1 Introduction

Most of the course is concerned with the "batch" learning problem. In this lecture, however, we look at a different model, called "online". Let us first compare and contrast the two.

In "batch" learning, an algorithm takes the data (training samples) and returns a hypothesis. We assume a "static" nature of the world: a new example that we will encounter will be similar to the training set. More precisely, we suppose that all the training samples, as well as the test point, are independent and identically distributed. Hence, given the bunch $z_1, \ldots, z_n$ of training samples drawn from a distribution p, the quality of the learned function f is $\mathbb{E}_{z \sim p} \ell(f, z)$, where ℓ is some loss function. The questions addressed by statistical learning theory are: how many examples are needed to have small expected error with certain confidence? what is the lowest error that can be achieved under certain conditions on p? etc.

If the world is not static, it might be difficult to take advantage of large amounts of data. We can no longer rely on statistical assumptions. In fact, we take an even more dramatic view of the world. We suppose that there are no correlations whatsoever between any two days. As there is no stationary distribution responsible for the data, we no longer want to minimize some expected error. All we want is to survive no matter how adversarial the world is. By surviving, we mean that we do not do too badly relative to other "agents" in the world. In fact, the goal is to do not much worse than the best "agent" (this difference is called *regret*). Note that this does not guarantee that we will be doing well in absolute terms, as the best agent might be quite bad. However, this is the price we have to pay for not having any coherence between today and tomorrow. The goal of "online learning" is, therefore, to do almost as well as the best comparator.

Today we will describe the most famous setting, "prediction with expert advice". We will then try to understand why the algorithm for this setting works by making some abstractions and moving to the Online Convex Optimization framework. Finally, we will unify the two via Regularization and provide some powerful tools for proving regret guarantees.

2 Prediction with Expert Advice

Suppose we have access to predictions of N experts. Denote these predictions at time t by $f_{1,t}, \ldots, f_{N,t}$. Fix a convex loss function ℓ. We suppose that $\ell(a, b) \in [0, 1]$ for simplicity.

At each time step $t = 1$ to T,

- Player observes $f_{1,t}, \ldots, f_{N,t}$ and predicts p_t
- Outcome y_t is revealed
- Player suffers loss $\ell(p_t, y_t)$ and experts suffer $\ell(f_{i,t}, y_t)$

Denote the cumulative loss of expert i by $L_{i,t} = \sum_{s=1}^{t} \ell(f_{i,s}, y_s)$ and the cumulative loss of the player by $L_t = \sum_{s=1}^{t} \ell(p_s, y_s)$.

The goal of the Player is to minimize the regret

$$R_T = \sum_{t=1}^{T} \ell(p_t, y_t) - \min_{i \in 1 \dots N} \sum_{t=1}^{T} \ell(f_{i,t}, y_t) = L_T - \min L_{i,T}.$$

How can we make predictions based on the history so that R_T is small?

Solution (called *Exponential Weights* or *Weighted Majority* [4]): keep weights $w_{1,t}, \dots, w_{N,t}$ over experts and predict

$$p_t = \frac{\sum_{i=1}^N w_{i,t-1} f_{i,t}}{\sum_{i=1}^N w_{i,t-1}}.$$

Once the outcome is revealed and the losses $\ell(f_{i,t}, y_t)$ can be calculated, the weights are updated as

$$w_{i,t} = w_{i,t-1} \cdot \exp(-\eta \ell(f_{i,t}, y_t)),$$

where η is the learning rate parameter.

Theorem 2.1. For the Exponential Weights algorithm with $\eta = \sqrt{\frac{8 \ln N}{T}}$,

$$R_T \le \sqrt{(T/2) \ln N}$$

Proof, see [3, 2]. Let $W_t = \sum_{i=1}^N w_{i,t}$. Suppose we initialize $w_{i,0} = 1$ for all experts i. Then $\ln W_0 = \ln N$. Furthermore,

$$\begin{aligned}
\ln \frac{W_T}{W_0} &= \ln \sum_{i=1}^{N} w_{i,T} - \ln N \\
&= \ln \sum_{i=1}^{N} \exp(-\eta L_{i,T}) - \ln N \\
&\ge \ln(\max_{i=1,\dots,N} \exp(-\eta L_{i,T})) - \ln N \\
&= -\eta \min_{i=1,\dots,N} L_{i,T} - \ln N.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\ln \frac{W_t}{W_{t-1}} &= \ln \frac{\sum_{i=1}^N w_{i,t}}{\sum_{i=1}^N w_{i,t-1}} \\
&= \ln \frac{\sum_{i=1}^N \exp(-\eta \ell(f_{i,t}, y_t)) \cdot \exp(-\eta L_{i,t-1})}{\sum_{i=1}^N \exp(-\eta L_{i,t-1})} \\
&= \ln \frac{\sum_{i=1}^N \exp(-\eta \ell(f_{i,t}, y_t)) \cdot w_{i,t-1}}{\sum_{i=1}^N w_{i,t-1}} \\
&\le -\eta \frac{\sum_{i=1}^N \ell(f_{i,t}, y_t) w_{i,t-1}}{\sum_{i=1}^N w_{i,t-1}} + \frac{\eta^2}{8} \\
&\le -\eta \ell(p_t, y_t) + \frac{\eta^2}{8}
\end{aligned}$$

where the last inequality follows by the definition of p_t and convexity of ℓ (via an application of Jensen's inequality). The next to last inequality holds because for a random variable $X \in [a, b]$,

$$\ln \mathbb{E}e^{sX} \le s\mathbb{E}X + \frac{s^2(b-a)^2}{8}$$

for any $s \in \mathbb{R}$. See [3, 2] for more details.

