CS6781 - Theoretical Foundations of Machine Learning

Lecture 16-17: Multiarmed Bandits

April 9, 2020

Lecturer: Nika Haghtalab *Readings: N/A*
Scribe: Varsha Kishore, Anmol Kabra, Ziyang Wu, and Jonathan Chang

1 Full versus Partial Information

In online *full information* setting, we are able to observe the historical costs of choosing any of the experts. More formally, at time t, after taking action $i \in [n]$ we know $c^t(j)$ for all actions $j \in [n]$. Such an assumption makes sense in some problems such as classification where by observing adversary's choice of labeled instances (x_t, y_t) we can calculate the cost of each classifier $h \in \mathcal{H}$ by taking $c^t(h) = \mathbb{I}(h(x_t) \neq y_t)$. On the other hand, many examples involve situations where we only observe the cost function for some of the experts, e.g., only the expert that we followed at that time step. Examples of this include online route optimization where we may only observe the total travel time for the route we took, or observe the traffic on different parts of this route.

This is referred to as *partial information* in online learning. Formally, when playing action i^t, we may observe costs of experts $c^t(j)$ for $j \in E_i$. When $E_i = [n]$, this is just the full information setting. When $E_{i^t} = \{i^t\}$, the online learning problem is commonly referred to Adversarial Multi-Armed Bandits. In more detail, in the adversarial multi-armed bandit setting, we have n experts and at each time step we choose one of the n experts and we only observe the cost of *that expert*.

In the following sections we will see an algorithm for the bandit setting and we will derive regret bounds for the algorithm. Since, we don't get to observe the cost of experts we didn't play in the bandit setting, we can never rule out picking an expert completely and we need to have some probability of picking every expert at any given point. This is referred to as *exploration* and allows us to uncover potentially great experts that we have not played in the past. On the other hand, similar to the full information setting, we also want to take experts whose past performances have been good with higher probability. This is referred to as exploitation. Therefore any bandit algorithm needs to balance exploration and exploitation of experts.

2 An Easy Algorithm for Adversarial Multi-Armed Bandits

In this section, we show that any no-regret algorithm from the full information setting can be altered to work for the bandit setting as well, albeit with slightly worse regret guarantees. For ease of presentation, in the following we make use of the Randomized Weighted Majority (RWM) Algorithm that is no-regret in the full information setting (See Section 2.4 of Lecture 11). The high level idea for this algorithm is as follows:
At iteration t,

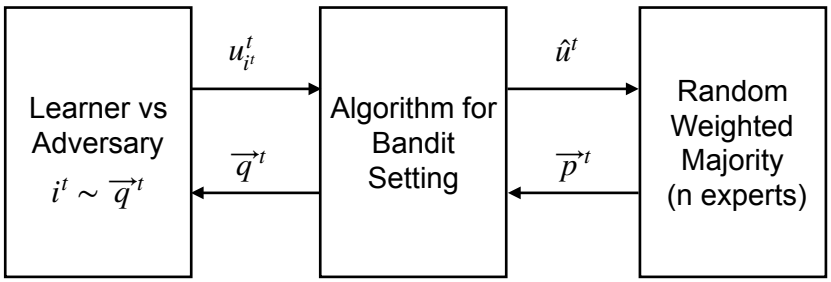

Figure 1: High level idea for algorithm in bandit setting

- Use probabilities suggested by a no-regret algorithm in a full information setting (such as RWM). Let us denote these probabilities by vector $\vec{p}^t$.

  Sample an expert i^t from a distribution $\vec{q}^t$, where $\vec{q}^t = (1-\gamma)\vec{p}^t + \gamma(\frac{1}{n}, \cdots, \frac{1}{n})$. This is equivalent to taking $i^t \sim \vec{p}^t$, $(1-\gamma)$ fraction of the time, and taking $i^t \sim [n]$ uniformly the rest of the time.

- Let the full-information algorithm update the weights of the experts. To do this, the algorithm needs to see the utilities (that is either the loss or reward) of every expert. For this we create a hypothetical utility vector as follows:

$$\hat{u}^t = \left(0, \cdots, 0, \frac{u^t_{i^t}}{q^t_{i^t}}, 0, \cdots, 0\right).$$

  Here $u^t_{i^t}$ is the utility of the one expert i^t that is chosen at time t and $q^t_{i^t}$ is the probability of choosing expert i^t. The vector $\hat{u}^t$ is then fed in to the full-information algorithm (such as RWM) to update the weights of the experts accordingly.

