CSE 599S: Online Learning: Bandits with Expert Advice – Analysis of EXP4* **(Spring 2012)**

Lecture 14

May 10th, 2012

Instructor: Dekel & McMahan *L. Elisa Celis*

1 Setup and Algorithm

We continue our investigation of online learning with partial information by considering the problem of bandits with expert advice. Instead of comparing against the best *single action*, we instead compare against the advice of the best *expert*. We think of an expert as a probability distribution over actions.[1] The setup is as follows:

- A actions $\{a_j\}$,
- M experts $\{e_i\}$ such that at time t, expert i gives probability distribution $e_{t,i}$ over actions (i.e. $e_{t,i}(a)$ is the probability with which expert i thinks action a is best at time t).

Given a loss vector ℓ_t on actions, we can in turn compute a loss vector g_t on experts by calculating their *expected loss*. Namely,

$$g_{t,i} = \mathbb{E}_{a\sim e_{t,i}}[\ell_t(a)] = e_{t,i} \cdot \ell_t.$$

Our goal is to perform well against the *best expert*, a harder (and more reasonable) task than performing well against a single action. Note, however, that by best expert we mean the *best sequence of probability distributions* $\{e_{t,i}\}_t$. Depending on the setting, it is *possible* that our experts are themselves running online algorithms, in fact, they may be conditioning on our behavior! However, we cannot compare against other *possible runs* of their algorithms – what we can compare against is their actual *behavior*, which is captured by these probability distributions.

Clearly, given a loss vector on experts, we can apply algorithms we have seen in the past, such as Follow the Regularized Leader, where the "leader" that we follow is an expert instead of an action. This bounds regret against an expert. However, recall that this algorithm is for the full information case where we have the full loss vector ℓ (and hence full loss vector g). In our case, as with all bandit settings, at each time step we choose a single action a_t^*, and only observe the loss of that one action $\ell_t(a_t^*)$. Thus, we will have to use an additional trick (similar to that of Lecture 12) to estimate $\tilde{\ell}_t$, and $\tilde{g}_t$ for use in our analysis. In summary, we will run the exponentiated gradient (EG) algorithm (i.e. multiplicative weight updates) on $\tilde{f}_t(w) = \tilde{g}_t \cdot w$. The key advantage of EXP4 (as opposed to just running EXP3 on the experts) is that by observing the outcome of a single action we may be able to gain information about multiple experts.

1.1 EXP4*

We will analyze EXP4, however use a different analysis which will yield a slightly stronger result, and hence add the * for clarity. The algorithm will maintain a probability vector over experts, and in each round will

[1] For computational tractability, we may make assumptions about finite support similar to our assumptions about a finite action space, though there is similar work considered for continuous space models.

choose an expert to follow, and then chose an action from the probability distribution that expert suggests. After observing the loss of that action, an estimate of the loss vectors over both actions and experts is updated, and a new probability distribution over experts is computed. The details for round t are as follows:

- Let $w_{t,i}$ be the probability with which we follow expert i in this round, defined as

  $$w_{t,i} = \exp(-\tilde{g}_{1:t-1,i})/Z$$

  where $Z = \sum_i \exp(-\tilde{g}_{1:t-1,i})$ is a normalizing factor.

- Sample expert $i_t^* \sim w_t$, and action $a_t^* \sim e_{t,i_t^*}$. Equivalently, consider the probability distribution such that $p_t(a) = \sum_i w_{t,i} e_{t,i}(a)$ and sample $a_t^* \sim p$.

- Play a_t^* and observe $\ell_t(a_t^*)$ (recall that this loss is chosen by the adversary).

- Update our unbiased estimator for ℓ_t:

  $$\tilde{\ell}_t(a) = \begin{cases} \frac{\ell_t(a_t^*)}{p_t(a_t^*)} & \text{if } a = a_t^*, \\ 0 & \text{otherwise.} \end{cases}$$

- Update our unbiased estimator for g_t:

  $$\tilde{g}_{t,i} = \mathbb{E}_{a \sim e_{t,i}}[\tilde{\ell}_t(a)] = e_{t,i} \cdot \ell_t = \frac{e_{t,i}(a_t^*)\ell_t(a_t^*)}{p_t(a_t^*)}.$$

Recall that an unbiased estimator simply means that the expected value of the estimator equals the true value of the value being estimated. This is easy to check for every coordinate of $\tilde{\ell}$ and $\tilde{g}$.

2 Regret Analysis

As in previous lectures, we assume $|\ell_t(a)| \leq G$ for all a, t and let $g_{1:T,i} = \sum_{t=1}^T g_{t,i}$. We can view our algorithm as internally running the EG algorithm (FTRL with entropic regularization) in the full-information setting against loss vectors $\tilde{g}_t$. This algorithm guarantees

$$\text{Regret}(T) \leq \sum_{t=1}^T w_t g_t - g_{1:T,i^*} \leq \frac{1}{\eta}\log(M) + \frac{\eta}{2}\sum_{t,i} w_{t,i} g_{t,i}^2. \tag{1}$$

This inequality was derived in Lecture 12. The inequality considers the *true* loss of an expert g – however, we only have an estimate of this loss. We proceed by calculating the *expected* bound on the regret using our unbiased estimator $\tilde{g}$.

