CSE599s, Spring 2014, Online Learning Lecture 7 - 04/22/2014

Exponentiated Gradient and Intro to Bandits

Lecturer: Ofer Dekel *Scribe: April Shen*

1 Exponentiated Gradient

1.1 Review

Recall that we saw a solution to the problem of online learning with expert advice using FTRL with entropic regularization, in which we use the regularization function

$$R(p) = \frac{1}{\eta}\left(\sum_{i=1}^{d} p_i \log p_i + \log d\right) + I_{\Delta_d}(p).$$

With this regularizer we get a regret bound of

$$\text{Regret} \le 2\sqrt{T \log d}.$$

The $\log d$ rather than d constant is particularly good, since it allows us to compare against a larger set of experts (and hence perform favorably against a larger set of comparators).

Since we're still using FTRL, it looks as though we still need to solve the following optimization on each round:

$$p_t = \operatorname*{argmin}_p \sum_{s=1}^{t-1} l_s \cdot p + R(p).$$

But it turns out we can derive a closed-form solution for this optimization.

1.2 Closed-Form Solution for FTRL with Entropic Regularization

Rewriting our optimization, we want to solve

$$\min_p p \cdot l_{1:(t-1)} + \frac{1}{\eta}\sum_{i=1}^{d} p_i \log p_i + I_{\Delta_d}(p). \tag{1}$$

To derive a closed-form solution, we'll apply the method of Lagrange multipliers. First note that the constraint function $I_{\Delta_d}(p)$ is equivalent to the pair of constraints

$$\forall i,\ p_i \ge 0$$
$$\sum_{i=1}^{d} p_i = 1.$$

We can ignore the first for now (in the end, we'll see we get a solution that satisfies it anyway) and hence rewrite the constraint as

$$I_{\Delta_d}(p) = \max_\lambda \lambda \left(1 - \sum_{i=1}^{d} p_i\right),$$

which is only finite if the sum of the p_is is equal to 1. Hence we can rewrite equation (1) as

$$\min_{p\in\mathbb{R}^d}\max_{\lambda\in\mathbb{R}} p\cdot l_{1:(t-1)} + \frac{1}{\eta}\sum_{i=1}^{d} p_i\log p_i + \lambda\left(1-\sum_{i=1}^{d} p_i\right) \tag{2}$$

$$= \min_{p\in\mathbb{R}^d}\max_{\lambda\in\mathbb{R}} L(p,\lambda), \tag{3}$$

where p is the primal variable, λ is the Lagrange variable or dual variable, and $L(p,\lambda)$ is called the Lagrangian. For technical reasons[1], strong duality holds, so this is equal to

$$= \max_{\lambda\in\mathbb{R}}\min_{p\in\mathbb{R}^d} L(p,\lambda). \tag{4}$$

Now we solve the inner minimization of (4) for a fixed λ, by setting the gradient of L with respect to p equal to zero. So we get for all i,

$$\frac{\partial L}{\partial p_i} = l_{1:(t-1),i} + \frac{1}{\eta}(1+\log p_i) - \lambda = 0 \tag{5}$$

$$\Rightarrow \log p_i = -\eta l_{1:(t-1),i} - (1-\eta\lambda) \tag{6}$$

$$\Rightarrow p_i = \frac{e^{-\eta l_{1:(t-1),i}}}{c} \tag{7}$$

for $c = e^{1-\eta\lambda}$. However, we know that the sum of the p_is must be equal to 1, so in fact c is the just the normalization constant, i.e.,

$$c = \sum_{i=1}^{d} e^{-\eta l_{1:(t-1),i}}.$$

Note also that this choice of p_i is nonnegative, as we wanted. Hence equation (7) with this constant c is our solution to problem (1).

1.3 The EG Algorithm

This gives us an algorithm called Exponentiated Gradient, which is as follows. For $t = 1,\ldots,T$:

- Define $\forall i$, $w_{t,i} = e^{-\eta l_{1:(t-1),i}}$.
- Define $p_t = \frac{w_t}{||w_t||_1}$.
- Draw $I_t \sim p_t$, incurring loss l_{t,I_t} and observing l_t.

We can also formulate the updates to w_t recursively:

$$w_1 = (1,\ldots,1)$$
$$w_{t,i} = (w_{t-1,i})(e^{-\eta l_{t,i}})\ \forall i$$

The recursive formulation makes the analogy to gradient descent quite apparent: rather than taking an additive step in the gradient direction, we essentially take a multiplicative step in the direction of the exponential of the gradient.

As a general aside, it's a good idea to check our intuitions against the math at the end of a derivation, to see if they line up. In this case they do: if an expert sucks really badly (high loss), we assign low probability to choosing that expert, and our probability of choosing an expert is highest if that expert's loss is 0.

[1] In particular, Slater's condition holds, which roughly states that the relative interior of the feasible set is nonempty. For more details, take a class in convex optimization.

It's also worth thinking about why we were able to get a smaller regret bound using entropic regularization as opposed to squared l2 or Euclidean regularization. Regularization can be used to encode prior knowledge about where the best point might be, so we want our regularization function to be minimized at the *a priori* most reasonable point. Intuitively, the Euclidean norm doesn't fit naturally with measuring distances in the probability simplex, while negative entropy is a natural function to use as it is minimized at the uniform distribution.

2 Adversarial Multi-Armed Bandits

We now move to the multi-armed bandits scenario, where rather than experts to take advice from, the analogy is of multiple slot machines with different payout probabilities (of course these both correspond to having multiple actions to take with differing losses). The catch here is that we only get feedback for the arm we choose to pull.

2.1 The Setting

We play the following game. For $t = 1, \dots, T$:

- Player chooses $p_t \in \Delta_d$ and draws $I_t \sim p_t$.
- Player incurs loss l_{t,I_t} and observes only l_{t,I_t}.

Our regret is still defined as

$$\text{Regret} = E\left[\sum_{t=1}^{T} l_{t,I_t}\right] - \min_{i\in\{1,\dots,d\}} \sum_{t=1}^{T} l_{t,i}.$$

Note that:

1. This is similar to the experts problem, but with less feedback, since the player never sees the full loss vector.
2. Because information now costs something (i.e., we can only learn about payout probabilities by actually pulling the arm and incurring loss), we need to both explore the various actions and exploit our current knowledge, hence the infamous exploration/exploitation tradeoff.
3. One common application of this framework is in online advertising, where we need to choose just a few ads to show to users and only receive click-through feedback for the ads we show.

2.2 The EXP3 Algorithm

The trick that allows us to conquer the bandits problem is that although the player doesn't observe the entire loss vector $l_t \in \mathbb{R}^d$, he can still estimate it. In other words, the player can construct $\hat{l}_t \in \mathbb{R}^d$, an unbiased statistical estimator of the loss[2].

Without prolonging the suspense, let

$$\hat{l}_t = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ l_{t,I_t}/p_{t,I_t} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

[2] Recall that a random variable X is an unbiased estimator of a if $E[X] = a$.

where the nonzero coordinate is at index I_t. Then, like magic, for any i,

$$E[\hat{l}_{t,i}|p_t] = (1 - p_{t,i}) \cdot 0 + p_{t,i} \cdot \frac{l_{t,i}}{p_{t,i}} = l_{t,i}.$$

So the EXP3 algorithm will involve constructing this $\hat{l}_t$ on each round and simply plugging it into the EG algorithm. The analysis of why and how this works will follow in a later lecture.