

Linear Bandits

Matthew Nazari Moni Radev
`{matthewnazari, sraddev}@college.harvard.edu`

1 Online Learning with Experts

Definition 1.1 (full feedback). In this context, all costs are revealed at the end of every round t .

Definition 1.2 (experts). In a full feedback setting, instead of K arms each corresponding to an action, we have K experts who each predict one of L labels. In the case of binary prediction, expert e_i recommends a binary label: $z_{i,t} \in \{0, 1\}$.

For a problem with K experts and T rounds, consider the cost table ($c_t(a) : a \in [K], t \in [T]$). Imagine that costs are decided by some adversary, there are three types of costs:

- **Deterministic, oblivious adversary:** The cost table is chosen and fixed before round 1. The adversary chooses costs independent of our actions. Here,

$$\text{Regret}(T) = \text{cost}(\text{ALG}) - \min_{a \in [K]} \text{cost}(a)^1.$$

- **Randomized oblivious adversary:** The cost table is drawn from a random distribution of cost tables before round 1. If we measure the best arm *in foresight* instead of *in hindsight*, we get

$$\text{Regret}(T) = \text{cost}(\text{ALG}) - \min_{a \in [K]} \mathbb{E}[\text{cost}(a)].$$

- **Adaptive adversary:** Costs change depending on the algorithm's past choices. This models scenarios where our choices alter the environment that we operate in. We study regret in terms of the *best-observed arm*, which may not always be satisfactory but is worth studying for specific situations where our actions do not substantially affect the total cost of the best arm.

Algorithm 1.1 (Majority Vote Algorithm). Consider binary prediction with experts advice. In each round t , pick the action chosen by the majority of the experts who did not err in the past.

Theorem 1.1. Assuming a perfect expert, the Majority Vote Algorithm takes at most $\log_2 K$ mistakes.

Instead of losing trust in an expert completely after one mistake, simply downweight our confidence by some factor.

Algorithm 1.2 (Weighted Majority Algorithm, WMA). Given a parameter $\epsilon \in [0, 1]$, initialize confidence weights $w_{a,1} = 1$ for all experts a . Make prediction $z_t \in [L]$ using weighted majority vote. Update weights for incorrect experts as follows: $w_{a,t+1} \leftarrow w_{a,t}(1 - \epsilon)$

Theorem 1.2. The number of mistakes WMA makes with $\epsilon \in (0, 1)$ is at most

$$\frac{2}{1 - \epsilon} \text{cost}^* + \frac{2}{\epsilon} \log K.$$

¹ $\text{cost}(a) := \sum_{t \in [T]} c_t(a)$

However, any deterministic algorithm has total cost T for some deterministic, oblivious adversary. The adversary knows and can rig costs to hurt the algorithm. Therefore, we define a randomized algorithm.

Algorithm 1.3 (Hedge Algorithm). Given a parameter $\epsilon \in (0, \frac{1}{2})$, initialize confidence weights as in WMA. At each round t sample an arm from $p_t(a)$ where

$$p_t(a) := \frac{w_{a,t}}{\sum_{a'=1}^K w_{a',t}}.$$

Observe the cost $c_t(a) \in \{0, 1\}$ and update each arm's weight $w_{a,t+1} \leftarrow w_{a,t}(1 - \epsilon)^{c_t(a)}$.

2 Online Routing Problem

In “linear bandits,” an action is a low-dimensional vector and the costs that round are linear. The cost at any round t for an action $a \in \mathbb{R}^d$ is $c_t(a) := a \cdot v_t$ where $v_t \in \mathbb{R}^d$ differs per round but not per arm.

In full feedback setting, use Hedge Algorithm with parameter $\epsilon = 1/\sqrt{dT}$ to achieve regret $\mathbb{E}[\text{Regret}(T)] \leq O(d\sqrt{dT})$.

In this problem, different feedbacks are

- **bandit feedback:** only cost $c_t(a_t)$ is observed;
- **semi-bandit feedback:** costs $c_t(e)$ for all $e \in a_t$ are observed;
- **full feedback:** costs $c_t(e)$ for all edges are observed.

Reduction to the Bandit Problem: Idea is to use the Hedge algorithm. This requires us to determine two things: a *selection rule* for using expert e_t to pick arm a_t , and defining “fake costs” $\hat{c}_t(e)$ for all experts.

Explore by choosing edge e uniformly at random.

Algorithm 2.1 (Semi-Bandit Hedge Algorithm). With our reduction from the Hedge Algorithm to the bandit problem, select an edge e randomly at uniform with probability γ . Choose a path that includes this edge and define fake costs as follows:

$$\hat{c}_t(e) = \begin{cases} \frac{c_t(e)}{\gamma/d} & \text{if event } \Lambda_{t,e} \text{ happens} \\ 0 & \text{otherwise} \end{cases}$$

where $\Lambda_{t,e}$ is the event that in round t , we choose random exploration and edge e to explore.

We want fake costs to be unbiased estimates of true costs, so that

$$\mathbb{E}[\widehat{\text{Regret}}_{\text{Hedge}}] \geq \mathbb{E}[\text{Regret}_{\text{Hedge}}(T)].$$

3 Combinatorial Semi-Bandits

Replace edges with d “atoms,” and $u - v$ paths with feasible subsets $a \in \mathcal{F}$ of atoms S .

Notable special cases include

- *news articles:* a site can only select a subset of articles to display. A user can either click or ignore on each article, and feasible subsets represent various constraints on articles.
- *advertisements:* a website can only display a subset of ads to each user, and the website receives payment if the user clicks and even which type of user clicks.

The largest challenge of solving bandit feedback version is estimating fake costs for all atoms in the chosen action.

4 Follow Perturbed Leader

Back to the full-feedback setting where we have any fixed arbitrary subset $\mathcal{A} \subset [0, 1]^d$ of feasible actions. We incur a cost $c_t(a) = v_t \cdot a$ and observe v_t after every round.

Definition 4.1 (optimization oracle). A subroutine which computes the best action for a given cost vector: $M(v) \in \operatorname{argmin}_{a \in \mathcal{A}} a \cdot v$.

Algorithm 4.1 (Follow the Leader). Choose the action $a_{t+1} = M(v_{1:t})$ where $v_{i:j} = \sum_{t=i}^j v_t \in \mathbb{R}^d$.

Follow the Leader breaks when adversarial costs can synchronize its costs to harm the algorithm: the total cost for any deterministic algorithm is again T . Consider

$$\begin{aligned}\mathcal{A} &= \{(1, 0), (0, 1)\} \\ v_1 &= \left(\frac{1}{3}, \frac{2}{3}\right) \\ v_t &= \begin{cases} (1, 0) & \text{if } t \text{ is even,} \\ (0, 1) & \text{if } t \text{ is odd.} \end{cases}\end{aligned}$$

Then

$$v_{1:t} = \begin{cases} \left(i + \frac{1}{3}, i - \frac{1}{3}\right) & \text{if } t = 2i, \\ \left(i + \frac{1}{3}, i + \frac{2}{3}\right) & \text{if } t = 2i + 1. \end{cases}$$

Algorithm 4.2 (Follow the Perturbed Leader, FPL). Pretend there was a 0-th round, with $v_0 \in \mathbb{R}^d \sim \mathcal{D}$.

Theorem 4.1. Assume $v_t \in [0, U/d]^d$ for some known parameter U . Then FPL achieves regret

$$\mathbb{E}[\operatorname{Regret}(T)] \leq 2U\sqrt{dT}$$

with running time in each round polynomial in d plus one call to the oracle.