CSE599s, Spring 2012, Online Learning Lecture 13 - 05/08/2012

Contextual Bandits

Lecturer: Brendan McMahan *Scribe: Sergey Feldman*

1 Review - The Bandit Problem

Last class, we introduced the "bandit" problem, i.e. online learning with partial feedback. Here is how it works. Let a be an action in set A (abusing notation, we will also use A to be the number of actions in set A). On the tth round:

1. The adversary picks a loss function for all possible actions $\ell_t(a) \in [0, 1]$.
2. We choose $a' \in A$ (using randomness). For the the exponentiated gradient (EG) algorithm, we would sample $a' \sim w_t(a)$.
3. We then pay $\ell_t(a')$, observing only our loss.

To deal with this setting, we used FTRL with the negative entropy regularizer (also known as EG), and replaced (the now unknown) $g_t \in [0, 1]^A$ with its unbiased estimate:

$$\tilde{g}_t(a) = \begin{cases} \frac{\ell_t(a')}{w_t(a')} & \text{if a = a'} \\ 0 & \text{else,} \end{cases}$$

with

$$E[\tilde{g}_t(a)] = g_t(a).$$

Using EG on $\tilde{g}$ is known as the EXP3 algorithm.

2 Contextual Bandits

So far we have defined regret with respect to a *single* best action. But this definition doesn't make a lot of sense in many real-world scenarios. Consider the case of serving ads on a search engine. Given the millions of different queries and ads, it is useless compare our algorithm's performance against a single best ad for all of the rounds.

So, what to do? A naive solution is to run EXP3 separately for each of the queries. But this solution is not the best. How do we decide what constitutes a separate query? Is "flower shop" different from "rose shop"? Thus, we turn to "contextual bandits," also known as "bandits with side information."

First, some notation. Let $x_t \in \mathbb{R}^n$ be the *context* on round t. Let $e : \mathbb{R}^N \to \Delta(A)$ be the *expert* that maps context x_t to the probability simplex $\Delta(A)$ over the set of actions A. In other words, each expert provides a probability distribution over all of the possible actions, as a function of the context available on the current round. We won't worry about where these experts are coming from, but just assume that we have them. In previous settings, we maintained and updated w_t, a distribution over actions. In the contextual bandit w_t is instead a distribution over experts. Thus, on tth round we replace "Choose $a' \sim w_t(a)$" with

(a) Choose expert $e' \sim w_t(e)$ and then

(b) choose $a' \sim e'(a)$,

after which we get feedback about our choice and the loss (or reward) of action a' only. Our goal is to minimize regret w.r.t. the post-hoc best *expert* (as opposed to the post-hoc best *action*). There are other formulations of the contextual bandits problem, but we will only consider this one, which is often called "multi-armed bandits with expert advice."

An Example

To get a better idea of this setting, let's look at one round of EXP3 using the ad example. We are given 4 experts: e_1, e_2, e_3, e_4. To keep things simple, assume these experts can only choose one ad (action) a from the set of ads (actions) $A = \{a_1, a_2, \ldots\}$. Here are some ads:

$a_1 =$ **"buy pet lizards"**

$a_2 =$ **"1-800-petunias"**

$a_3 =$ **"cheap mp3 players"**

$a_4 =$ **"find local florists"**

$a_5 =$ **"affordable dragon souls",**

and so on. Let's say on round t the experts made the following choices:

e_1 **chose** a_2

e_2 **chose** a_2

e_3 **chose** a_4

e_4 **chose** a_4

The algorithm chose e_1 by sampling from $w_t(e)$, but this expert's ad choice, a_2, was not clicked (i.e. it was the wrong action). The algorithm then assigns loss $\frac{\ell_t(a_2)}{w_t(a_2)}$ to expert e_1. We estimate that the rest of the experts e_2, e_3, and e_4 all incur a loss of 0. However, observe that we should be able to do better than this: we know expert e_2 also wanted to play action a_2, and since a_2 was actually the ad we played, we should be able to use a better estimate for the loss of e_2. This is the idea of the EXP4 algorithm.

3 Regret Bounds for Bandits with Expert Advice

We assume that there are M experts and A actions. There are a number of algorithms we could run in the contextual bandit setting. Here they are, along with their expected regret:

Run EXP3 on the experts and get
$$E[\text{regret}] \leq \sqrt{2TM \log M}$$

Run the original EXP4 and get
$$E[\text{regret}] \leq \sqrt{TA \log M}$$

Run a revised EXP4 and get
$$E[\text{regret}] \leq \sqrt{TS \log M},$$

where $S \leq \min\{A, M\}$.

Clearly, the revised EXP4 is the best of both worlds, so that's what we're going to analyze. We will finish up today's lecture by establishing some notation.

Let $e_{t,i} \in \Delta(A)$ be the recommendation of the ith expert on the tth round, so $\sum_a e_{t,i}(a) = 1$ and $e_{t,i}(a) \geq 0$ for all i and t. For the contextual bandit setting we can write $e_{t,i} = \hat{e}_i(x_t)$, where $\hat{e}_i$ is a function that makes recommendations strictly as a function of the context. We will analyze the more general setting, allowing the experts to vary with time and feedback: for our purposes, an expert i is nothing more than t vectors $e_{t,i} \in \Delta(A)$. We don't care how these are generated, our bounds will only hold with respect to whatever set of expert recommendations the algorithm receives.

Let random variable L_i be the cumulative loss of the ith expert. Its expectation can be written

$$E[L_i] = \sum_{t=1}^{T} \sum_{a \in A} e_{t,i}(a) \ell_t(a).$$

Let

$$L_{\text{opt}} = \min_i E[L_i]$$

be the optimal post-hoc cumulative loss, and let

$$L_{\text{alg}} = \sum_{t=1}^{T} \ell_t(a'_t)$$

be the cumulative loss of the algorithm. The expected regret is then

$$E[\text{regret}] = E[L_{\text{alg}}] - L_{\text{opt}}.$$