COS-511: Learning Theory **Spring 2017**

Lecture 18: Expert Advice (Hedge Algorithm) & Online To Batch

Lecturer: Roi Livni

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

18.1 Expert Advice

In the last lecture we've applied OGD algorithm to the expert advice problem to obtain a regret bound of $O(\sqrt{nT})$. The first natural question is whether OGD obtain optimal rate? in terms of T and n. We will see in the next lecture that we can in fact achieve a much better rate of $O(\sqrt{T}\log n)$. A good way to obtain intuition as to the optimality of a solution is to consider an analogue stochastic setting: Suppose at each iteration the adversary chooses $\mathbf{g}_t$ IID. Given a sequence of IID examples $\mathbf{g}_1, \ldots, \mathbf{g}_t$, what is the best strategy for the learning? One thing the learner can do is to apply a learning algorithm. Therefor a simple ERM rule would be

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}^* \in \mathcal{K}} \mathbf{g}_t \cdot \mathbf{x}^*.$$

When discussing statistical learning theory we called this strategy ERM in online learning this is sometimes referred to as *follow the leader* approach. Since we assume that the samples are chosen IID, we know by radamacher theory that:

$$\mathbb{E}\left[\mathbf{g}_{t+1} \cdot \mathbf{x}_{t+1}\right] \le \min_{\mathbf{x}^* \in \mathcal{K}} \mathbb{E}\left[\mathbf{g}_{t+1} \cdot \mathbf{x}^*\right] + \mathfrak{R}_t(\mathcal{K}) + O\left(\frac{1}{\sqrt{t}}\right)$$

We have analysed the Rademacher complexity of the class $\mathcal{K}$: this is the class of ℓ_1 regulerized classifiers and we have that $\mathfrak{R}_t(\mathcal{K}) \le \sqrt{\frac{\log n}{t}}$. Thus, if at each iteration we choose the leader or apply an ERM rule we have that:

$$\sum \mathbb{E}(\mathbf{g}_t \cdot \mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \mathbb{E}(\mathbf{g}_t \cdot \mathbf{x}^*) \le \sum O(\sqrt{\frac{\log n}{t}}) = O(\sqrt{T \log n})$$

Therefore we see that, at least in the stochastic case OGD is sub–optimal. As we will see, we can achieve the same regret bound as in the stochastic case, but we will need to use a different algorithm. Namely the Multplicative Weight algorithm which we next discuss

Algorithm 1 Follow the Leader

Inititalization $\mathbf{x}_1 \in \mathcal{K}$.
for $t = 1, 2 \ldots T$ **do**
  Observe f_t and suffer cost $f_t(\mathbf{x}_t)$.
  Set $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t' \le t} f_{t'}(\mathbf{x}^*)$
end for
return

18.2 MW Algorithm

We next consider a different algorithm called *Multiplicative Weights*, or *Hedge* to the expert advice problem. Similar to the OGD algorithm, Hedge keeps weights over the different expert at each time setp and update those according to the observed losses. In contrast to OGD the update is multiplicative:

Algorithm 2 Hedge

Inititalization $W_1 = \mathbf{1} \in \mathbf{R}^n$ % $W_1 = (1, 1, \ldots, 1)$.
SET $\mathbf{x}_1 = \frac{1}{n} W_1$.
for $t = 1, 2 \ldots T$ **do**
  Pick $i_t \propto \mathbf{x}_t$. i.e. $p(i_t = i) = \mathbf{x}_t(i)$.
  incure loss $\mathbf{g}_t(i_t)$.
  Update Weights $W_{t+1}(i) = W_t(i) \cdot e^{-\epsilon \mathbf{g}_t(i)}$
  Set $\mathbf{x}_{t+1}(i) = \frac{W_{t+1}}{\sum_j W_t(j)}$.
end for
return

Lemma 18.1. *Let $\mathbf{g}^2$ denote the n-dimensional vector of pointwise square losses (i.e $\mathbf{g}^2(i) = (\mathbf{g}(i))^2$), let $\epsilon > 0$ and assume all losses $\mathbf{g}$ to be non-negative. The Hedge Algorithm satisfies for every expert i^*:*

$$\sum_{t=1}^{T} \mathbf{x}_t \cdot \mathbf{g}_t \leq \sum_{t=1}^{T} \mathbf{g}_t(i^*) + \epsilon \sum_{t=1}^{T} \mathbf{x}_t \cdot \mathbf{g}_t^2 + \frac{\log n}{\epsilon} \tag{18.1}$$

Proof. Set $\Phi_t = \sum_{t=1}^{T} W_t(i)$ for all $t \leq T$, and note that $\Phi_1 = n$.