This high level idea is summarized in Figure 1. A few remarks:

1. At a high level, for the update rule to work well for the full information algorithm, we would want $\hat{u}^t$ to represent the full vector of utilities $\vec{u^t}$ in expectation. That is, $\mathbf{E}_{i^t \sim q^t}[\hat{u}^t] = (u^t_1, \cdots, u^t_n)$. We prove this later in this section.

2. The version of RWM that we described in Lecture 11 assumes that the utilities are in range $[0, 1]$. On the other hand, we are creating an expert with utility $\frac{u^t_{i^t}}{q^t_{i^t}} \in [0, n/\gamma]$. Therefore, we need to adapt the regret bound of RWM to scale with this larger range of utilities.

Algorithm 1 shows the algorithm that corresponds to using Figure 1. In this version, we consider utilities as rewards to the algorithm instead of costs. This results in $+\epsilon$ used in update rue of Algorithm 1 in line 6.

Algorithm 1 Reduction to Full Information

Initialize $w_1^1, \ldots, w_n^1 = 1$
for time $t = 1, \ldots, T$ **do**
  $\forall i \in [n], q_i^t \leftarrow (1-\gamma)\frac{w_i^t}{\sum_{j=1}^n w_j^t} + \frac{\gamma}{n}$
  Pick $i^t \sim \vec{q}^t$ and receive $u_{i^t}^t \in [0,1]$
  Let $\hat{u}^t = \left[0, \ldots, 0, \frac{u_{i^t}^t}{q_{i^t}^t}, 0, \ldots, 0\right]$
  $\forall i \in [n], w_i^{t+1} \leftarrow w_i^t \left(1 + \frac{\epsilon\gamma\hat{u}_i^t}{n}\right)$ %(Approximately equal to $w_i^t \exp\left(\frac{\epsilon\hat{u}_i^t}{n} \cdot \gamma\right)$
end for

Theorem 2.1 (Reduction to Full Information)**.** *With $\gamma = \epsilon$, Algorithm1 guarantees that*

$$\mathbf{E}\left[\sum_{t=1}^T u_{i^t}^t\right] \geq \mathbf{E}\left[\max_j \sum_{t=1}^T u_j^t\right](1-\gamma)^2 - \frac{n}{\gamma^2}\ln(n)$$

It follows that $\text{REGRET} \leq T^{2/3}(n\ln(n))^{1/3}$ *if* $\gamma = \left(\frac{n\ln(n)}{T}\right)^{1/3}$.

We will prove the theorem by combining the proofs of several facts

Fact 2.2. *$\forall j \in [n], \hat{u}_j^t$ is an unbiased estimator of u_j^t.*

Proof. This follows from:

$$\mathop{\mathbf{E}}_{i^t \sim \vec{q}^t}\left[\hat{u}_j^t\right] = q_j^t \cdot \frac{u_j^t}{q_j^t} + \left(1 - q_j^t\right) \cdot 0 = u_j^t. \qquad \square$$

Let $\text{OPT}_{\text{RWM}} = \max_j \sum_{t=1}^T \hat{u}_j^t$ be the best utility in hindsight according to RWM and $\text{OPT} = \max_j \sum_{t=1}^T u_j^t$ be the true best in hindsight utility. Next we show that OPT_{RWM} is more competitive than OPT.

Fact 2.3. $\mathbf{E}_{i^t \sim \vec{q}^t}\left[\text{OPT}_{\text{RWM}}\right] \geq \text{OPT}$.

Proof. We will use Jensen's inequality in proving this fact for asserting that expectation of a maximum is greater than maximum of an expectation. This is because the maximum accounts for the random variable that is taken expectation of, when the maximum is inside of the expectation.

$$\mathop{\mathbf{E}}_{i^t \sim \vec{q}^t}\left[\text{OPT}_{\text{RWM}}\right] = \mathop{\mathbf{E}}_{i^t \sim \vec{q}^t}\left[\max_j \sum_{t=1}^T \hat{u}_j^t\right] \geq \max_j \mathop{\mathbf{E}}_{i^t \sim \vec{q}^t}\left[\sum_{t=1}^T \hat{u}_j^t\right] = \max_j \sum_{t=1}^T u_j^t = \text{OPT},$$

where the penultimate transition is by Fact 1.2. $\square$

Fact 2.4.

$$\sum_{t=1}^T \vec{p}^t \hat{u}^t \geq (1-\epsilon)\text{OPT}_{\text{RWM}} - \mathcal{O}\left(\frac{1}{\epsilon}\ln(n)\frac{n}{\gamma}\right)$$

