Linear Bandits

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

1 Online Learning with Experts

Definition 1 (full-feedback)**.** In this context, all costs are revealed at the end of every round t.

Definition 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,

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

- **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

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

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

Theorem 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 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 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 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 $w_{a,t+1} \leftarrow w_{a,t}(1-\epsilon)^{c_t(a)}$.

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 ick arm a_t, and defining "fake costs" $\widehat{c}_t(e)$ for all experts.

2 Online Routing Problem

3 Combinatorial Semi-Bandits

4 Follow Perturbed Leader

5 Literature Review