CSE 522: Algorithms and Uncertainty Spring 2017

Lecture 11 — May 8, 2017

Lecturer: Nikhil R. Devanur

1 Bandits

No, this is not about highway robbery. When we discussed the shortest path problem with unknown delays, one of the assumptions was that we see the delays on all the edges. We now relax this assumption, and the resulting problems are studied under various names with the word "bandit" in them. Let's start with the most general online convex optimization problem.

Definition 1 (Convex bandits)**.** *The action space is an arbitrary set $S \subseteq \mathbb{R}^n$, which is given ahead of time. For $t = 1..T$,*

- *Pick $w_t \in S$.*
- *Simultaneously, the adversary picks a convex loss function ℓ_t.*
- *See $\ell_t(\mathbf{w}_t) \in \mathbb{R}^n$.*

The goal is to minimize the regret, defined as

$$\text{REGRET} = \sum_{t=1}^{T} \ell_t(\mathbf{w}_t) - \min_{\mathbf{u} \in S} \sum_{t=1}^{T} \ell_t(\mathbf{u}).$$

The main difference from the "full information" setting is the part where the feedback to the algorithm is just the evaluation of the function at $\mathbf{w}_t$, instead of the full convex function as before.

Remark 1. *For simplicity, we will consider oblivious adversaries, where the adversary picks all the loss functions ahead of time. For deterministic algorithms, there is no difference betweeen an oblivious and an adaptive adversary, since the adversary can simulate the algorithm. For randomized algorithms, such a simulation will only give the probability distribution over the actions, and not the actual action itself. An adaptive adversary can pick his loss functions based on the actual actions the algorithm picked in previous rounds (but not the current round), and is therefore potentially more powerful. In this case, we may measure the expected regret, but now the loss functions are themselves random variables. Most of the results do hold for adaptive adversaries, but we need to define what is called the pseudo regret. We will not get into the details of this in this class.*

1.1 Applications

Bandit problems were first studied as a problem in clinical trials. Patients are to be assigned to one of many potential treatments, where each treatment has different effectiveness, i.e., probability of curing a patient. We would like to cure as many patients as possible, without knowing a priori the effectiveness of each treatement.

Recently, bandit algorithms have found many uses in the online world. For instance, supose you have many different versions of a user interface and you want to deploy the most effective one, in terms of a specific response you want to elicit. Each interface has an effective response rate that you don't know about. Similarly, search engines would like to estimate the relevance of a result or an advertisement, which can be thought of as being roughly equivalent to the probability of click. We would like to maximize clicks, but we start with no knowledge of these probabilities.

1.2 Special cases:

In the full information setting we saw that there was essentially no difference between convex loss functions and linear losses. We could essentially replace ℓ_t by its gradient. This equivalence breaks down in the bandit case because of the information structure. The algorithm tries to learn about the loss function by carefully exploring the action space, and the information it can gather this way crucially depends on whether the loss is linear or convex. We therefore deal this special case separately. (In fact, the convex version is quite hard and only recently have there been reasonable algorithms for it. We will probably not have time for this.)

Definition 2 (Bandit linear optimization (BLO))**.** *This is the special case of Convex bandits where the loss functions are linear. In this case, we will abuse notation and write*

$$\ell_t(\mathbf{u}) := \ell_t \cdot \mathbf{u}.$$

Another equivalence we used in the full information setting was the equivalence between an action space and the convex hull of its representation in a vector space. Once again this equivalence does not quite hold in the bandit setting: suppose $\mathbf{u} \in S$ was actually a convex combination of the allowed actions. In the full information setting, we said we could randomize and pick an action in the action space whose expectation was $\mathbf{u}$. Since the losses were linear, this made no difference in the expected loss. Once agin, the bandit information changes things. Whether you observe $\ell_t(\mathbf{u})$ or $\ell_t(\mathbf{a})$ where $\mathbf{a}$ is a random variable whose expectation is $\mathbf{u}$ can make a difference. We will make this explicit when it comes up.

As a special case, we consider the bandit version of the experts problem, where the action space was a finite set of size n, which we thought of as vertices of a unit simplex in $\mathbb{R}^n$. This version is simply called the Multi-armed bandit problem.

Definition 3. *This is the special case of BLO where S is the unit simplex in $\mathbb{R}^n$ and the algorithm is required to pick one of the corners in each round.*

Further, in the shortest path problem, there are 2 versions. One, where you only observe the total delay of the path you have chosen. This corresponds to BLO as we have defined. The other version is where you observe the delay on each edge in the path. This version is called 'semi-bandit' feedback. Such feedback makes sense mostly in combinatorial settings, where in each round, you are required to pick some combinatorial set that satisfies certain property, such as the set of edges with the property that they form a path between two given nodes in a graph.

Definition 4 (Combinatorial semi-bandits)**.** *This is once again a special case of BLO with the following properties. The action set A is some collection of subsets of $[n]$. We represent elements* *of A by their indicator vectors in $\mathbb{R}^n$.*[1] *If the algorithm plays $\mathbf{a}_t$ in round t, then it observes each co-ordinate of the loss vector ℓ_t that corresponds to an element in $\mathbf{a}_t$.*

[1] the incidence vector of a subset $\mathbf{a} \subset [n]$ is a vector of 0s and 1s, where the co-ordinate i is 1 if and only if i is an element of the subset $\mathbf{a}$.