Proof. This fact follows from Theorem 1.1 of Lecture 12 but using utilities instead of costs. Note that the utilities used by RWM in the bandit algorithm are in $[0, n/\gamma]$ instead of the usual $[0, 1]$, leading to a scaling factor in the last term of the expression above. □

Fact 2.5. $u^t_{i^t} \geq (1-\gamma) \cdot p^t_{i^t} \cdot \hat{u}^t_{i^t}$, *i.e., The true reward is a large fraction of* RWM*'s expected reward.*

Proof. Let $i = i^t$ for readability purposes. Then,

$$u^t_i = \hat{u}^t_i q^t_i = p^t_i \cdot \hat{u}^t_i \cdot \frac{q^t_i}{p^t_i} \geq p^t_i \cdot \hat{u}^t_i (1-\gamma).$$

Here, $p^t_i \cdot \hat{u}^t_i$ is RWM's reward and $\frac{q^t_i}{p^t_i}$ is a scaling factor, which is small as shown. □

Proof of Theorem 2.1. Combining all facts together, the bandit algorithm's reward is:

$$\begin{aligned}
\mathop{\mathbf{E}}_{i^t \sim \vec{q}^t} \left[\sum_{t=1}^{T} u^t_{i^t} \right] &\geq (1-\gamma) \mathop{\mathbf{E}}_{i^t \sim \vec{q}^t} \sum_{t=1}^{T} \left[p^t_{i^t} \cdot \hat{u}^t_{i^t} \right] && \text{(By Fact 1.5)} \\
&\geq (1-\gamma)(1-\epsilon)\, \mathbf{E}\left[\mathrm{OPT}_{\mathrm{RWM}}\right] - (1-\gamma)\frac{1}{\epsilon\gamma} n \ln(n) && \text{(By Fact 1.4)} \\
&\geq (1-\gamma)^2\, \mathbf{E}\left[\mathrm{OPT}\right] - \frac{n}{\gamma^2} \ln(n) && \text{(By Fact 1.3).}
\end{aligned}$$

□

3 EXP3 Algorithm: A Better Bandit Algorithm

While Algorithm 1 is a no-regret algorithm, its regret bound dependence $T^{2/3}$ is far from optimal. Below, we introduce EXP3 that enjoys better regret guarantees. Interestingly, EXP3 is essentially equal to Algorithm 1, with the exception that its update rule does not have an additional γ dependence (See line 6). This requires a much more careful analysis of EXP3 that give us the following regret bound.

Algorithm 2 EXP3: Exponential Weights for Exploration and Exploitation

1: Initialize $w^1_1, \ldots, w^1_n = 1$
2: **for** time $t = 1, \ldots, T$ **do**
3: $\quad \forall i \in [n], q^t_i \leftarrow (1-\gamma)\frac{w^t_i}{\sum_{j=1}^{n} w^t_j} + \frac{\gamma}{n}$
4: $\quad$ Pick $i^t \sim \vec{q}^t$ and receive $u^t_{i^t} \in [0,1]$
5: $\quad$ Let $\hat{u}^t = \left[0, \ldots, 0, \frac{u^t_{i^t}}{q^t_{i^t}}, 0, \ldots, 0\right]$
6: $\quad \forall i \in [n], w^{t+1}_i \leftarrow w^t_i \exp\left(\frac{\epsilon \hat{u}^t_i}{n}\right)$
7: **end for**

Theorem 3.1. *(*EXP3 *Regret). When* $\gamma = \epsilon$*,* EXP3 *algorithm guarantees*

$$\mathbf{E}\left[\sum_{t=1}^{T} u_{i^t}^t\right] \geq \mathbf{E}\left[\max_j \sum_{t=1}^{T} u_j^t\right](1-\gamma) - \frac{n}{\gamma}\ln(n).$$

It follows that $\textsc{Regret}\,(\text{EXP3}) \leq \sqrt{Tn\ln(n)}$ *if* $\gamma = \sqrt{\frac{n\ln(n)}{T}}$.

EXP3 requires a more careful analysis (i.e. proving Theorem 3.1) than we have done for Theorem 2.1. We will not prove EXP3's regret bound in this lecture and instead refer the interested reader to Auer et al. [2002].

Let us remark that the regret bound of Theorem 3.1 is nearly tight in both its dependence on T and its dependence on n. Note that in the partial information setting our regret bounds grow with $\sqrt{\ln(n)}$ while in the partial information setting they grow with $\sqrt{n}$. This polynomial dependence in the partial information setting is unavoidable because we need to explore nearly all experts to uncover potentially excellent ones that have never been played before.

References

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Yoav Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121 (2):256–285, 1995.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.