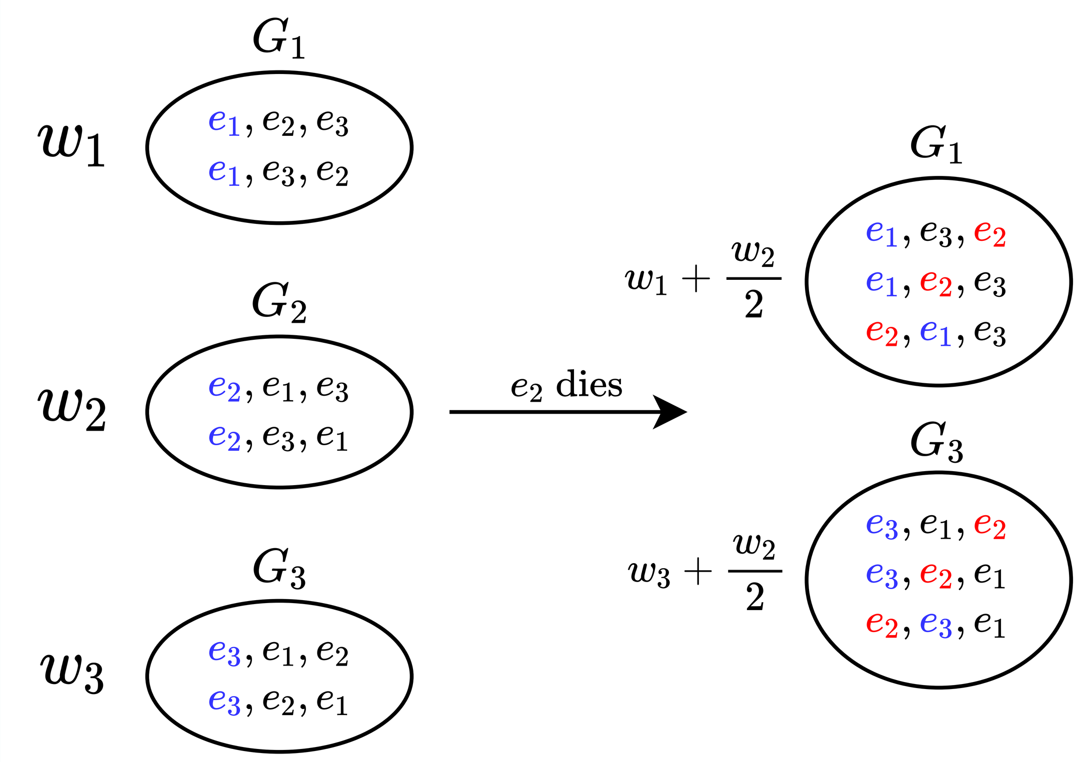
$$R_{\Pi}(1,T) = \sum_{s=1}^{m+1} \sqrt{\frac{T}{2}(m+1)\log(K-s)} = \Omega\left(\sqrt{Tm \log K}\right)$$



Effective Orderings

Note that not all $K!$ orderings are needed anymore

Number of *effective* orderings reduces from 6 (3!) to 2 (2!) after e_2 dies



Efficient Algorithm (Unknown Order)

$c_{i,1} = 1$, $h_{i,1} = (K-1)!$, $E_a = \{e_1, e_2, \dots e_K\}$

for $t = 1, 2, \dots T$ **do**

  play $p_{i,t} = \mathbf{1}\left[e_i \in E_a\right]\left(\frac{h_{i,t} \cdot c_{i,t}}{\sum_{j=1}^{k} h_{j,t} \cdot c_{j,t}}\right)$

  receive $(\ell_{1,t}, \dots, \ell_{K,t})$

  for $e_i \in E_a$ **do**

    $c_{i,t+1} = c_{i,t} \cdot e^{-\eta \ell_{i,t}}$

    $h_{i,t+1} = h_{i,t}$

  if *expert j dies* **then**

    $E_a = E_a \setminus \{e_j\}$

    for $e_i \in E_a$ **do**

      $h_{i,t+1} = h_{i,t+1} \cdot c_{i,t+1} + (h_{j,t+1} \cdot c_{j,t+1})/|E_a|$

      $c_{i,t+1} = 1$

The algorithm for the case of known order of dying is slightly different

We show that the algorithms simulate hedge over Π (for unknown order) and $\mathcal{E}$ (for known order)

Upper Bound (Unknown Order)

Strategy: Resetting-Hedge: run Hedge over the set of initial experts E and, after each night, reset the algorithm

Theorem: Resetting-Hedge strategy enjoys a regret of $R_{\Pi}(1,T) = \mathcal{O}(\sqrt{mT \log K})$.

Note: resetting can be wasteful in practice

Running on $K!$ orderings on the other hand is inefficient

$\rightarrow$ We propose an efficient implementation of Hedge over $K!$ orderings

Number of Effective Experts

- Assumption (for simplicity): the experts die in order, e_1 dies first, e_2 second, ...
- Behavior of π is a sequence of predictions $(\sigma^1(\pi), \sigma^2(\pi), \dots, \sigma^T(\pi))$
- π and π' behave the same if they use the same initial experts in *every* round.
- Set of effective orderings $\mathcal{E} \subseteq \Pi$: for each unique behavior of orderings, there only exists one ordering in $\mathcal{E}$.

Theorem: The number of effective orderings in Π is $f(\{d_1, d_2, \dots, d_m\}, A) = A \cdot \Pi_{s=1}^{m}(d_s + 1)$.

- d_i is the number of experts that die on i^{th} night
- If no expert dies, then $f(\{\}, A) = A$
- The maximum number of effective experts is $2^m(K-m)$

Bounds in Known Order

Strategy: Create effective orderings and run Hedge on them

Note: we only have $2^m(K-m)$ experts (orderings) instead of K!

Theorem: For the case of known order of dying, the strategy as described above achieves a regret of $\mathcal{O}(\sqrt{T(m + \log K)})$.

We have a matching lower bound:

Theorem: When Learner knows the order of dying, the minimax regret is $\Omega(\sqrt{mT})$.

Beyond Adaptivity to m

We show how Follow the Leader(FTL) algorithm can be implemented efficient while maintaining the loss of best permutation expert

Using FTL, we discuss how to set the learning rate in HPU/HPK to recover AdaHedge

Eventually, by combining AdaHedge and FTL, we implement FlipFlop to have an algorithm which does well in both adversarial and stochastic setting.

Corollary: HPU and HPK simulate FlipFlop over set of experts A (where $A = \Pi$ for HPU and $A = \mathcal{E}$ for HPK) and achieve regret

$$R_A(1,T) < \min\left\{C_2 \sqrt{\frac{L_T^*(T - L_T^*)}{T} \ln(|A|)}\right. \left. C_0 R_A^{\mathrm{ftl}}(1,T) + C_1, + C_3 \ln(|A|)\right\},$$

where C_0, C_1, C_2, C_3 are constants.