Inspecting the sum of weights

$$\begin{aligned}
\Phi_{t+1} &= \sum_i W_t(i) e^{-\epsilon \mathbf{g}_t(i)} \\
&= \Phi_t \sum \mathbf{x}_t e^{-\epsilon \mathbf{g}_t(i)} && \mathbf{x}_t(i) = \frac{W_t(i)}{\sum W_t(j)} \\
&\leq \Phi_t \sum \mathbf{x}_t (1 - \epsilon \mathbf{g}_t(i) + \epsilon^2 \mathbf{g}_t(i)) && \forall x \geq 0,\ e^{-x} \leq 1 - x + x^2 \\
&= \Phi_t (1 - \epsilon \mathbf{x}_t \cdot \mathbf{g}_t + \epsilon^2 \mathbf{x}_t \cdot \mathbf{g}_t^2) \\
&\leq \Phi_t e^{-\epsilon \mathbf{x}_t \cdot \mathbf{g}_t + \epsilon^2 \mathbf{x}_t \cdot \mathbf{g}_t^2} && 1 + x \leq e^x
\end{aligned}$$

And by definition of expert i^* we have that:

$$W_T(i^*) = e^{-\epsilon \sum_{t=1}^{T} \ell_t(i^*)}.$$

Given that $W_T(i^*)$ is less then the sum of weights Φ_T, we have that

$$W_T(i^*) \leq \Phi_T \leq N e^{-\epsilon \sum \mathbf{x}_t \cdot \mathbf{g}_t + \epsilon^2 \sum \mathbf{x}_t \cdot \mathbf{g}_t^2}$$

Taking logarithm of both sides we get:

$$-\epsilon \sum_{t=1}^{T} \mathbf{g}_t(i^*) \leq \log N - \epsilon \sum_{t=1}^{T} \mathbf{x}_t \cdot \mathbf{g}_t + \epsilon^2 \sum_{t=1}^{T} \mathbf{x}_t \cdot \mathbf{g}_t^2$$

And the result follows immediately. $\square$

Theorem 18.2. *Apply Alg. 2 to the Online Expert problem, with* $\epsilon = \sqrt{\frac{\log n}{T}}$ *then*

$$Regret_T = O(\sqrt{T \log N})$$

Proof. First observe that $\mathbf{g}^2 \leq 1$ hence $\mathbf{x}_t \cdot \mathbf{g}_t^2 \leq 1$. Plugging this into Eq. 18.1 we obtain that for every i^*:

$$\sum \mathbf{x}_t \cdot \mathbf{g}_t - \mathbf{g}_t(i^*) \leq T\epsilon + \frac{\log n}{\epsilon}$$

The algorithm picks the action of expert i at iteration i according to $\mathbf{x}_t$ hence incurs expected loss of $\mathbf{x}_t \cdot \mathbf{g}_t$ overall we have that for our choice of ϵ:

$$\text{Regret}_T \leq T\epsilon + \frac{\log n}{\epsilon} \leq 2\sqrt{T \log n}$$

□

Currently, the MW algorithm and OGD algorithm seem like two completely different algorithms that happen to solve different instances of OCO. In future lectures we will see how they can both be derived from a unifying setting.

18.3 Online to Batch

We have already seen intuitively that some relation between online guarantees on the generalization bounds in the stochastic setting. For example, we could derive the SGD algorithm from the OGD algorithm, and we could achieve regret bounds that are similar to the performance of the ERM algorithm. We next show that indeed we can always derive generalization bounds for the statistical setting from the online framework.

Theorem 18.3 (Online to Batch)**.** *Let $\mathcal{A}$ be an OCO algorithm whose regret after T iteration is guaranteed to be* $\text{Regret}_T(\mathcal{A})$. *Let $f_1, \ldots, f_T$ be an IID sequence of convex functions bounded by 1 s.t. $\mathbb{E}(f_t) = f$. Then for any $\delta > 0$, with probability at least $(1-\delta)$ (over the sample of convex functions), it holds for $\bar{\mathbf{x}} = \frac{1}{T}\sum \mathbf{x}_t$:*

$$f(\bar{\mathbf{x}}) \leq \min_{\mathbf{x}^*} f(\mathbf{x}^*) + \frac{\text{Regret}_T(\mathcal{A})}{T} + \sqrt{\frac{8\log(2/\delta)}{T}} \tag{18.2}$$

