Online Learning with Stochastically Partitioning Experts

Puranjay Datta[1] Sharayu Moharir[1] Jaya Prakash Champati[2]

[1]Department of Electrical Engineering, IIT Bombay, India

[2]Computer Science Department, University of Victoria, Canada

Decision-Theoretic Online Learning (DTOL)

- N experts at the disposal of learner
- In round t, learner chooses a distribution $\boldsymbol{p}_t$ over the set of the experts
- Environment reveals loss vector $\boldsymbol{l}_t$
- Learner suffers loss $\langle \boldsymbol{p}_t, \boldsymbol{l}_t \rangle$
- Cumulative loss in T rounds:

$$L_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{l}_t \rangle \text{ for learner,}$$

$$L_T(i) = \sum_{t=1}^{T} l_t(i) \text{ for expert } i.$$

- **Objective:** minimize regret $= L_T - \min_i L_T(i)$

The celebrated *Hedge algorithm* achieves minimum achievable regret $O(\sqrt{T \log N})$.

Partitioning Experts: Variant of DTOL

Motivation
Out-of-Distribution (OOD) Detection

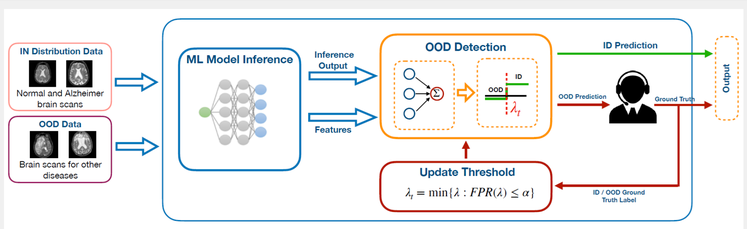

Figure 1. The threshold λ is updated online based on feedback.

Problem Formulation: Stochastically Partitioning Experts

Experts are partitions of a state space, revealed stochastically over time

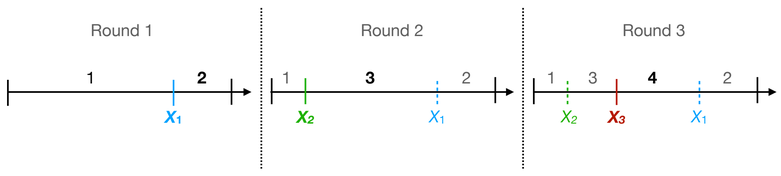

Figure 2. Stochastically Partitioning Experts in 1-D

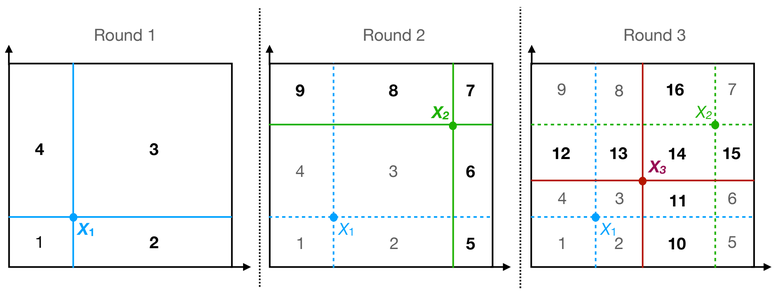

Figure 3. Stochastically Partitioning Experts in 2-D

- **Assumption:** Each new point X_t is drawn i.i.d.
- **Algorithmic Challenge:** Determine which expert to choose over time
- **Goal:** Minimize the total **expected regret**:

$$R_T = \mathbb{E}\left[L_T - \min_{i \in \mathcal{B}_T} \sum_{t=1}^{T} l_t(i)\right]$$

Algorithm 1: Hedge-G

An adaptation of Hedge for growing experts.

Initialize: $\mathcal{B}_0 = \{1\}$, $w_1 = 1$, $W_1 = 1$.