Summing the last inequality over $t = 1, \ldots, T$ and observing that logs telescope,

$$\ln \frac{W_T}{W_0} \le -\eta \sum_{t=1}^{T} \ell(p_t, y_t) + \eta^2 T/8.$$

Combining the upper and lower bounds for $\ln W_T/W_0$,

$$L_T \le \min_{i,T} L_{i,T} + \frac{\ln N}{\eta} + \frac{\eta}{8}T.$$

Balancing the two terms with $\eta = \sqrt{\frac{8 \ln N}{T}}$ gives the bound.

□

We've presented the "Prediction with Expert Advice" framework and the Exponential Weights algorithm because they are the most widely known. However, the proof above does not give us much insight into why things work. Let us now go to a somewhat simpler setting by removing the extra layer of combining predictions of experts to make our own prediction. The following setting is closely related, but simpler to understand.

3 Online Gradient Descent

Consider the following repeated game:

At each time step $t = 1$ to T,

- Player predicts w_t, a distribution over N experts
- Vector of losses $\ell_t \in \mathbb{R}^N$ is revealed
- Player suffers $\ell_t \cdot w_t$, the vector product

The goal is to minimize the regret, defined as

$$R_T = \sum_{t=1}^{T} \ell_t w_t - \min_{w^* \in N-\text{simplex}} \sum_{t=1}^{T} \ell_t w^*.$$

Since the minimum over the simplex occurs at the vertices, we could equivalently write $\min_i$.

One can see that the game is closely related to the one introduced in the beginning of the lecture. Here w_t are the normalized versions of the weights kept by the Exponential Weights algorithm. Think of the loss vector ℓ_t as the vector of $\ell(f_{i,t}, y_t)$.

The repeated game we just defined is called an Online Linear Optimization game. In fact, we can define the Online Linear Optimization over any convex set K. In this case

$$R_T = \sum_{t=1}^{T} \ell_t w_t - \min_{w^* \in K} \sum_{t=1}^{T} \ell_t w^*.$$

Suppose we just perform gradient steps $w_{t+1} = w_t - \eta \ell_t$, followed by a projection onto the set K. It turns out that this is a very good (often *optimal*) algorithm. Let's prove that this algorithm is good.

Theorem 3.1 (Online Gradient Descent [5]). Suppose the Online Linear Game is performed by updating $w_{t+1} = w_t - \eta \ell_t$ with $\eta = T^{-1/2}$, followed by the projection onto the set K. Then the regret

$$R_T \leq GD\sqrt{T},$$

where G is the maximum norm of the gradient of ℓ_t's and D is the diameter of K.

Proof. By the definition of the update,

$$\|w_{t+1} - w^*\|^2 \leq \|w_t - \eta \ell_t - w^*\|^2 = \|w_t - w^*\|^2 + \eta^2 \|\ell_t\|^2 - 2\ell_t(w_t - w^*),$$

where $w^* \in K$ is the optimum which can only be computed in hindsight. Solving for the last term,

$$\ell_t(w_t - w^*) \leq \frac{\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2}{2\eta} + \frac{\eta}{2}\|\ell_t\|^2$$

Summing over time,

$$\sum_{t=1}^{T} \ell_t w_t - \sum_{t=1}^{T} \ell_t w^* \leq \frac{\|w_1 - w^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\ell_t\|^2.$$

Assuming the upper bound on the norms and choosing $\eta = \frac{D}{G\sqrt{T}}$, we obtain the bound on the regret. □

4 Regularization

Let us compare the bounds $DG\sqrt{T}$ and $\sqrt{(T/2)\log N}$. The asymptotic dependence on T is the same – this is typical for online learning if no second-order "curvature" information is taken into account. However, the bounds are different. For the simplex, the L_2-diameter $D \propto \sqrt{N}$ instead of $\sqrt{\log N}$. This difference is noticeable whenever one has a large (exponential in parameters) number of experts. So, have we lost something by considering the Online Gradient Descent instead of Exponential Weights? The answer is no. It turns out that Exponential Weights can be viewed as Online Gradient Descent, but in a different (dual) space. So, the difference in $\sqrt{N}$ vs $\sqrt{\log N}$ arises from the choice of Euclidean potential vs Entropy potential. The dual-space gradient descent is known as Mirror Descent (see Chapter 11 of [3]). Somewhat surprisingly, *regularization* leads to Mirror Descent, and this is a very general framework subsuming many known online algorithms.

By regularization we mean the following Follow The Regularized Leader (FTRL) algorithm

$$w_{t+1} = \arg\min_{w} \sum_{s=1}^{t} \ell_s w + \eta^{-1}\mathcal{R}(w)$$

where $\mathcal{R}$ is an appropriate convex differentiable regularizer. When we take $\mathcal{R}(w) = \frac{1}{2}\|w\|^2$, we obtain the Online Gradient Descent described above. When we take $\mathcal{R}(w) = \sum_{i=1}^{N} w(i) \log w(i)$, we get Exponential Weights for the simplex (or, more generally, Exponentiated Gradient).

References

[1] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.

[2] Nicolò Cesa-Bianchi and Gábor Lugosi. On prediction of individual sequences. *Ann. Statist*, 27(6):1865–1895, 1999.

[3] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[4] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[5] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.