Application to Learning Before proving the online to batch result, let us consider its application to learning convex problems: Assume that $(\mathbf{x}, y)$ are drawn IID from some arbitrary distribution D, and we let for every $\mathbf{w}$ $\mathcal{L}(\mathbf{w}) = \mathbb{E}(\ell(\mathbf{w}; (\mathbf{x}, y)))$, where we assume ℓ is a convex function in $(\mathbf{w})$ for every $(\mathbf{x}, y)$. Then by drawing and IID sequence of samplex $\{\mathbf{x}_t, y_t\}_{t=1}^T$ and applying on Online algorithm, we obtain (by setting $f_t(\mathbf{w}_t) = \ell(\mathbf{w}_t, (\mathbf{x}_t, y_t))$ with probability $(1-\delta)$ that

$$\mathcal{L}(\bar{\mathbf{w}}) \leq \mathcal{L}(\mathbf{w}^*) + O\left(\frac{\text{Regret}_T(\mathcal{A}) \log 1/\delta}{\sqrt{T}}\right)$$

In particular for every algorithm with regret $O(DG\sqrt{T})$, if $T > \frac{1}{D^2G^2\epsilon^2}$ we have w.h.p

$$\mathcal{L}(\bar{\mathbf{w}}) \leq \mathcal{L}(\mathbf{w}^*) + O(\epsilon)$$

Thus, we obtain a learning algorithm.

Note that we have seen this phenomena in the special case where we applied OGD algorithm for learning and obtained the SGD algorithm.

Proof of Online2Batch The proof (as always) relies on bounding the expected error and concentration result. The concentration inequality we will use is Azuma's inequality for martingales:

Definition 18.4. *A sequence of random variables $X_1, X_2, \ldots,$ is called a martingale is*

$$\mathbb{E}(X_{t+1}|X_t, \ldots, X_1) = X_t \quad \forall t > 0$$

Theorem 18.5 (Azuma's inequality)**.** *Let $\{X_i\}_{t=1}^T$ be a martingale of T random variables that satisfy $|X_i - X_{i+1}| \le 1$. Then:*

$$\mathbb{P}(|X_T - X_0| > c) \le e^{-\frac{c^2}{2T}}$$

For the proof we start by defining a sequence of martingales: Let us write $Z_t = f(\mathbf{x}_t) - f_t(\mathbf{x}_t)$ and $X_t = \sum_{i=1}^t Z_i$. We first verify that X_t is indeed a martingale. Notice that by definition we have:

$$\mathop{\mathbb{E}}_{f_t\sim D}(Z_t|X_{t-1}) = \mathbb{E}(f(\mathbf{x}_t)) - \mathop{\mathbb{E}}_{f_t\sim D}(f_t(\mathbf{x}_t)) = 0$$

Where the last equality is true, since f_t is independent of $\mathbf{x}_t$. Thus by definition of Z_t we have that:

$$\mathbb{E}(X_{t+1}|X_1, \ldots, X_t) = \mathbb{E}(Z_{t+1}|X_t) + X_t = X_t$$

In addition by the bound on f_t we have that

$$|X_{t+1} - X_t| = |Z_t| \le 1$$

Applying Azuma's inequality to the martingale $X_1, \ldots, X_T$ we have that:

$$P(X_T > c) \le e^{-\frac{c^2}{T}}$$

By definition of X_T, dividing by T and setting $c = \sqrt{2T\log 2/\delta}$ we have that:

$$\mathbb{P}\left(\underbrace{\frac{1}{T}\sum f_t(\mathbf{x}_t) - f(\mathbf{x}_t)}_{\Gamma_1} > \sqrt{\frac{2\log 2/\delta}{T}}\right) \le \frac{\delta}{2}.$$

A similar martingale can be defined to $\mathbf{x}^*$ and we obtain analogously

$$\mathbb{P}\left(\underbrace{\frac{1}{T}\sum f_t(\mathbf{x}^*) - f(\mathbf{x}^*)}_{\Gamma_2} < -\sqrt{\frac{2\log 2/\delta}{T}}\right) \le \frac{\delta}{2}.$$

Overall we obtain that

$$\begin{aligned} f(\bar{\mathbf{x}}_t) f(\mathbf{x}^*) \le & \\ \sum_{t=1}^T f(\mathbf{x}_t) - f(\mathbf{x}^*) = & \quad \text{convexity} \\ \Gamma_1 - \Gamma_2 + \frac{1}{T}\sum f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*) \le \frac{\text{Regret}_T(\mathcal{A})}{T} & + \Gamma_1 - \Gamma_2 \end{aligned}$$

Thus with probaibliyt at least $1 - \delta$ by our bounds for Γ_1, Γ_2 we obtain the result.