1. for $t = 1, \ldots, T$ do
2. Draw X_t, reveal new partitions, update expert set $\mathcal{B}_t$.
3. For each new expert $i \in \mathcal{B}_t \setminus \mathcal{B}_{t-1}$, set initial weight $w_t(i) = e^{-\eta L_{t-1}(\text{parent}(i))}$.
4. Update total weight $W_t \leftarrow W_t + \sum_{i \in \mathcal{B}_t \setminus \mathcal{B}_{t-1}} w_t(i)$.
5. Form prediction distribution: $p_t(i) = w_t(i)/W_t$.
6. Incur loss and update all weights: $w_{t+1}(i) = w_t(i) e^{-\eta l_t(i)}$.
7. end for

Regret Analysis of Hedge-G

Define $Y_t = \dfrac{\sum_{i=n_{t-1}+1}^{n_t} w_t(i)}{W_t} = \dfrac{\sum_{i=n_{t-1}+1}^{n_t} e^{-\eta L_{t-1}(i)}}{\sum_{j \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(j)}}$.

Y_t is the ratio of cumulative weight of new experts to the cumulative weight of old experts

Lemmas:

- $P(X_t \in \text{partition } i) = \dfrac{1}{t^d}$
- $\mathbb{E}[Y_t] \leq \dfrac{2^d}{t}$.

Theorem: Hedge-G Loss Bound

$$L_T \leq L_T^* + \frac{T\eta}{8} + \frac{1}{\eta} \sum_{t=1}^{T} Y_t$$

Corollary: Hedge-G Loss Bound

- **Expected Regret Guarantee:** For $\eta \propto 1/\sqrt{T}$

$$R_T = O(\sqrt{2^d T \log T})$$

- **Sample-Path Regret Guarantee:** For $\eta \propto 1/\sqrt{T^{1-\epsilon}}$

$$\tilde{R}_T = O(\sqrt{2^d T^{1+\epsilon} \log T})$$

Limitations of Hedge-G

1. No learning rate η simultaneously achieves optimal expected regret and sample-path regret guarantees
2. Our solution: An adaptive version of Hedge-G using a **doubling trick**. It runs Hedge-G in segments, restarting with a new learning rate when the accumulated cost of new experts, $S = \sum Y_t$, exceeds a dynamic threshold.

Algorithm 2: AdaHedge-G

A variant of Hedge-G

Initialize: Segment cost $S \leftarrow 0$, threshold $b \leftarrow 2^d$.

1. for $t = 1, \ldots, T$ do
2. Calculate Y_t.
3. if $S + Y_t > b$ then *% Start new segment*
4. Reinitialize all weights to be uniform.
5. Reset cost $S \leftarrow 0$, and double threshold $b \leftarrow 2b$.
6. Update learning rate: $\eta \leftarrow \sqrt{8(b + d \log t)/T}$.
7. end if
8. $S \leftarrow S + Y_t$.
9. Use one step of Hedge-G with current η.
10. end for

Regret Analysis of AdaHedge-G

Recall that Y_t is the ratio of cumulative weight of new experts to the cumulative weight of old experts

Theorem: AdaHedge-G Loss Bound

$$L_T \leq L_T^* + O\left(\sqrt{T\left(\sum_{t=1}^{T} Y_t + 1\right)}\right)$$

Corollary: AdaHedge-G Regret Bounds

(i) **Expected Regret:** $R_T = O(\log(\log T)\sqrt{T \log T})$
(ii) **Sample-Path Regret:** $\tilde{R}_T = O(\log T \sqrt{T \log T})$ with high probability.

- The doubling trick improves the dependency on the new expert cost from linear to square-root, eliminating the trade-off in Hedge-G
- AdaHedge-G simultaneously achieves near-optimal expected regret and a strong high-probability sample-path regret

Lower Bound and Optimality

We establish a lower bound on the regret for any algorithm in our setting by reducing it from the standard Prediction with Expert Advice problem.

Theorem: Lower Bound

$$R_T = \Omega(\sqrt{dT \log T})$$

AdaHedge-G is near-optimal, as its upper bound matches this lower bound up to a negligible $\log(\log T)$ factor.

References

1. Datta, P., Moharir, S., & Champati, J. P. (2024). *Online Learning with Stochastically Partitioning Experts*. ArXiv preprint.
2. Gofer, E., Cesa-Bianchi, N., Gentile, C., & Mansour, Y. (2013). *Regret minimization for branching experts*. In Conference on Learning Theory (COLT).
3. Freund, Y., & Schapire, R. E. (1997). *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of computer and system sciences.