18.657: Mathematics of Machine Learning

Lecturer: Philippe Rigollet
Scribe: Zach Izzo

Lecture 15
Oct. 27, 2015

Part III

Online Learning

It is often the case that we will be asked to make a sequence of predictions, rather than just one prediction given a large number of data points. In particular, this situation will arise whenever we need to perform **online classification**: at time t, we have $(X_1, Y_1), \dots, (X_{t-1}, Y_{t-1})$ iid random variables, and given X_t, we are asked to predict $Y_t \in \{0, 1\}$. Consider the following examples.

Online Shortest Path: We have a graph $G = (V, E)$ with two distinguished vertices s and t, and we wish to find the shortest path from s to t. However, the edge weights $E_1, \dots, E_t$ change with time t. Our observations after time t may be all of the edge weights $E_1, \dots, E_t$; or our observations may only be the weights of edges through which our path traverses; or our observation may only be the sum of the weights of the edges we've traversed.

Dynamic Pricing: We have a sequence of customers, each of which places a value v_t on some product. Our goal is to set a price p_t for the tth customer, and our reward for doing so is p_t if $p_t \leq v_t$ (in which case the customer buys the product at our price) or 0 otherwise (in which case the customer chooses not to buy the product). Our observations after time t may be $v_1, \dots, v_t$; or, perhaps more realistically, our observations may only be $\mathbb{1}(p_1 < v_1), \dots, \mathbb{1}(p_t < v_t)$. (In this case, we only know whether or not the customer bought the product.)

Sequential Investment: Given N assets, a portfolio is $\omega \in \Delta^N = \{x \in \mathbb{R}^n : x_i \geq 0, \sum_{i=1}^N x_i = 1\}$. ($\omega$ tells what percentage of our funds to invest in each stock. We could also allow for negative weights, which would correspond to shorting a stock.) At each time t, we wish to create a portfolio $\omega_t \in \Delta^N$ to maximize $\omega_t^T z_t$, where $z_t \in \mathbb{R}^N$ is a random variable which specifies the return of each asset at time t.

There are two general modelling approaches we can take: statistical or adversarial. Statistical methods typically require that the observations are iid, and that we can learn something about future points from past data. For example, in the dynamic pricing example, we could assume $v_t \sim N(v, 1)$. Another example is the Markowitz model for the sequential investment example, in which we assume that $\log(z_t) \sim \mathcal{N}(\mu, \Sigma)$.

In this lecture, we will focus on adversarial models. We assume that z_t can be any bounded sequence of numbers, and we will compare our predictions to the performance of some benchmark. In these types of models, one can imagine that we are playing a game against an opponent, and we are trying to minimize our losses regardless of the moves he plays. In this setting, we will frequently use optimization techniques such as mirror descent, as well as approaches from game theory and information theory.

1. PREDICTION WITH EXPERT ADVICE

1.1 Cumulative Regret

Let $\mathcal{A}$ be a convex set of actions we can take. For example, in the sequential investment example, $\mathcal{A} = \Delta^N$. If our options are discrete–for instance, choosing edges in a graph–then think of $\mathcal{A}$ as the convex hull of these options, and we can play one of the choices randomly according to some distribution. We will denote our adversary's moves by $\mathcal{Z}$. At time t, we simultaneously reveal $a_t \in \mathcal{A}$ and $z_t \in \mathcal{Z}$. Denote by $\ell(a_t, z_t)$ the loss associated to the player/decision maker taking action a_t and his adversary playing z_t.

In the general case, $\sum_{t=1}^{n} \ell(a_t, z_t)$ can be arbitrarily large. Therefore, rather than looking at the absolute loss for a series of n steps, we will compare our loss to the loss of a benchmark called an **expert**. An expert is simply some vector $b \in \mathcal{A}^n$, $b = (b_1, \ldots, b_t, \ldots, b_n)^T$. If we choose K experts $b^{(1)}, \ldots, b^{(K)}$, then our benchmark value will be the minimum cumulative loss amongst of all the experts:

$$\text{benchmark} = \min_{1 \le j \le K} \sum_{t=1}^{n} \ell(b_t^{(j)}, z_t).$$

The **cumulative regret** is then defined as

$$R_n = \sum_{t=1}^{n} \ell(a_t, z_t) - \min_{1 \le j \le K} \sum_{t=1}^{n} \ell(b_t^{(j)}, z_t).$$

At time t, we have access to the following information:

1. All of our previous moves, i.e. $a_1, \ldots, a_{t-1}$,
2. all of our adversary's previous moves, i.e. $z_1, \ldots, z_{t-1}$, and
3. All of the experts' strategies, i.e. $b^{(1)}, \ldots, b^{(K)}$.