First, fix some t, and note that

$$\begin{aligned} \mathbb{E}\left[\tilde{g}_{t,i}^2 | w_t\right] &= \sum_a p_t(a) \left(\frac{e_{t,i}(a)\ell_t(a)}{p_t(a)}\right)^2 \\ &\leq \sum_a \frac{e_{t,i}(a)^2 G^2}{p_t(a)}. \end{aligned}$$

Now, recall that $p_t(a) = \sum_i w_{t,i} e_{t,i}(a)$, and let $s_t(a) = \max_i e_{t,i}(a)$ and $S_t = \sum_a s_t(a)$. Then, by using these definitions and exchanging the order of summation, we get

$$\begin{aligned}
\mathbb{E}\left[\sum_i w_{t,i}\tilde{g}_{t,i}^2 | w_t\right] &= \sum_i w_{t,i} \sum_a \frac{e_{t,i}(a)^2 G^2}{p_t(a)} \\
&\leq G^2 \sum_a \sum_i w_{t,i} \frac{e_{t,i}(a) s_t(a)}{\sum_i w_{t,i} e_{t,i}(a)} \\
&= G^2 \sum_a s_t(a) \frac{\sum_i w_{t,i} e_{t,i}(a)}{\sum_i w_{t,i} e_{t,i}(a)} \\
&\leq G^2 \sum_a s_t(a) \leq G^2 S_t.
\end{aligned}$$

Note the above holds for all t, and let $S = \max_t S_t$. Thus, our overall expected regret is bounded by

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{\eta}\log M + \frac{\eta}{2}\sum_t \sum_i w_{t,i}\tilde{g}_{t,i}^2 | w_t\right] &\leq \frac{1}{\eta}\log M + \frac{\eta}{2} G^2 \sum_t S_t \\
&\leq \frac{1}{\eta}\log M + \frac{\eta}{2} G^2 T S.
\end{aligned}$$

This is minimized for

$$\eta = \frac{\sqrt{2\log M}}{G\sqrt{TS}},$$

which yields the following expected regret bound:

$$\mathbb{E}[\text{Regret}] \leq G\sqrt{\frac{1}{2} TS \log M}.$$

2.1 Comparison to Other Algorithms

Recall that if we had full information, we could treat the experts as actions and could get a regret bound of

$$O(\sqrt{T \log M}.$$

In the bandit setting, running EXP3 on experts (i.e. we only update the loss of the selected expert), as in Lecture 12, we get a regret bound of

$$O(\sqrt{TM \log M}.$$

The original analysis of EXP4, which we have not seen in class, yields a regret bound of

$$O(\sqrt{TA \log M}).$$

So how do we measure up? From above, we see that our regret is

$$O(\sqrt{TS \log M}),$$

so it depends on the value of S. However, first note that since $e_{t,i}(a) \leq 1$ for all t, i, then for $t^* = \text{argmax}_t S_t$,

$$S = S_{t^*} = \sum_a \max_i e_{t^*,i}(a) \leq A.$$

Additionally, since $e_{t^*,i}$ is a probability distribution over a,

$$S = S_{t^*} \leq \sum_a \sum_i e_{t^*,i}(a) = \sum_i \sum_a e_{t^*,i}(a) = M.$$

Hence, this bound is no worse than that of either EXP3 or the original analysis of EXP4. In fact, it can be much better. Assume, for example, that we know that all the experts place positive probability on at most K different actions on any given round. Then $S \leq K$. Alternately, assume that all experts choose the same probability at every time step. Then $S = 1$ [2].

The key difference between our analysis of EXP4* and the original analysis EXP4 is that we work in loss space wheras the original is done in reward space. The fact that we are in loss space means that we are always dealing with nonnegative numbers, which is necessary in the analysis in Lecture 12 used to attain Equation 1 used here.

3 Further Discussion

Recall that we are analyzing regret; adding experts may increase regret because we have a larger search space. Additionally adding *good* experts means that we are comparing to something better, but this may still increase regret. However, adding good experts also has the benefit of potentially decreasing our actual *loss* – we can learn from this good expert and end up with a better *objective* performance.

Keep in mind throughout regret analysis that our goal is to compare against the *best reasonable alternative*, and we measure our success accordingly. The key observation one should find in this discussion is that, as always, one should choose their experts based on the particular problem. The problem should lead us to a reasonable guess as to the type of expert that could behave well, and we typically use the smallest class of experts that contain this type.

[2] Of course, in this case our regret is 0 so the bound could be improved, but this at least shows that this analysis can significantly improve over prior work