Naively, one might try a strategy which chooses $a_t = b_t^*$, where b^* is the expert which has incurred minimal total loss for times $1, \ldots, t-1$. Unfortunately, this strategy is easily exploitable by the adversary: he can simply choose an action which maximizes the loss for that move at each step. To modify our approach, we will instead take a convex combination of the experts' suggested moves, weighting each according to the performance of that expert thus far. To that end, we will replace $\ell(a_t, z_t)$ by $\ell(p, (b_t, z_t))$, where $p \in \Delta^K$ denotes a convex combination, $b_t = (b_t^{(1)}, \ldots, b_t^{(K)})^T \in \mathcal{A}^K$ is the vector of the experts' moves at time t, and $z_t \in \mathcal{Z}$ is our adversary's move. Then

$$R_n = \sum_{t=1}^{n} \ell(p_t, z_t) - \min_{1 \le j \le K} \sum_{t=1}^{n} \ell(e_j, z_t)$$

where e_j is the vector whose jth entry is 1 and the rest of the entries are 0. Since we are restricting ourselves to convex combinations of the experts' moves, we can write $\mathcal{A} = \Delta^K$. We can now reduce our goal to an optimization problem:

$$\min_{\theta \in \Delta^K} \sum_{j=1}^{K} \theta_j \sum_{t=1}^{n} \ell(e_j, z_t).$$

From here, one option would be to use a projected gradient descent type algorithm: we define

$$q_{t+1} = p_t - \eta(\ell(e_1, z_t), \ldots, \ell(e_K, z_T))^T$$

and then $p_{t+1} = \pi^{\Delta^K}(p_t)$ to be the projection of q_{t+1} onto the simplex.

1.2 Exponential Weights

Suppose we instead use stochastic mirror descent with Φ = negative entropy. Then

$$q_{t+1,j} = p_{t+1,j} \exp(-\eta \ell(e_j, z_t)), \quad p_{t+1,j} = \frac{q_t}{\sum_{l=1}^K q_{t+1,l}},$$

where we have defined

$$p_t = \sum_{j=1}^K \left(\frac{w_{t,j}}{\sum_{l=1}^K w_{t,l}} e_j \right), \quad w_{t,j} = \exp\left(-\eta \sum_{s=1}^{t-1} \ell(e_j, z_s) \right).$$

This process looks at the loss from each expert and downweights it exponentially according to the fraction of total loss incurred. For this reason, this method is called an **exponential weighting (EW) strategy**.

Recall the definition of the cumulative regret R_n:

$$R_n = \sum_{t=1}^n \ell(p_t, z_t) - \min_{1 \le j \le K} \sum_{t=1}^n \ell(e_j, z_t).$$

Then we have the following theorem.

Theorem: Assume $\ell(\cdot, z)$ is convex for all $z \in \mathcal{Z}$ and that $\ell(p, z) \in [0, 1]$ for all $p \in \Delta^K, z \in \mathcal{Z}$. Then the EW strategy has regret

$$R_n \le \frac{\log K}{\eta} + \frac{\eta n}{2}.$$

In particular, for $\eta = \sqrt{\frac{2 \log K}{n}}$,

$$R_n \le \sqrt{2n \log K}.$$

Proof. We will recycle much of the mirror descent proof. Define

$$f_t(p) = \sum_{j=1}^K p_j \ell(e_j, z_t).$$

Denote $\|\cdot\| := |\cdot|_1$. Then

$$\frac{1}{n} \sum_{t=1}^n f_t(p_t) - f_t(p^*) \le \frac{\eta \frac{1}{n} \sum_{t=1}^n \|g_t\|_*^2}{2} + \frac{\log K}{\eta n},$$

where $g_t \in \partial f_t(p_t)$ and $\|\cdot\|_*$ is the dual norm (in this case $\|\cdot\|_* = |\cdot|_\infty$). The 2 in the denominator of the first term of this sum comes from setting $\alpha = 1$ in the mirror descent proof. Now,

$$g_t \in \partial f_t(p_t) \Rightarrow g_t = (\ell(e_1, z_t), \ldots, \ell(e_K, z_t))^T.$$

Furthermore, since $\ell(p, z) \in [0, 1]$, we have $\|g_t\|_* = |g_t|_\infty \leq 1$ for all t. Thus

$$\frac{\eta \frac{1}{n} \sum_{t=1}^n \|g_t\|_*^2}{2} + \frac{\log K}{n\eta} \leq \frac{\eta}{2} + \frac{\log K}{\eta n}.$$

Substituting for f_t yields

$$\sum_{t=1}^n \sum_{j=1}^K p_{t,j} \ell(e_j, z_t) - \boxed{\min_{p \in \Delta^K} \sum_{j=1}^K \sum_{t=1}^n p_j \ell(e_j, z_t)} \leq \frac{\eta n}{2} + \frac{\log K}{\eta}.$$

Note that the boxed term is actually $\min_{1 \leq j \leq K} \sum_{t=1}^n \ell(e_j, z_t)$. Furthermore, applying Jensen's to the unboxed term gives

$$\sum_{t=1}^n \sum_{j=1}^K p_{t,j} \ell(e_j, z_t) \geq \sum_{t=1}^n \ell(p_t, z_t).$$

Substituting these expressions then yields

$$R_n \leq \frac{\eta n}{2} + \frac{\log K}{\eta}.$$

We optimize over η to reach the desired conclusion. □

We now offer a different proof of the same theorem which will give us the optimal constant in the error bound. Define

$$w_{t,j} = \exp\left(-\eta \sum_{s=1}^{t-1} \ell(e_j, z_s)\right), \quad W_t = \sum_{j=1}^K w_{t,j}, \quad p_t = \frac{\sum_{j=1}^K w_{t,j} e_j}{W_t}.$$

For $t = 1$, we initialize $w_{1,j} = 1$, so $W_1 = K$. It should be noted that the starting values for $w_{1,j}$ are uniform, so we're starting at the correct point (i.e. maximal entropy) for mirrored descent. Now we have

$$\begin{aligned}
\log\left(\frac{W_{t+1}}{W_t}\right) &= \log\left(\frac{\sum_{j=1}^K \exp\left(-\eta \sum_{s=1}^{t-1} \ell(e_j, z_s)\right) \exp(-\eta \ell(e_j, z_t))}{\sum_{l=1}^K \exp\left(-\eta \sum_{j=1}^{t-1} \ell(e_l, z_s)\right)}.\right) \\
&= \log\left(\mathbb{E}_{J \sim p_t}\left[\exp(-\eta \ell(e_J, z_t))\right]\right) \\
\text{Hoeffding's lemma } &\Rightarrow \leq \log\left(e^{\frac{1}{8}\eta^2} e^{-\eta \mathbb{E}_J \ell(e_J, z_t)}\right) \\
&= \frac{\eta^2}{8} - \eta \mathbb{E}_J \ell(e_J, z_t) \\
\text{Jensen's } &\Rightarrow \leq \frac{\eta^2}{8} - \eta \ell(\mathbb{E}_J e_J, z_t) = \frac{\eta^2}{8} - \eta \ell(p_t, z_t)
\end{aligned}$$

since $\mathbb{E}_J e_j = \sum_{j=1}^K p_{t,j} e_j$. If we sum over t, the sum telescopes. Since $W_1 = K$, we are left with

$$\log(W_{n+1}) - \log(K) \leq \frac{n\eta^2}{8} - \eta \sum_{t=1}^n \ell(p_t, z_t).$$

We have

$$\log(W_{n+1}) = \log\left(\sum_{j=1}^K \exp\left(-\eta \sum_{s=1}^n \ell(e_j, z_s)\right)\right),$$

so setting $j^* = \operatorname{argmin}_{1 \leq j \leq K} \sum_{t=1}^n \ell(e_j, z_t)$, we obtain

$$\log(W_{n+1}) \geq \log\left(\exp\left(-\eta \sum_{s=1}^n \ell(e_{j*}, z_s)\right)\right) = -\eta \sum_{t=1}^n \ell(e_{j^*}, z_t).$$

Rearranging, we have

$$\sum_{t=1}^n \ell(p_t, z_t) - \sum_{t=1}^n \ell(e_{j^*}, z_t) \leq \frac{\eta n}{8} + \frac{\log K}{\eta}.$$

Finally, we optimize over η to arrive at

$$\eta = \sqrt{\frac{8 \log K}{n}} \Rightarrow R_n \leq \sqrt{\frac{n \log K}{2}}.$$

The improved constant comes from the assumption that our loss lies in an interval of size 1 (namely $[0, 1]$) rather than in an interval of size 2 (namely $[-1, 1]$).

18.657 Mathematics of Machine Learning
Fall 2015